L'User's Guide to the MSU 4π Array
3rd Edition

National Superconducting Cyclotron Laboratory,
Michigan State University

in collaboration with

Department of Chemistry, University of Maryland

and

Department of Chemistry, SUNY - Stony Brook

and

Department of Physics, University of Iowa.

Chief Editors: A.M. Vander Molen and J.S. Winfield, Emeritus

Subservient Editor: N.T.B. Stone

Original version: November 1989
Last revised: October 17, 1996

Preface

"Why this document?"
— Jeff Karn

This document is a an attempt to descibe the various hardware and software associated with the MSU 4π Array. It is intended to give outside users and newcomers to the 4π a brief description of the detector and the equipment used to control it. References are made throughout to hardware manuals and to additional documentation on the acquisition software.

This is a 'living document' – as the 4π evolves, so must the User's Guide. An appendix has been added to aid devoted scribes in their venerable attempts to modify this document. This appendix also tells users how to receive the most up-to-date version of this paper.
–J.S.W. 1st Edition

Many changes have occured to the 4π detector since its inception and with J.W. moving on to the A1200 I felt it was time to revamp the L'Users guide. The basic content and structure is the same but an attempt to remove dated material and add the new was done. The most noted is the new transputer based acquisition system. Many other smaller changes in procedures and the like were also made. Because of this new errors may have been introduced and your help in correcting these is appreciated. As always suggestions and contributions are welcomed.

Editor-in-chief: A.M. Vander Molen.

Contents

Chapter 1

History of the MSU 4π Array

"If we do not succeed, then we run the risk of failure."
— Dan Quayle

1.1 Conception and Proposal

The MSU 4π Array is designed to detect all charged particles emitted from collisions of intermediate energy nucleus-nucleus collisions. The charged particles range from protons to fission fragments. Intermediate energy nucleus-nucleus collisions offer the possibility of studying the nuclear equation of state and phase transitions in nuclear matter. To achieve this goal, one must detect as many of the emitted particles as possible. As opposed to large detector arrays designed for high energy nucleus-nucleus collisions, the MSU 4π Array must deal with a very large range of charges and energies of the emitted particles.

The MSU 4π Array was conceived at a detector workshop at the Oak Ridge National Laboratory in late 1982. The basic premise of the MSU 4π Array was the logarithmic detection system that presented an increasingly thick detector to increasingly penetrating particles. The concept of the MSU 4π Array was presented at a Users' Workshop at NSCL in December, 1982 where many designs for major detection systems were presented including the MSU 4π Array, a Time Projection Chamber, a Streamer Chamber, and various spectrometers. The consensus of the Workshop was that the MSU 4π Array should be built with top priority. Prototyping commenced immediately.

1.2 Construction

Based on the idea of modular logarithmic detectors, a hexagonal prototype module was constructed and used in experiments and in a Ph.D. dissertation. This prototype featured 6 CaF_2/fast plastic phoswiches, a Bragg curve spectrometer, and a low pressure multi-wire counter. In the intervening time, Bicron announced a new long time constant plastic. This slow plastic was much easier to machine than the CaF_2 crystals but emitted less light and thus was not suitable for serving as a ΔE counter. The phoswiches were redesigned to feature a fast plastic ΔE and a slow plastic E. This new technique was tested successfully by building sixteen small phoswiches used in a close-packed array behind a multi-wire counter.

The geometry chosen for the Array was that of a truncated icosahedron consisting of 20 regular hexagonal faces and 12 regular pentagonal faces. Each of the hexagonal (pentagonal) faces was instrumented with six (five) light particle detectors, one Bragg curve spectrometer, and one low pressure multiwire counter. This geometry yielded only two different shapes which saved machining costs. A vacuum vessel was designed to house the sub-arrays in such as way that they could be easily removed and the structure still be intact. The resulting vacuum chamber prompted the nickname of "Soccer Ball".

The entrance and exit of the beam is accomplished through two opposite pentagonal faces. The entrance and exit faces also contain large weldments that support the vessel and detectors on two large bearings that allow the chamber to rotate for easy removal of the subarrays. Both the entrance and exit end weldments are constructed to allow unobstructed access to detector systems. The entrance weldment is instrumented with the target mechanism and target viewing camera and light. The exit weldment is presently instrumented with a forward array of 45 light particle telescopes called the High Rate Array (HRA).

1.3 Experiments and Publications to Date

The first experiment was done with the MSU 4π Array in April of 1988 using beams of 35 MeV/nucleon ^{40}Ar and 50 MeV/nucleon ^{12}C. A list of experiments carried out to date with the MSU 4π Array follows:

(with the K500)

Extraction of Global Variables – 87008
- 35 MeV/n Ar+V,Au
- 50 MeV/n C+C,Au

Phoswich forward array
MSU/NSCL

Evolution of Interaction Zone – extension of 86013
35,50 MeV/n O+Ni
Phoswich forward array
Hope College, MSU/NSCL

Production of Complex Fragments – Discretionary time
35,45 MeV/n Ne+NaF,Sc,Au
CRNL forward array and silicon telescopes
CRNL, Laval, Toronto, MSU/NSCL

(with the K1200)

Search for the Disappearance of Flow – 88012
45, 55, 65, 75, 85 MeV/n ^{40}Ar+V
Phoswich forward array
MSU/NSCL

Impact Parameter Dependence of High-Energy γ Rays in Heavy Ion Collisions – 88014
65 MeV/n ^{40}Ar+V,Au
Forward array, large Cerenkov, BaF_2, and BGO counters
MSU/NSCL

Entropy Production in Central Collisions – 88029
45, 75, 100 MeV/n ^{40}Ar+V,Au
Phoswich forward array and silicon-plastic telescopes
MSU/NSCL

Search for the Liquid-Vapor Phase Transition – 88016
25, 35, 45, 65, 75, 85, 95, 105, 115 MeV/n ^{40}Ar+Sc
Phoswich forward array and Bragg curve counters
MSU/NSCL

Disappearance of Flow – 90004
35, 45, 55, 65, 70, and 75 MeV/n ^{86}Kr+Nb

55, 75, 95, 105, 115, 125, 135, and 140 MeV/n ^{20}Ne+Al
Phoswich forward array and Bragg curve counters
MSU/NSCL + Iowa University

Space-Time Characteristics – 88026
80 MeV/n Ar+Sc
Phoswich forward array and 56-element hodoscope
MSU/NSCL

Small Angle Correlations – 90016
40 MeV/n ^{16}O+Al
CsI-plastic hodoscope and phoswich forward array
Hope College, MSU/NSCL

Impact-parameter independence of charged particle distributions – 88009
45, 80 MeV/n ^{36}Ar+KCl, 80 MeV/n ^{20}Ne+NaF,Mg
Silicon telescopes and CRNL forward array
CRNL, Laval U, and MSU/NSCL

Disappearance of Flow – 91045
and
Multifragment Final States – 91043
55–155 MeV/n ^{12}C+C and 25–60 MeV/n ^{129}Xe+La
Phoswich forward array and Bragg curve counters
MSU/NSCL, Stony Brook, and Iowa U

Isotope Yield Ratios – 91038
53 MeV/n ^{40}Ar,^{40}Ca,^{40}Cl+^{58}Ni,^{58}Fe
Silicon detectors, Phoswich forward array and Bragg curve counters
MSU/NSCL and Univ. of Maryland

Below are listed the publications to date in refereed scientific journals of results from the MSU 4π Array:

1. A Logarithmic Detection System Suitable for a 4π Array, G.D. Westfall, J.E. Yurkon, J. van der Plicht, Z.M. Koenig, B.V. Jacak, R. Fox, G.M. Crawley, M.R. Maier, B.E. Hasselquist, R.S. Tickle, and D. Horn,

Nucl. Instrum. Methods, **A238**, 347 (1985).

2. Determination of the Impact Vector in Intermediate Energy Heavy Ion Collisions, C.A. Ogilvie, D.A. Cebra, J. Clayton, S. Howden, J. Karn, A. Vander Molen, G.D. Westfall, W.K. Wilson, and J.S. Winfield, Phys. Rev. **C40**, 654 (1989).

3. Longitudinal Collective Motion in Intermediate Energy Heavy-Ion Collisions, C.A. Ogilvie, D.A. Cebra, J. Clayton, P. Danielewicz, S. Howden, J. Karn, A. Nadasen, A. Vander Molen, G.D. Westfall, W.K. Wilson, and J.S. Winfield, Phys. Lett. **B231**, 35 (1989).

4. Transverse Collective Motion in Intermediate Energy Heavy-Ion Collisions, C.A. Ogilvie, D.A. Cebra, J. Clayton, P. Danielewicz, S. Howden, J. Karn, A. Nadasen, A. Vander Molen, G.D. Westfall, W.K. Wilson, and J.S. Winfield, Phys. Rev. **C40**, 2952 (1989).

5. Azimuthal Asymmetry in Ar + V Collisions from E/A = 35 to 85 MeV, W.K. Wilson, W. Benenson, D.A. Cebra, J. Clayton, S. Howden, J. Karn, T. Li, C.A. Ogilvie, A. Vander Molen, G.D. Westfall, J.S. Winfield, B. Young, and A. Nadasen, Phys. Rev. **C41**, R1881 (1990).

6. Event Shape Analysis: Sequential versus Simultaneous Multifragment Emission, D.A. Cebra, S. Howden, J. Karn, A. Nadasen, C.A. Ogilvie, A. Vander Molen, G.D. Westfall, W.K. Wilson, J.S. Winfield, and E. Norbeck, Phys. Rev. Lett **64**, 2246 (1990).

7. Disappearance of Flow and Its Relevance to Nuclear Matter Physics, C.A. Ogilvie, W. Bauer, D.A. Cebra, J. Clayton, S. Howden, J. Karn, A. Nadasen, A. Vander Molen, G.D. Westfall, W.K. Wilson, and J.S. Winfield, Phys. Rev. **C42**, R10 (1990).

8. Observation of a minimum in collective flow for Ar + V collisions, D. Krofcheck, D.A. Cebra, M. Cronqvist, R. Lacey, T. Li, C.A. Ogilvie, A. Vander Molen, K. Tyson, G.D. Westfall, W.K. Wilson, J.S. Winfield, A. Nadasen, and E. Norbeck, Phys. Rev. **C43**, 350 (1991).

9. Bragg curve spectroscopy in a 4π geometry, D.A. Cebra, S. Howden, J. Karn, D. Kataria, M. Maier, A. Nadasen, C.A. Ogilvie, N. Stone, D.

Swan, A. Vander Molen, W.K. Wilson, J.S. Winfield, J. Yurkon, G.D. Westfall, and E. Norbeck, Nucl. Instrum. Methods, **A300**, 518 (1991).

10. Sources of light particles in peripheral collisions, P.L. Gonthier, J.D. Lenters, M.T. Vonk, D.A. Cebra, W.K. Wilson, A. vander Molen, J. Karn, S. Howden, A. Nadasen, J.S. Winfield, and G.D. Westfall; Phys. Rev. **C43**, R1504 (1991).

11. Mean field deflection in peripheral heavy-ion collisions, W.K. Wilson, D. Cebra, S. Howden, J. Karn, D. Krofchek, R. Lacey, T. Li, A. Nadasen, T. Reposeur, A. Vander Molen, C.A. Ogilvie, G.D. Westfall, and J.S. Winfield; Phys. Rev. **C43**, 2696 (1991).

12. Impact-parameter independence of participant energy spectra measured in symmetric heavy-ion collisions, C.A. Pruneau, G.C. Ball, E. Hagberg, D. Horn, S. Gilbert, L. Potvin, C. Rioux, C. St-Pierre, T.E. Drake, A. Galindo-Uribarri, G. Zwartz, D.A. Cebra, S. Howden, J. Karn, C.A. Ogilvie, A. Vander Molen, G.D. Westfall, W.K. Wilson, and J.S. Winfield; Nucl. Phys. **A534**, 204 (1991).

13. Impact-parameter dependence of high energy gamma ray production in heavy-ion collisions, T. Reposeur, J. Clayton, W. Benenson, M. Cronqvist, S. Hannuschke, S. Howden, J. Karn, D. Krofchek, A. Nadasen, C. Ogilvie, R. Pfaff, J.D. Stevenson, A. Vander Molen, G.D. Westfall, W.K. Wilson, J.S. Winfield, B. Young, M.F. Mohar, and D.J. Morrissey; Phys. Lett. B **276**, 418 (1992).

Chapter 2

Technical Description of the Detectors

"That's nothing to Bragg about."
— Gary Westfall

2.1 Vacuum Chamber

2.1.1 Description

The aluminum vacuum chamber that houses the detectors is a 32 face truncated icosahedron with 20 hexagon faces and 12 pentagon faces. Two of the pentagon faces serve as the beam entrance and exit ports. The ball has an inside diameter of 70 inches and a length of 101 inches. The chamber has a volume of 3400 cubic liters and weighs around 6000 pounds with all detector inserted. Over 60 feet of RTV was used to seal the aluminum pieces making up the framework of the array. Sealing on the 30 modules and 126 feed through plates required 145 feet of rubber O-rings. The ball is supported by the rims of the end weldments, which rest on a set of wheels attached to the support cradle. The wheels allow it to be rotated $\pm 180°$ about the beam line axis so that modules may be removed or replaced. For rotation the ball must be at atmospheric pressure and disconnected at the entrance and exit beam lines.

The blueprints for all parts of the vacuum chamber are in the filing cabinets in the design department under the heading **7-MDx-xx-x.**

2.1.2 Vacuum Feedthroughs

- **Module Feedthroughs:**

Each of the 30 module faces have 4 plates...

1. A 4 inch (ID) plate that contain 7 SHV and 5 BNC feedthroughs.
2. A 4 inch (ID) plate that contains 12 BNC feedthroughs.
3. A 6 inch (ID) plate with 4 valve-controlled gas feedthroughs. These are to be used for Bragg Curve in, Bragg Curve out, PPAC in and PPAC out.
4. A 3 inch (ID) plate used for PPAC signal feedthroughs.

- **Entrance End Plate:**
In addition to the beam entrance there are 4 openings on the end plate.

1. A 4 inch (ID) plate that presently has just a clear plexiglass blank off.
2. A special purpose plate that mounts the target mechanism.
3. A special entrance can that contains a camera for viewing the target with light bulb used for illuminating the target. The bulb is a 24 volt bulb and hooked into the MODICON.
4. A 4 inch(ID) plate with a mechanical vacuum gauge.
5. A Feed through plate for the target motor/MODICON signals.

- **Exit End Plate:**
In addition to the beam exit pipe there are 4 feedthrough plates on the exit end plate...

1. An 8 inch (ID) blank plate.
2. An 8 inch (ID) plate with 16 SHV, 16 BNC, and two 40 pin ribbon cable vacuum feedthroughs.
3. A 4 inch (ID) plate containing 31 Lemo vacuum feedthroughs.
4. A 10 inch (ID) plate with 45 SHV and 15 BNC vacuum feedthroughs
5. An 8 inch (ID) plate with BNC, and two value-controlled gas feedthroughs.

Table 2.1: Phoswich scintillator specifications

Element	Material	Thickness	Rise time	Decay time
Ball Fast ΔE	BC412	3 mm	1 ns†	3.3 ns†
Ball Slow E	BC444	25 cm	20 ns	180 ns
HRA Fast ΔE	NE110	1.7 mm	1.1 ns†	3.3 ns†
HRA Slow E	NE115	19.4 cm	8 ns	320 ns

†: signal from phototube will have significantly longer rise and fall times.

2.2 Phoswiches

2.2.1 Scintillators

The outermost detectors in the 4π Array are a set of phoswich scintillators. These detectors will stop all but the most energetic light ions. The phoswiches consist of a thin wafer of fast plastic scintillator (to measure the rate of energy loss), followed by a thick block of slow plastic scintillator (to stop the charged particle and measure its total energy). The specifications of the scintillation elements in the ball and the High Rate Array (HRA) are listed in Table 2.1 below.

2.2.2 Detector Geometries and Solid Angles

The phoswich detectors in the main ball are truncated triangular pyramids which are subdivisions of either hexagons (60^o, 60^o, 60^o) or pentagons (72^o, 54^o, 54^o). The **NSCL High Rate Array** (installed Dec. 1993) is constructed of five wedges which are subdivided into nine phoswich detectors. A schematic view of the NSCL HRA is shown in Fig. 2.1. All the phoswich detectors are close-packed, leaving only a layer of opaque epoxy and reflective foil between adjacent scintillators. The solid angle subtended by the ball phoswiches are listed in Table 2.2. The solid angles for the 45 phoswiches in the NSCL HRA are given in Table 2.3 (HRA detector numbering scheme is shown in Fig. 2.7).

2.2.3 Angles

Table 2.4 lists the mean angles for the ball phoswiches (in degrees). Here, θ is the *polar*

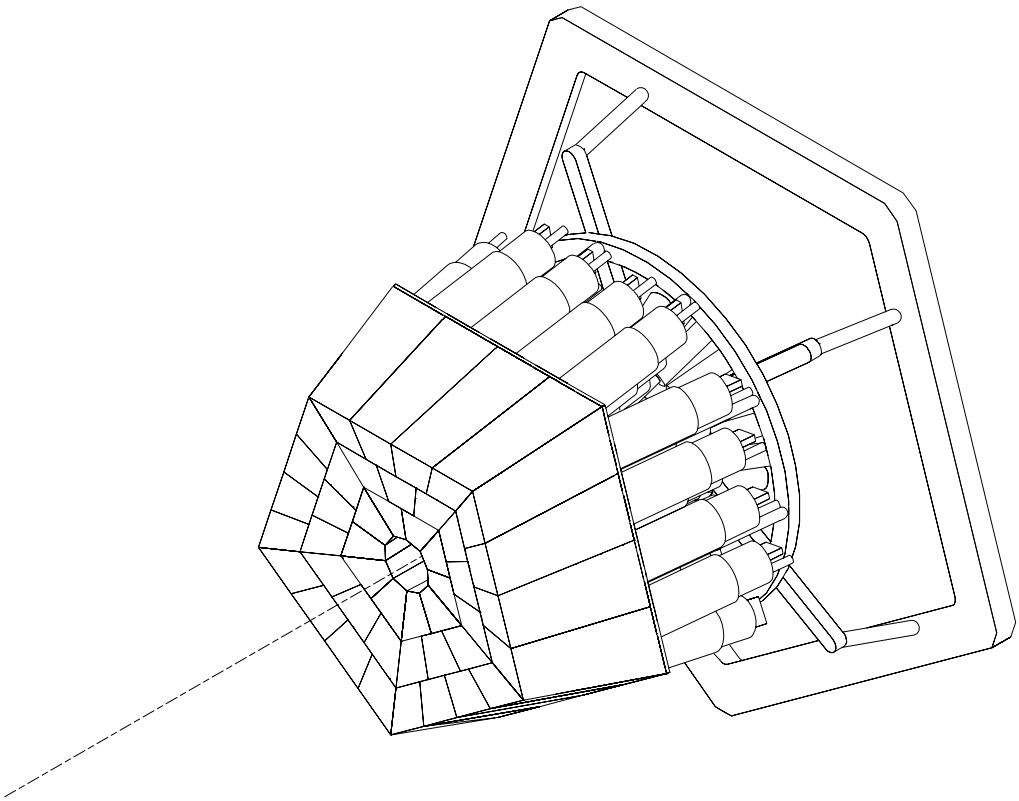


Figure 2.1: Schematic view of the NSCL High Rate Array (HRA).

Table 2.2: Solid angle subtended by the Ball Phoswiches

Type	Solid angle (msr)
Ball Hexagon	6×65.96
Ball Pentagon	5×49.92

Table 2.3: Solid angle subtended by the NSCL High Rate Array

HRA Detector Number	Solid Angle (msr) (each)
1,2,3,4,5,6,7,8,9,10	5.107
12,15,18,21,24	6.272
11,13,14,16,17,19,20,22,23,25	6.184
27,28,31,32,35,36,39,40,43,44	6.882
26,29,30,33,34,37,38,41,42,45	6.652

Table 2.4: Mean angles for the Ball Phoswiches

Module	A		B		C		D		E		F	
	θ	ϕ	θ	ϕ	θ	ϕ	θ	ϕ	θ	ϕ	θ	ϕ
1	23.1	342.0	32.3	5.6	46.0	356.3	51.7	342.0	46.0	324.7	32.3	318.4
2	23.1	270.0	32.3	293.6	46.0	287.3	51.7	270.0	46.0	252.7	32.3	246.4
3	23.1	198.0	32.3	221.6	46.0	215.3	51.7	198.0	46.0	180.7	32.3	174.4
4	23.1	126.0	32.3	149.6	46.0	143.3	51.7	126.0	46.0	108.7	32.3	102.4
5	23.1	54.0	32.3	77.6	46.0	71.3	51.7	54.0	46.0	36.7	32.3	30.4
6	54.7	298.0	54.7	314.0	67.3	317.5	74.6	306.0	67.3	294.5		
7	54.7	226.0	54.7	242.0	67.3	245.5	74.6	234.0	67.3	222.5		
8	54.7	154.0	54.7	170.0	67.3	173.5	74.6	162.0	67.3	150.5		
9	54.7	82.0	54.7	98.0	67.3	101.5	74.6	90.0	67.3	78.5		
10	54.7	10.0	54.7	26.0	67.3	29.5	74.6	18.0	67.3	6.5		
11	64.9	342.0	72.4	355.0	86.5	354.4	93.5	342.0	86.5	329.6	72.4	329.0
12	64.9	270.0	72.4	283.0	86.5	282.4	93.5	270.0	86.5	257.6	72.4	257.0
13	64.9	198.0	72.4	211.0	86.5	210.4	93.5	198.0	86.5	185.6	72.4	185.0
14	64.9	126.0	72.4	139.0	86.5	138.4	93.5	126.0	86.5	113.6	72.4	113.0
15	64.9	54.0	72.4	67.0	86.5	66.4	93.5	54.0	86.5	41.6	72.4	41.0
16	86.5	306.0	93.5	318.4	107.6	319.0	115.1	306.0	107.6	293.0	93.5	293.6
17	86.5	234.0	93.5	246.4	107.6	247.0	115.1	234.0	107.6	221.0	93.5	221.6
18	86.5	162.0	93.5	174.4	107.6	175.0	115.1	162.0	107.6	149.0	93.5	149.6
19	86.5	90.0	93.5	102.4	107.6	103.0	115.1	90.0	107.6	77.0	93.5	77.6
20	86.5	18.0	93.5	30.4	107.6	31.0	115.1	18.0	107.6	5.0	93.5	5.6
21	105.4	342.0	112.7	353.5	125.3	350.0	125.3	334.0	112.7	330.5		
22	105.4	270.0	112.7	281.5	125.3	278.0	125.3	262.0	112.7	258.5		
23	105.4	198.0	112.7	209.5	125.3	206.0	125.3	190.0	112.7	186.5		
24	105.4	126.0	112.7	137.5	125.3	134.0	125.3	118.0	112.7	114.5		
25	105.4	54.0	112.7	65.5	125.3	62.0	125.3	46.0	112.7	42.5		
26	128.3	306.0	134.0	323.3	147.7	329.6	156.9	306.0	147.7	282.4	134.0	288.7
27	128.3	234.0	134.0	251.3	147.7	257.6	156.9	234.0	147.7	210.4	134.0	216.7
28	128.3	162.0	134.0	179.3	147.7	185.6	156.9	162.0	147.7	138.4	134.0	144.7
29	128.3	90.0	134.0	107.3	147.7	113.6	156.9	90.0	147.7	66.4	134.0	72.7
30	128.3	18.0	134.0	35.3	147.7	41.6	156.9	18.0	147.7	354.4	134.0	0.7

Table 2.5: Mean angles for the NSCL High Rate Array Phoswiches

Detector	θ	ϕ	Detector	θ	ϕ	Detector	θ	ϕ
1	5.4	0.0	16	10.6	246.0	31	14.3	279.0
2	5.4	324.0	17	10.6	222.0	32	14.3	261.0
3	5.4	288.0	18	9.6	198.0	33	15.9	243.0
4	5.4	252.0	19	10.6	174.0	34	15.9	225.0
5	5.4	216.0	20	10.6	150.0	35	14.3	207.0
6	5.4	180.0	21	9.6	126.0	36	14.3	189.0
7	5.4	144.0	22	10.6	102.0	37	15.9	171.0
8	5.4	108.0	23	10.6	78.0	38	15.9	153.0
9	5.4	72.0	24	9.6	54.0	39	14.3	135.0
10	5.4	36.0	25	10.6	30.0	40	14.3	117.0
11	10.6	6.0	26	15.9	9.0	41	15.9	99.0
12	9.6	342.0	27	14.3	351.0	42	15.9	81.0
13	10.6	318.0	28	14.3	333.0	43	14.3	63.0
14	10.6	294.0	29	15.9	315.0	44	14.3	45.0
15	9.6	270.0	30	15.9	297.0	45	15.9	27.0

angle, defined as the angle between the beam axis and a line from the target to the center of the detector. The *azimuthal angle*, ϕ, is the angle between an arbitrary, fixed reference plane that includes the beam axis and the plane that passes through the beam axis and the center of the detector.

Table 2.5 gives the mean angles for the NSCL High Rate Array phoswiches (in degrees). The **Maryland Forward Array** (installed August 1992) consists of 16 triangular elements, all at a polar angle θ of 2.2°. A schematic view of the Maryland Forward Array (MFA) as it attaches to the frame of the High Rate Array is shown in Fig. 2.2. The azimuthal angles ϕ, relative to the reference used for the NSCL HRA are given in Table 2.6 (in degrees).

2.2.4 Energy Thresholds

Table 2.7 lists punch-in energies (i.e. the minimum energy required to enter the medium) in MeV of various particle types entering the E element of a *Ball* phoswich. Next, in Table 2.8 we give the punch-in energies in MeV for various particle types entering the E element of a

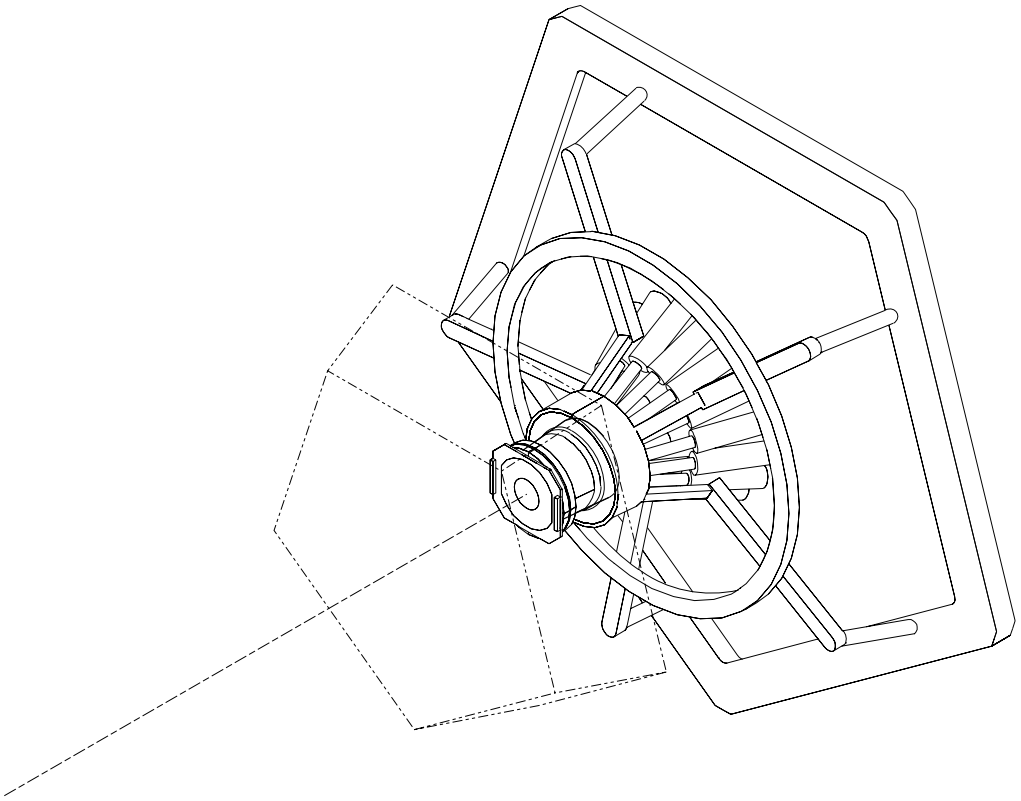


Figure 2.2: Schematic view of the Maryland Forward Array (MFA) as it attaches to the frame of the High Rate Array.

Table 2.6: Azimuthal Angles for the Maryland Forward Array

Detector	ϕ	Detector	ϕ	Detector	ϕ	Detector	ϕ
1	162.0	5	252.0	9	342.0	13	72.0
2	184.5	6	274.5	10	4.5	14	94.5
3	207.0	7	297.0	11	27.0	15	117.0
4	229.5	8	319.5	12	49.5	16	139.5

Table 2.7: Energy Thresholds for the Ball Phoswiches

Particle Type	Punch-in Energy (MeV)	Particle Type	Punch-in Energy (MeV)
p	17	Be	214
d	24	B	300
t	28	C	380
He	70	N	483
Li	140	O	593

Table 2.8: Energy Thresholds for the NSCL High Rate Array Phoswiches

Particle Type	Punch-in Energy (MeV)	Particle Type	Punch-in Energy (MeV)	Particle Type	Punch-in Energy (MeV)
p	13	C	269	Al	877
d	17	N	341	Si	962
t	20	O	419	P	1079
He	50	F	515	S	1170
Li	99	Ne	591	Cl	1294
Be	152	Na	687	Ar	1455
B	212	Mg	767		

NSCL *High Rate Array* phoswich. The 4.5 kÅ-thick (0.12 mg/cm^2) Al layer evaporated onto the front face the HRA is negligible in the determination of these energies. Finally, in Table 2.9 we list the punch-in energies in MeV for various particle types entering the E element of a *Maryland Forward Array* phoswich (without the annular Si detector present).

2.3 Maryland Forward Array

The Maryland Forward Array is designed for the study of peripheral intermediate-energy heavy-ion collisions. Its low angle coverage provides for the detection of relatively heavy projectile-like fragments normally missed by the rest of the 4π array.

Table 2.9: Energy Thresholds for the Maryland Forward Array Phoswiches

Particle Type	Punch-in Energy (MeV)	Particle Type	Punch-in Energy (MeV)	Particle Type	Punch-in Energy (MeV)
p	10	C	200	Al	642
d	12	N	253	Si	704
t	15	O	311	P	790
He	37	F	381	S	859
Li	74	Ne	438	Cl	954
Be	113	Na	505	Ar	1075
B	157	Mg	563		

The Maryland Forward array is shown schematically in Fig. 2.2. It consists of an annular Si detector backed by a plastic phoswich array. The Si detector, manufactured by Micron Semiconductor, has a thickness of 300 μm, an active inner diameter of 4.8 cm, and an outer diameter of 9.6 cm. The Si ring is segmented into 16 on the back side and into 16 concentric rings on the front (not shown in the figure). The plastic phoswich array consists of 16 individual detectors. Each phoswich is made of a 1-mm thick fast plastic scintillator (BC400) glued to a 10-cm thick slow plastic scintillator (BC444), and the Hamamatsu R1924 photomultiplier tube is optically coupled to the scintillator through a light guide. The 16 phoswich detectors are arranged in a cone-like geometry, with the front of the cone mating with the back of the Si ring. The front of the Si detector is designed to be positioned 94.5 cm from the target. The entire array covers an azimuthal angular range of 1.5° to 2.9°.

2.4 Zero Degree Detector

"Everybody must get Stone-d..."
- Bob Dylan

2.4.1 Motivation

The "Zero Degree Detector" (ZDD) is a part of the MSU 4π Array, which covers the laboratory polar angles between 0.56° and 1.55° . It consists of two layers of fast scintillating plastic, and can resolve fragment charges and energies via two methods. In the first, fragments which pass through the first thin layer are identified via ΔE/E measurements. In the second, slower fragments which stop in the first layer can be identified via ΔE/Time-of-Flight(TOF) measurements.

There are several motivations for building such a detector. The mere fact of its small-angle coverage is valuable for identifying projectile-like fragments, and there is also the increased angular coverage. More importantly, it is anticipated that the data from this detector will provide valuable information relating to the impact parameter of triggered events. Simulations have been performed using the code FREESCO, and the results (Fig. 2.3, bottom right) predict a nearly monotonic relation between the charges of fragments emitted in this region and the impact parameter of the collision, for $Z \gtrsim 3$. At low incident energies (Fig. 2.3, bottom left), evaporation residues survive the collision, and the charges of such residues also show a monotonic relation to the impact parameter. These residues can be distinguished from other fragments by their TOF. This information is unaffected by the dynamics of the collision, and hence provides a means of selection of impact parameter which is alternative to *e.g.* centrality cuts on global observables like the charged particle multiplicity and the total transverse kinetic energy.

2.4.2 Construction Details

The phoswich detector technique is already extensively used in the 4π Array. The implementation of this design is, however, slightly different from that of the rest of the Array. The primary difference in this application is the optical isolation between the thick and thin plastic layers. The reason for this is to allow each layer to be made from equally fast scintillating plastic, in contrast to the fast and slow scintillating layers in phoswich applications. In addition, the ZDD is placed much further from the target (2.74 meters), to improve timing resolution.

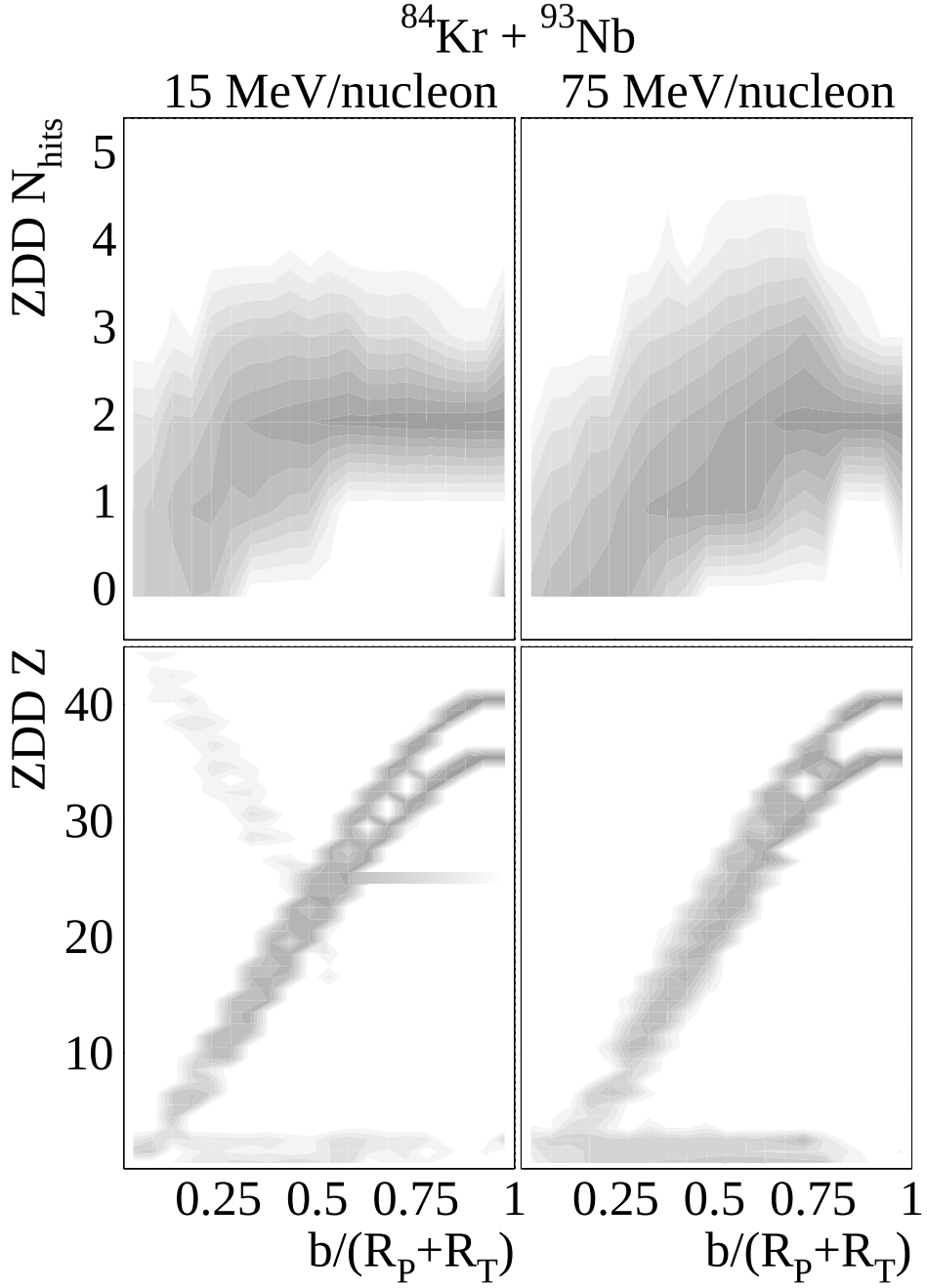


Figure 2.3: Results of FREESCO model calculations for 15 and 75 MeV/nucleon beams.

The ZDD consists of two layers of fast scintillating plastic (Bicron's BC404), which was chosen for its fast decay time (1.8 ns) and high light output (14% more light than BC412, which is used in the rest of the 4π ball). The first (ΔE) layer of the detector is 0.67 mm thick, allowing transmission of particles with Z $\leq$ 18 for KE/A $\gtrsim$ 21 MeV. The actual punchin energies are tabulated below in table 2.10. The second (E) layer is 30 cm thick, stopping all particles with KE/A $<$ 220 MeV. These two layers are optically isolated by a 4.5 kÅ

Z	Punchin energy (MeV) / (AMeV)	Z	Punchin energy (MeV) / (AMeV)	Z	Punchin energy (MeV) / (AMeV)
1 (p)	7.2 / 7.2	13	504 / 18.7	25	1375 / 25.0
2	29.5 / 7.4	14	555 / 19.8	26	1450 / 25.9
3	57.5 / 8.2	15	625 / 20.2	27	1535 / 26.0
4	90 / 10.0	16	685 / 21.4	28	1590 / 27.4
5	119 / 11.8	17	756 / 21.6	29	1695 / 26.9
6	158 / 13.2	18	845 / 21.1	30	1770 / 27.6
7	201 / 14.4	19	895 / 22.9	31	1880 / 27.2
8	248 / 15.5	20	952 / 23.8	32	1995 / 27.0
9	304 / 16.0	21	1055 / 23.4	33	2075 / 27.7
10	347 / 17.3	22	1135 / 23.6	34	2190 / 27.4
11	396 / 17.2	23	1220 / 23.9	35	2250 / 28.5
12	440 / 18.3	24	1290 / 24.8	36	2360 / 28.1

Table 2.10: Punchin energies for the ZDD, based on 0.67mm thick scintillating plastic.

aluminum coating, evaporated onto the front of the thick plastic, and a 1.5 micron mylar film stretched onto the back of the thin plastic. The detector is divided into 8 azimuthally symmetric segments, which is necessary to avoid multiple hits. As seen in Fig. 2.3 (top frames), the fragment population of the ZDD is anticipated to be routinely as high as 2, and can be as high as 4 particles per event. Fragments entering this region often result from the same source, closely correlating them in phase space.

The ΔE segments have been fitted with adiabatic UVT light guides, which are optically coupled to Phillips XP2262B photomultiplier tubes (PMTs)– moderately fast, reliable PMTs which are widely used in high-speed physics applications. The PMTs operate in the negative HV mode for high speed and pulse fidelity, and are surrounded by μ-metal shields, to isolate

them from electromagnetic interference. Below (Fig. 2.4 and 2.5) are the circuit diagrams for the PMT sockets and HV divider cards, respectively.

The thick plastic is directly coupled to the PMTs at the back of the detector (Fig. 2.6). As illustrated, there is a small hole in the ZDD (2.7 cm radius, corresponding at 2.74 m to $0^{\circ} \leq \Theta \leq 0.56^{\circ}$) which minimizes the likelihood of extraneous hits from non-interacting beam particles. Its size was determined by considering typical emittances for beams extracted in the 4π vault. The outer active radius of the detector, defined by the entrance flange on the ZDD chamber, is 7.4 cm. However, the whole face extends out to 12.0 cm to allow for full coverage of the available solid angle while permitting lateral motion over a limited range. At 2.74 meters from the target, the ZDD has a total solid angle coverage of 1.9 msr (0.24 msr per telescope). The mean polar angle of each ZDD segments (they are azimuthally symmetric) is thus 1.1° , and the azimuthal angles are listed in Table 2.11 below.

Detector	ϕ	Detector	ϕ
1	338.3	5	158.3
2	293.3	6	113.3
3	248.3	7	68.3
4	203.3	8	23.3

Table 2.11: Azimuthal angles for the ZDD. †

†**Note:** Since the ZDD chamber is separate from the 4π ball, if the ball is rotated, these angles will change accordingly. These angles are reported assuming a 3.7° tilt (clockwise, looking down the beam pipe) of the 4π ball.

The length of the flight path was chosen to maximize the quality of the timing resolution. This is important when identifying fragments via the ΔE/TOF method. In this method , a very reliable timing signal is needed, and an absolute time signal would be ideal. Currently, the "cleanest" source of an absolute time signal (start detectors can introduce background) is the cyclotron RF signal. The width of a typical RF "bucket" is nominally 40° , in a 360° cycle. The cyclotron phase slits can be used, however, to tune this to as low as 4° with some effort. For the ^{84}Kr beams extracted from the K1200 cyclotron at 25 (95) AMeV, such a width in RF corresponds to a timing width of $\sim$ 1.1 (0.4) nS per bucket.

Trimming the RF buckets in this way improves RF timing, but also carries a cost of some intensity; such losses may range between factors of 3 and 30. However, even with such intensity losses, it is expected that the cyclotron, fed by the super-conducting ion source, can still provide sufficient beam intensities ($\gtrsim$5 p pA) for a wide range of extracted beams.

Figure 2.4: PMT socket circuit diagram.

Figure 2.5: High Voltage divider card circuit diagram.

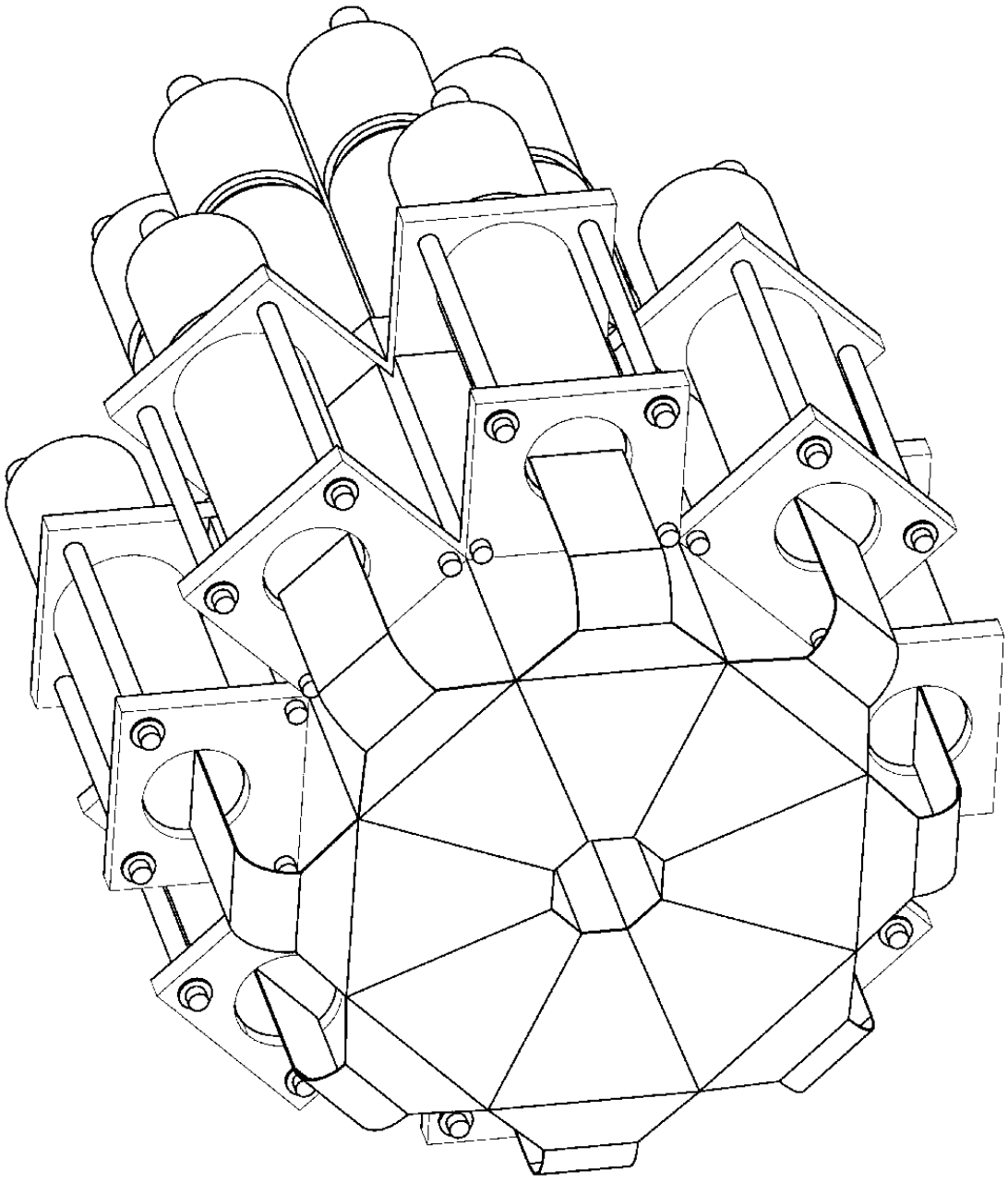


Figure 2.6: Diagram of the ZDD, showing placement of PMTs and light guides. The beam enters from the lower right.

2.4.3 Vacuum Chamber and Procedures

The long flight path for this application required the addition of a separate vacuum chamber. The ZDD chamber design (Fig. 2.7) includes a goniometer, a simple device which allows for lateral motion of the detector within the chamber. The goniometer will be used to center the detector around the beam, in cases where the alignment is poor. Misalignments of the beam have been empirically observed to be as large as 0.6^{o} . This chamber can also be used as a free-standing, independently functional test-chamber. The arrival of this chamber has necessitated minor, but reversible, revisions to the existing beam line behind the 4π Array. The new beamline configuration is shown in Fig. 2.8.

With the advent of this new vacuum chamber to the 4π array comes additional vacuum pumps and pumping procedures. The ZDD chamber cart has been designed to allow convenient location of all necessary additions that come with such pumps. A turbo pump controller, a thermocouple meter, and an ion gauge readers are among those additions. Furthermore, the chamber itself has a KF25 fitting mounted on its underside which generally supports a trio of vacuum gauges (an ion gauge, a thermocouple, and a mechanical "Bourdon" gauge), and a vent valve.

The pumps required for this chamber are an 8 inch turbo pump (typically an Alcatel model 450, which is water-cooled) and an appropriate fore pump. The turbo pump should be mounted to the chamber using an insulating ISO-160 coupler and plastic clamps, so that the electrical noise on the pumps does not get passed to the signal cables. For this same reason, the thermocouple reader should be unplugged during normal running conditions. A valve should also be placed between the fore pump and the turbo pump, for the inevitable emergency shut down (*i.e.* power failure), although it is also be used during standard pumping procedures. And, finally, a 110V power strip is (usually) mounted on the cart frame, to allow convenient access to the power for all devices.

To Pump Down the chamber

- First **VERIFY** the following:
  - All vacuum flanges are on securely.
  - Vent valves (one beneath the chamber, and one at the beam dump) are closed.
  - Cooling water supply to the turbo pump is running (the casing of that pump should feel cold to the touch).
  - 4π Exit gate valve is closed.
  - The bias voltages are off.
  - All devices are plugged into the power strip.
- **CLOSE** the valve between the fore pump and turbo pump.

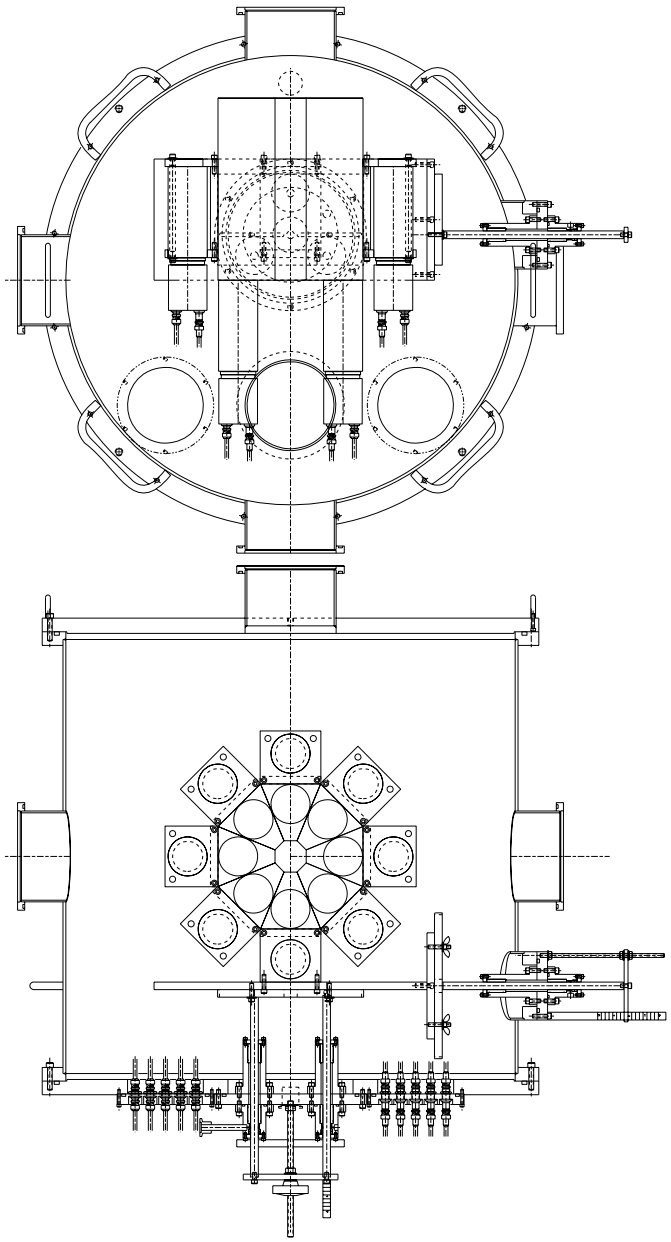
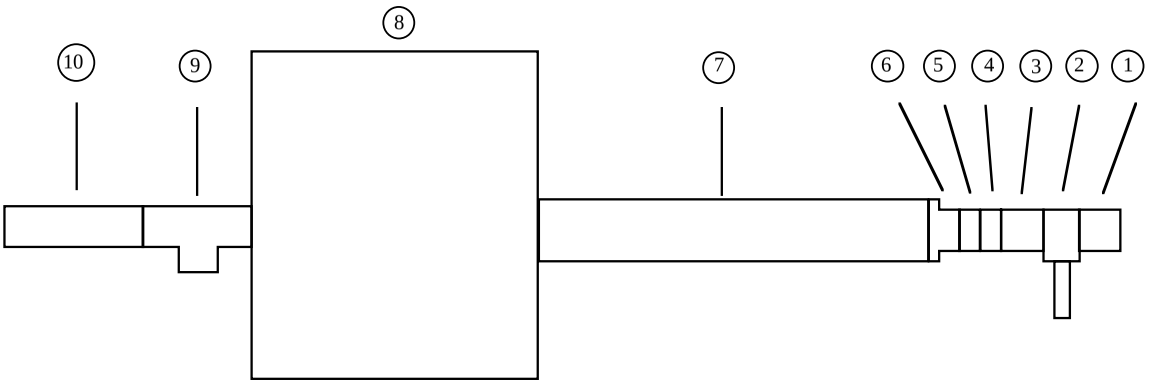


Figure 2.7: Mechanical drawing of the ZDD chamber. Above: Top view, with beam entering from the top of the page. Below: Rear view, looking upstream. This also shows the goniometer, used for positioning of the detector.



Component List

1	Custom / ASA-6 adapter	6	ISO-100 / ISO-160 Reducer
2	Gate Valve	7	6" Beam Tube
3	ASA-6 / Marmon adapter	8	Vacuum Chamber
4	Insulating Marmon ring	9	4" T adapter
5	Marmon / ISO-100 Adapter	10	Bellows

Figure 2.8: Scale diagram of the new exit beam line, with a description of components.

• **TURN ON** the power strip. This will start the fore pump, and power up all other devices.

• Slowly **OPEN** the valve to the turbo pump (open it all the way in 5 to 10 seconds). The chamber should now begin its descent fast enough that the acute observer can see the needle on the mechanical gauge moving.

• Once the thermocouple gauge reads 100 to 200 mT, **TURN ON** the turbo pump. Watch the turbo pump for at least 10 minutes (or its timeout period) to be sure that it stays on.

• When the pressure in the chamber reaches ion gauge range (typically within one hour) **UNPLUG** the thermocouple gauge. This should be the last electrical connection between dirty power and the chamber body (in the standard configuration).

This whole process will typically bring the ZDD chamber into the 10^{-5}T range within 2 hours after beginning, although it has been done in less time.

In Case of Fire: Vent the chamber by following the bullets below.

• First **VERIFY** the following:

- The 4π exit gate valve is closed.
- The voltages are off.

• **CLOSE** the valve between the fore pump and turbo pump; this will ensure that no oil from the fore pump is drawn into the chamber while it is venting. As an aside, for the same reason, the valve should be closed in the event of a power failure, even if you do not vent the chamber.

• **TURN OFF** the power strip; this will, obviously, shut off all pumps and gauges.

• **OPEN** the vent valve (located beneath the chamber, nearest to the 4π ball) slowly, just enough to hear air seeping into the chamber. Rapid air flow could inflate the wrapping on the light-guides, damaging the optical isolation.

Once the mechanical gauge shows a discernable change (five minutes), the user can open this valve more, to allow a quicker ascent to air pressure. The whole venting procedure can take between 15 and 30 minutes.

2.4.4 User Warnings

The ZDD is designed to cover a sensitive part of the overall solid angle. In order to achieve this task, some materials and electronics had to be pushed to their limits in the implementation of this detector array. One such example is the high voltage divider cards which accompany the PMTs. The PMTs are 12-stage tubes with a maximum voltage of -2500 V.

Care should be taken, as with any PMT, not to exceed this voltage. In prototyping the design for the ZDD, it was empirically observed that the maximum event rate for a PMT is almost directly proportional to the standing current in the voltage divider, and monotonically related to the number of active stages in the divider. For this reason, the standing current in the HV divider cards was set to 2 mA, reasonably high, yet safely below the current limit of the LeCroy 1440 mainframe power supply. With this high current comes high power dissipation.

This is your **first** warning: Care has been taken to provide adequate heat sinks for the HV divider cards by lining each one with an aluminum-oxide impregnated silicone rubber sheet, which is supported on its opposite side by an aluminum sheet inside the extruded HV card box. This allows the heat to be drawn off the circuit boards and out to the extruded boxes. The user must ensure that these boxes are thermally attached to the chamber, thus allowing the dissipation of this heat away from the boxes. If this condition is not met, loss of some HV cards may likely result, and has in the past.

This is your **second** warning: Unplug the thermocouple gauge once the ion gauge has been turned on. This is done in order to ensure separation of the chamber body (which is connected to the thermocouple gauge) from any and all sources of dirty power (which is connected to the thermocouple reader).

2.4.5 DAQ/Trigger Electronics Details

Following is a description of the DAQ electronics and implementation specific to the ZDD. For more in-depth descriptions of the how's and how-to's of 4π DAQ, the reader should read the appropriate chapters in the remainder of the User's Guide.

Traditional 4π phoswich ΔE-E signals are transmitted together along the HV cables and, therefore, must be split in three ways (ΔE/E/T), in addition to passing through an effective high-pass filter, which isolates the high voltage. The ZDD segments, as aforementioned, are optically isolated, and the signals are separated at the PMT base from the HV cables. However, the signals from these detectors must still be split (ΔE/T and E/T) to obtain a time signal. Currently (June 1995) this is being done with a 4π, non-HV, NIM splitter module similar to the ones used for the MFA (which also has separated HV and signal cables). In order to preserve good cable termination, one of the outputs of this splitter (the "E" ribbon output) must be terminated with 100Ω. This is easily accomplished with a block-connector fitted with 100Ω resistors.

Since the ΔE and E signals are inherently separate, they also have separate times. The 16 "Time" (LEMO) outputs from the splitter have been divided into ΔE and E bundles (8 cables each), which currently (June 1995) go into separate LeCroy Discriminators. Since a ZDD

trigger, and the ZDD.Live signal, is generated from the SUM output of the discriminators, this separation allows the experimenter to easily choose between triggering on any ZDD element (*e.g.* **any** ΔE or E) and triggering on ΔE's or E's only. At this point, the division ends, and the ECL outputs of these two discriminators are joined back onto one cable via a split-ended ribbon cable, so that all ZDD TFC and Scaler inputs are on one ribbon cable, This restores the signals to one bank (and one FERA).

The time signals should come, if the experimenter has set up the time gate "properly," within the standard 4π time gate, but the ΔE/E signals (which occupy the top/bottom of one FERA) will, of course, need a gate of their own. The ZDD gate (one only) is the output of a QDGG triggered by the ZDD.Live signal, and the ZDD.Live is simply the .AND. of the ZDD single (from the "singles" octal CFD) and MASTER.Live . Below are listed (Table 2.12) some critical settings for the ZDD trigger electronics as used for experiment # 93033 (the ZDD's maiden voyage).

Characteristic	Setting (nS)
Singles delay (DGG X out)	12.8
Singles width (DGG Y out)	22.0 (minimum setting)
ΔE/E delay (DGG X out)	15.0
ΔE/E width (DGG Y out)	280.0

Table 2.12: Critical settings for the ZDD trigger logic circuit.

2.4.6 Response Function

During the first ZDD experiment (# 93033), a fragmentation calibration beam was obtained by colliding a ^{40}Ar beam on a ^{9}Be target in the A1200 particle separator. That secondary beam provided 51 distinct data points for the calibration of this detector (Fig. 2.9). The magnet setting used to obtain these points was Bρ = 1.9744 Tm (B = 0.63607 T) . These points, in conjunction with the punch-in points and several data points at 95 AMeV, provided an excellent basis for the extraction of the response function of this detector. Two different forms of the response function were considered: the first is the form explored by Cebra[1], and the second was a derivative of Birks' energy loss relations, as reported by Pastor[2]. Following are the forms and parameterizations of each:

Cebra:

$$\Delta L = 0.7 \Delta E^{0.519} Z^{0.279} A^{0.488} \tag{2.1}$$

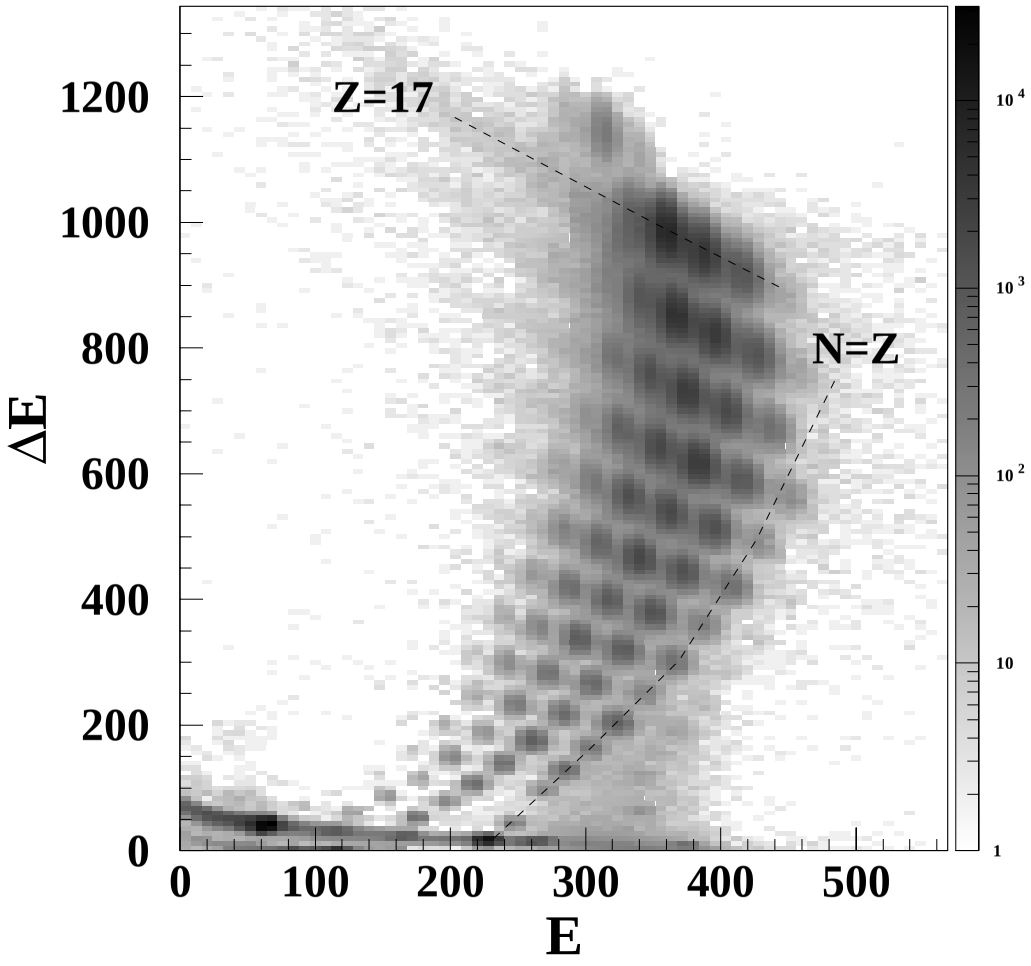


Figure 2.9: Calibration data from the A1200 fragmentation beam, as recorded in segment 2 of the ZDD.

$$L = 0.2E^{1.118}Z^{0.106}A^{-0.597} \tag{2.2}$$

Birks:

$$\Delta L = 1.059(\Delta E - C\ln(\frac{C + (\Delta E + E)}{C + E})) \tag{2.3}$$

$$L = 0.205(E - C\ln(1 + \frac{E}{C}) \tag{2.4}$$

$$C = 4.700A^{0.0}Z^{1.735} \tag{2.5}$$

The calibration points were used as the input to an IMSL, χ^2 fitting routine, via which the best values for the response parameters were identified. The numerical values listed in the above equations represent the results of that fitting process. The Cebra parameterization (eq. 2.1 and 2.2) was empirically observed to reproduce the curvature of the response lines better than the Birks formulation. For this reason, the Cebra parameterization was used to create the energy tables for the ZDD.

The current (June 1995) template (*i.e.* ZPID file) contains the charges from Z=2 to Z=27, inclusive. Protons are not well measured in the ZDD due to the small thickness of the ΔE plastic.

References

[1] D. Cebra *et al.*, Nucl. Inst. Meth. **A313** , 367 (1192).

[2] C. Pastor *et al.*, Nucl. Inst. Meth. **A212**, 209-215 (1983).

2.5 Bragg Curve Counters

2.5.1 Principle of Operation

A Bragg curve counter (BCC) is basically an ionization chamber with its field parallel to the incoming particles. In this way, the range of particles which are stopped within the detector can be measured. The BCC takes advantage of the fact that the maximum specific ionization of a stopping ion is proportional to the atomic number of the particle. By measuring the maximum of the ionization process, one obtains the charge Z of the particle. The integral of the ionization is a measure of the energy E.

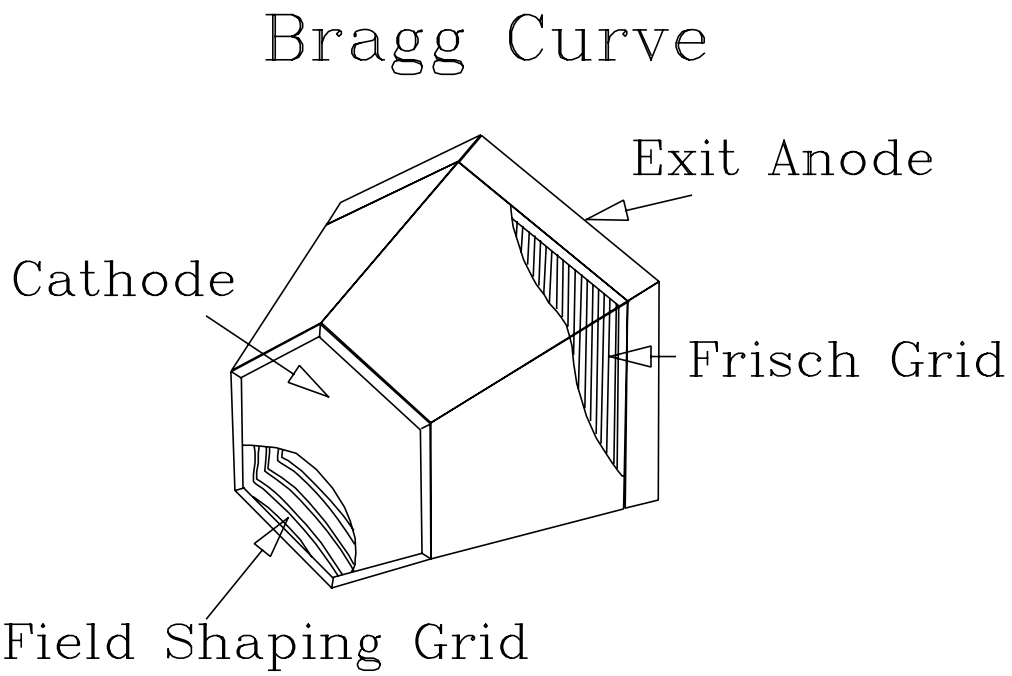


Figure 2.10: Schematic drawing of a Bragg Curve Counter. Cutaways show the Frisch grid and field shaping grid.

The electrons liberated by the ionization drift to an anode which is shielded from induced charge by a Frisch grid. Since the electric field is parallel to the path of the particle, the charge is measured as a function of time, and thus the complete energy-loss distribution of the stopping ion may be obtained.

2.5.2 Constructional Details of the 4π BCC's

A schematic view of a 4π BCC module is shown in Fig. 2.10. The overall length of the counter is 15 cm and the entrance window for a hexagon module is 10 cm across from side to side. Because of the large dimensions, it was important to include radial field shaping strips on the inside surfaces of the walls. The walls are made from 6.4 mm thick G10 fiberglass-epoxy laminate. The pressure window is a 900 μg/cm^2 Kapton foil. This is supported by a grid of wires, 0.024 inches in diameter and of approximately 1 cm spacing. The fraction of the window area covered by this grid is roughly 11%. The separation (drift space) between cathode and Frisch grid is 14 cm. The Frisch grid is made from 12.5 μm gold-plated tungsten wires with 0.5 mm spacing. The back plane of the BCC is the front surface of the scintillator

telescope, which lies 1 cm behind the Frisch grid. This also forms the anode of the BCC, since the surface has a 25 kÅ-thick aluminum layer evaporated onto it.

The Bragg Curve Counters for the most forward hexagons (1 through 5) are modified from the original design in that the anodes are split into six triangles to match the phoswiches behind them. Separate signals are taken from each of the split anodes and these are processed by individual preamplifiers and amplifiers. This modification of the forward Bragg Curve Counters gives both higher angular granularity and also reduces the counting rate in the most forward channels.

2.5.3 Operation of the 4π BCC's

The 4π BCC was designed to operate at 500 torr of P5 gas (95% argon, 5% methane). More recently, C_2F_6 gas (heavy freon) has been used at 100 to 150 torr with favorable results. The operating voltages for P5 gas are -1000 V on the cathode and +120 V on the anode. For C_2F_6, the cathode is run at -500 V and the anode at +100 V. The voltages have a pronounced effect on the signal shape and therefore the ability to measure the Z of the ion. The Bragg peak signal from the anode is split and analyzed with separate amplifiers having short and long shaping times. The short shaping time (differentiation) will give a measure of the peak amplitude, while from the long shaping time (integration) one obtains the total energy.

Heavy ions with as little as 100 keV/nucleon will penetrate the BCC entrance foil, but in order to give a usable signal, a good fraction of the Bragg peak must occur within the chamber. Thus the lower identification threshold is about 1 MeV/nucleon for heavy ions. Heavy ions in the range A = 20-30 with energies above about 7 MeV/nucleon will "punch-through" the BCC into the backing scintillator. This, of course, will be a function of the gas and the gas pressure. If the ions stop in the fast plastic scintillator, they may be identified by a plot of the energy lost in the BCC against the light output from the fast plastic (a ΔE-E plot).

2.5.4 BCC References

Below are listed some BCC references:

- C.R. Gruhn *et al.*, Nucl. Instrum. Meth. **196**, 33 (1982).
- D.A. Cebra *et al.*, MSU-NSCL Annual Report, 95 (1989), and Nucl. Instrum. Meth., **A300**, 518 (1991).

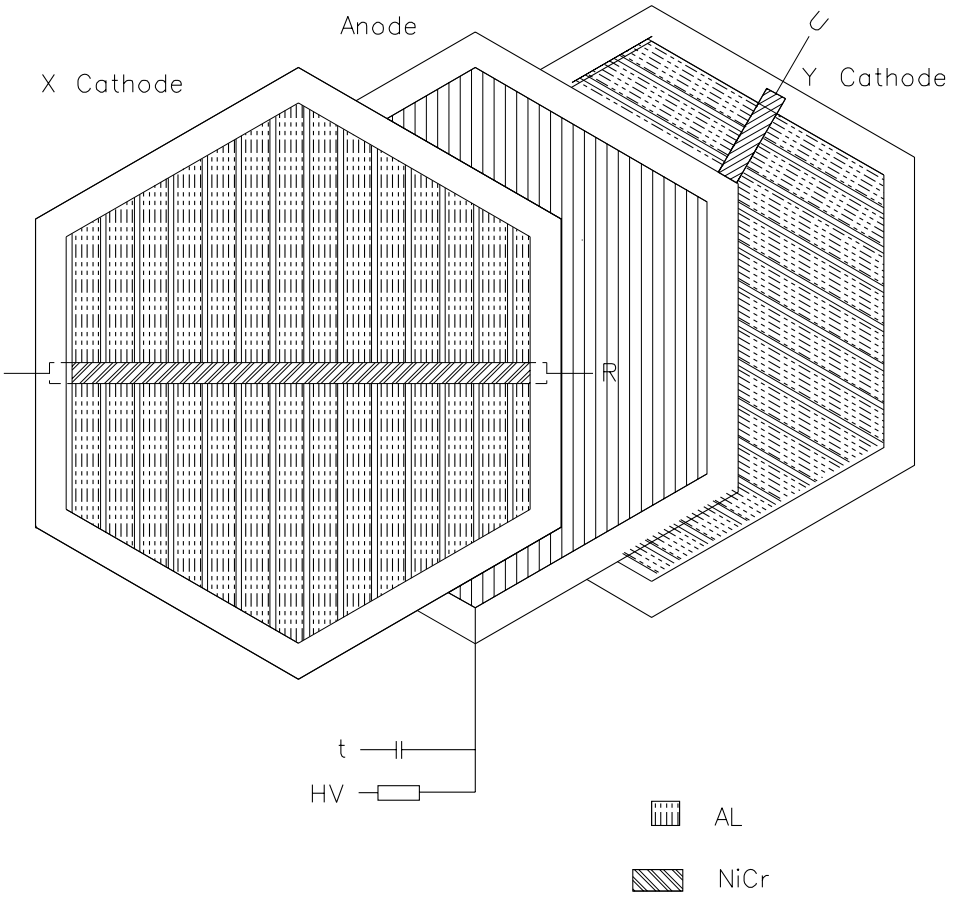


Figure 2.11: Exploded view of a hexagonal Multiwire Proportional Counter.

2.6 Multiwire Proportional Counters

2.6.1 Principle of Operation

The Multiwire Proportional Counter (MWPC) is a low pressure gas detector which makes use of the double amplification process discovered by Breskin. It is essentially a development of the Parallel Plate Avalance Counter (PPAC) by placing a plane of anode wires between two cathode foils. Detectors of this type form the inner layer of the 4π array. They are mounted off the 4π Bragg Curve Counters.

2.6.2 Constructional Details of the 4π MWPC's

A schematic view of a 4π MWPC module is shown in Fig. 2.11. The total thickness of the counter is 1.6 cm. The distance from target to the center of the MWPC is 15 cm; the solid angle subtended is about 300 msr. The anode is a plane of gold plated tungsten wires, 12 μm in diameter, in the center of the detector. The wire spacing is 1 mm. These wires are

connected to the high voltage and provide the fast timing signal. The two cathodes are at ground potential and are located 3.2 mm either side of the anode plane. Each cathode plane consists of a polypropylene foil stretched to about 75 μg/cm^2. The cathode foils are coated with a resistive strip of Nichrome (Ni/Cr 80/20%). The Nichrome strip is in turn coated with 5 mm-wide Al strips, perpendicular to the Nichrome strip. The resistance between the end contacts of the cathode foils is typically in the range of 4 to 10 kΩ. For the hexagonal MWPC's, the orientation of one striped cathode (say the X) is 60° with respect to the second (Y). From each cathode, the position is calculated by the data acquisition software with the charge division method; *viz.* $X = L / (L + R)$ and $Y = U / (U + D)$, where L, R, U, and D stand for the Left, Right, Up, and Down signals. Note that, because the stripes on the two foils are not perpendicular to each other, these pairs of signals do not give positions in Cartesian co-ordinates as the names may suggest. The pressure windows are 0.3 mil (7.6 μm) Kapton foils.

2.6.3 Operation of the 4π MWPC's

The 4π MWPC is designed to operate at 5 torr of isobutane. A typical anode operating voltage is 510 V. At 5 torr pressure, the breakdown voltage is about 530 V.

2.6.4 MWPC References

- A. Breskin, Nucl. Instrum. Meth. **196**, 11 (1982).
- G.D. Westfall *et al.*, MSU-NSCL Annual Report, 1983-4, p. 186.

2.7 Module Numbering

The modules in the 4π **BALL** are numbered in the following manner (going from the most forward angles to the most backward, each 'circle' being concentric with the beam axis):

- **1 to 5 – first circle (forward hexagons)**
  The numbers increase in a clockwise direction around the beam axis, as viewed from the beam exit. (This convention applies to all the circles.)
- **6 to 10 – second circle (forward pentagons)**
  Module 6 is adjacent to modules 1 and 2 in the first circle, and to modules 11, 12, and 13 of the third (middle) circle.

- **11 to 15 – third circle (middle front hexagons)**
  These are labeled such that module 11 is next to module 1 of the forward circle, 12 is next to 2, *etc.*.

- **16 to 20 – fourth circle (middle back hexagons)**
  These are labeled such that module 16 is next to module 6 of the forward pentagons, 17 is next to 7, *etc.*.

- **21 to 25 – fifth circle (back pentagons)**
  Module 21 is adjacent to module 11 of the middle forward hexagons, 22 is adjacent to 12, *etc.*.

- **26 to 30 – sixth circle (back hexagons)**
  Module 26 is adjacent to module 16 of the fourth circle, 27 adjacent to 17, *etc.*.

The ball numbering convention is illustrated in Figs. 2.12and 2.13.

The NSCL **High Rate Array** detectors are arranged in three pentagonal rings, and are numbered in the following manner.

- **1 to 10 – first (inner) pentagon**
  These are the most forward angle detectors. Detector 1 lies in the wedge adjacent to ball module 1. The other detectors are arranged in a counterclockwise direction (as viewed from the target) around the pentagon.

  11 to 25 – second (middle) pentagon
  Detector 11 lies adjacent to detector 1 in the inner pentagon. The other detectors are arranged in a counterclockwise direction around the pentagon.

  26 to 45 – third (outer) pentagon
  Detector 26 lies adjacent to detector 11 in the middle pentagon. The other detectors are arranged in a counterclockwise direction around the pentagon.

The arrangement of the High Rate Array detectors is shown in Fig. 2.14.

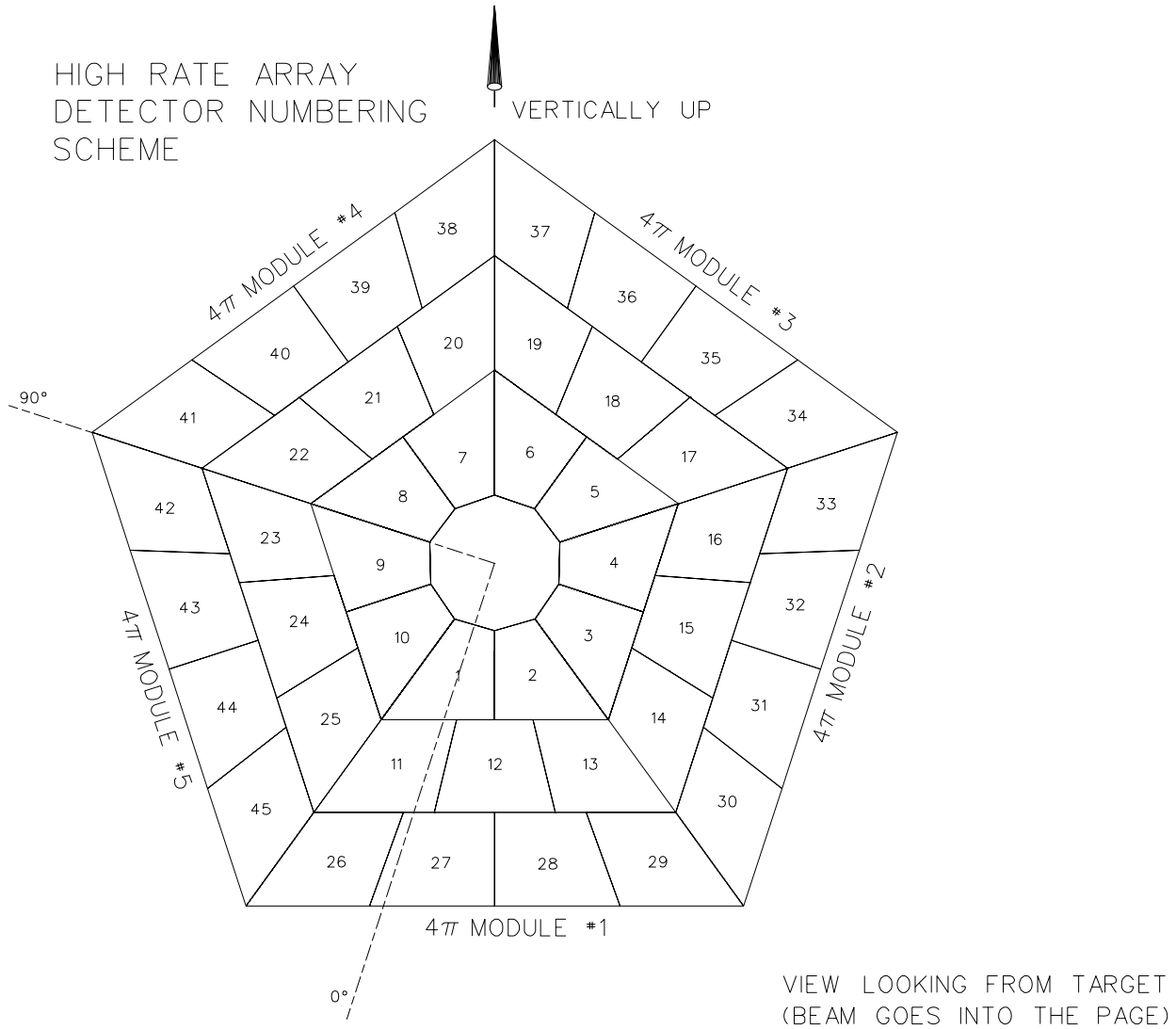


Figure 2.14: Arrangement of the High Rate Array detectors in the exit weldment of the ball.

2.7.1 Rings

It is often convenient to group detectors that have the same θ angle. For this purpose, the concept of "rings" has been developed. Each ring contains 5 or 10 detectors, all with the same θ. All detectors in a given ring should count at a similar rate and show similar features. The arrangement in rings of the phoswich detectors in the 4π Array is given in Tables 2.10 and 2.11.

2.8 Module Orientations

The individual modules are positioned in the 4π ball so the 'A' detector is closest to the *downstream* end (where *downstream* refers to the end of the ball with modules 1 to 5.) The only exception is the first ring of pentagons (modules 6 to 10) where two sides are equally close to the *downstream* end. For these modules the *top right* side is defined to be the "A" detector (as viewed from the side with *downstream* on the right.)

After the installation of the gas lines on the outside of the ball it became necessary to standardize the orientation of the backplate with respect to the plastic scintillator. It was found that four different orientations would allow the minimum amount of RTV tubing needed to connect the gas lines on the ball to the gas feedthrough valves on the modules. Sketches of these four orientations can be found in the 4π vault. A brief description follows...

- **Orientation HEX1**
  Hex module with gas valve feedthrough plate between detectors D and E. First two hex rings: Modules 1–5, and 11–15.

- **Orientation HEX2**
  Hex module with gas valve feedthrough plate between detectors A and F. Last two hex rings: Modules 16–20, and 26–30.

- **Orientation PENT1**
  Pent module with gas valve feedthrough plate between detectors A and B. First pent ring: Modules 6–10.

- **Orientation PENT2**
  Pent module with gas valve feedthrough plate between detectors C and D. Second pent ring: Modules 21–25.

Table 2.13: Arrangement of ball modules in rings

Ring										
1	1A	2A	3A	4A	5A					
2	1B	2B	3B	4B	5B	1F	2F	3F	4F	5F
3	1C	2C	3C	4C	5C	1E	2E	3E	4E	5E
4	1D	2D	3D	4D	5D					
5	6A	7A	8A	9A	10A	6B	7B	8B	9B	10B
6	11A	12A	13A	14A	15A					
7	6C	7C	8C	9C	10C	6E	7E	8E	9E	10E
8	11B	12B	13B	14B	15B	11F	12F	13F	14F	15F
9	6D	7D	8D	9D	10D					
10 †	11C	12C	13C	14C	15C	11E	12E	13E	14E	15E
	16A	17A	18A	19A	20A					
11 ††	16B	17B	18B	19B	20B	16F	17F	18F	19F	20F
	11D	12D	13D	14D	15D					
12	21A	22A	23A	24A	25A					
13	16C	17C	18C	19C	20C	16E	17E	18E	19E	20E
14	21B	22B	23B	24B	25B	21E	22E	23E	24E	25E
15	16D	17D	18D	19D	20D					
16	21C	22C	23C	24C	25C	21D	22D	23D	24D	25D
17	26A	27A	28A	29A	30A					
18	26B	27B	28B	29B	30B	26F	27F	28F	29F	30F
19	26C	27C	28C	29C	30C	26E	27E	28E	29E	30E
20	26D	27D	28D	29D	30D					

†formerly rings 10 and 11
††formerly rings 12 and 13

Table 2.14: Arrangement of High Rate Array detectors in rings

Ring										
23	1	2	3	4	5	6	7	8	9	10
24	12	15	18	21	24					
25	11	13	14	16	17	19	20	22	23	25
26	27	28	31	32	35	36	39	40	43	44
27	26	29	30	33	34	37	38	41	42	45

It should be noted that the orientation of the gas valve feedthrough plate with respect to the backplate is also unique for each module orientation.

2.9 Cabling

This section details the length, type, and number of cables that allow the 4π to talk to the world outside its vault.

- **Main Ball Phoswich Detector Cables**
  Each of the 170 main ball phoswich detectors receives its high voltage and transmits its signal over the same cable. Inside the vacuum chamber this is a two foot section of RG-59 that runs from the PMT base to the SHV vacuum feed-thru. Outside the ball a 56 foot long RG-59 cable runs from the SHV feedthru on the ball to the splitter boxes in the electronic racks.

- **High Rate Array Phoswich Detector Cables**
  The 45 High Rate Array (HRA) detectors also receive their high voltage and transmit their signals over the same cables. Inside the vacuum chamber there are 5 foot RG-174 cables that run from the PMT base to the SHV feed-thru. Outside the ball there are 55 foot RG-59 cables that supply the HRA with high voltage. They run from the SHV feed-thru on the endplate of the ball to the splitter boxes for the HRA in the electronic racks.

- **Patch Panels**
  There is one patch panels from the N2 vault to Data U-3. This has

20 RG-58 cables with BNC connectors and 9 RG-59 cables with SHV connectors. There is also a video signal patch panel in the N2 vault with 5 cables that run to the video patch panel just outside the entrance to the K500.

Chapter 3

Electronics

"Goddamn Westfall!"
— Michael Maier

3.1 Introduction

There are presently 12 standard 19 inch racks in the N2 vault for the 4π electronics. The first of these racks (the one closest to the vault door) holds the target motor controller, vacuum ionization gauge, terminal server and MODICON controls (a PanelMate terminal). The next three adjacent racks hold the controls for the Bragg Curve Counter and PPAC gas handling system. This includes the MKS controllers, open/close relays and a "Fermi Station" terminal. The remaining racks contain CAMAC, NIM, VME and delay crates for the electronics and data acquisition. The last four racks are used for data acquisition. Two racks are located near the exit of the BALL and house PPAC power supplies, MFA electronics and temporary or future electronics. The crate layout of four data acquisition racks is shown in Fig. 3.1. The LeCroy high voltage power supply is also located in the lower right hand rack, underneath the splitter boxes.

Figure 3.2 is an electronics diagram of one of the phoswich detectors. The components that are contained within the vacuum chamber (PMT, PMT-base and divider box) are surrounded by the dashed box in the figure. Each detector receives its high voltage and transmits its signal over a single SHV cable. Therefore, 215 such cables run from their respective feed-thrus on the vacuum chamber, along the cable trays, and to the electronics racks. The cables plug into the back of the 15 splitter box modules where the phoswich signal is separated into its fast, slow, and timing signals.

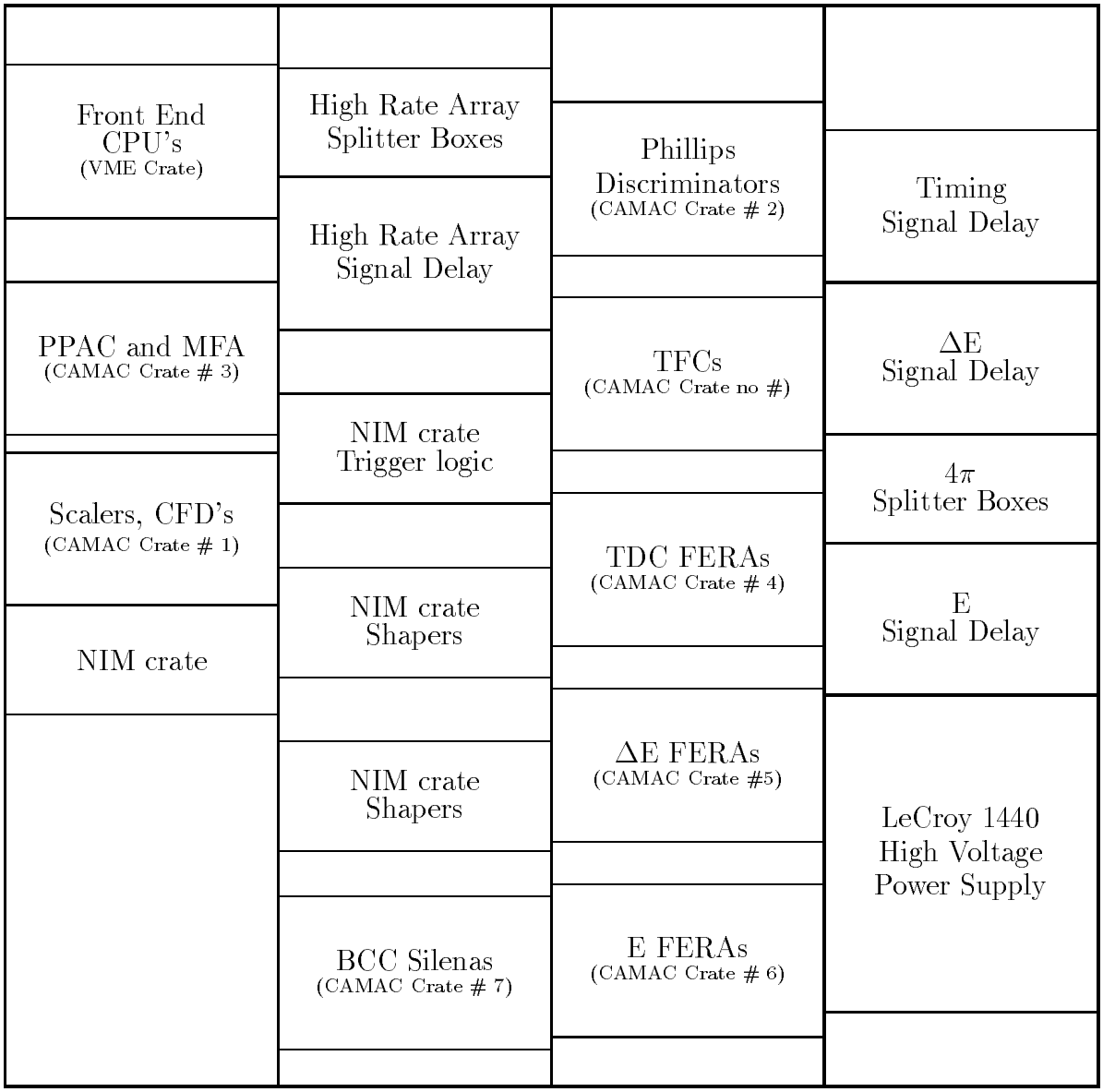


Figure 3.1: 4π Electronics Crate Layout

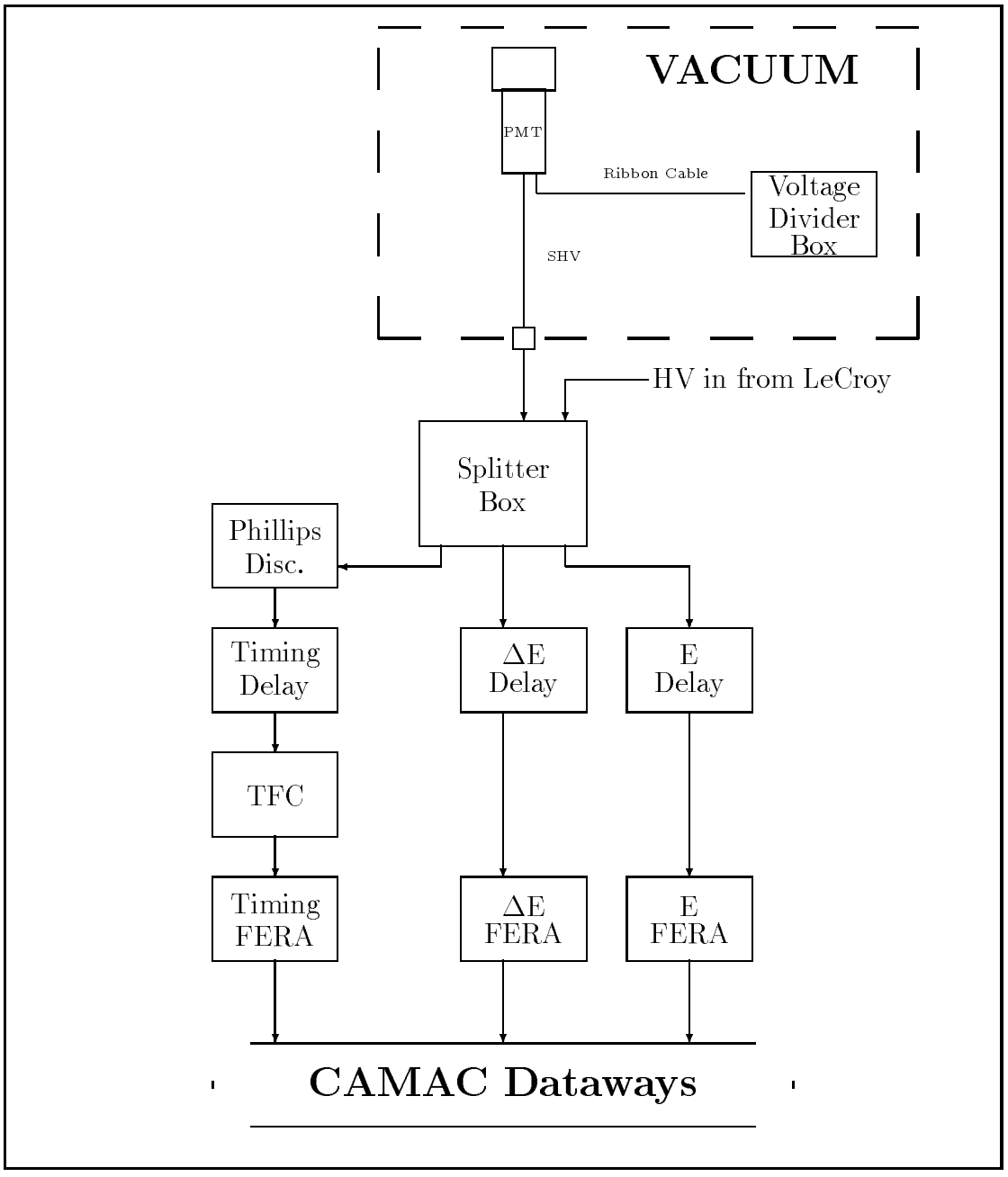


Figure 3.2: Phoswich Detector Electronics Diagram

The following sections give a more detailed description of the above mentioned components. In addition, other electronic topics are covered.

3.2 Photomultiplier Tubes

3.3 Photomultiplier Tube Bases

3.4 Voltage Divider Boxes

3.5 4π Signal Splitters

The signal splitter boxes were designed for the 4π Array by Michael Maier and built in-house. The boxes are passive and each can handle up to 16 channels. The splitter boxes built for the ball detectors are the same as those used for the High Rate Array.

- **Ball:** There are 12 splitter boxes each handling a 'bank' of detectors. (*i.e.* odd A's, even A's, *etc.*) The back panel accepts 16 channels of high voltage from the LeCroy using a block connector. These voltages are passed through the back of the box via 16 SHV feed-thrus, which in turn go to the phototubes. The signal from a phototube is returned over the same HV wire and the splitter box separates it from the high voltage. This raw signal is connected to 16 Lemo feed-thrus located on the front of the box (sent to the discriminator). In addition, the fast and slow components are separated from the raw signal and passed through the front panel by two 34-pin connectors (sent to the E and ΔE delay boxes). The ratio of the split ΔE:E:time is 35:20:45.

- **HRA:** There are 3 splitter boxes for the 45 detectors in the HRA. Each splitter handles 15 dectectors from the HRA in succession. The circuit is exactly the same as described for the ball.

3.6 FERAs (Fast Encoding Readout ADCs)

3.6.1 Introduction

The Fast Encoding Readout ADC (FERA) is a commercial current integrating ADC produced by LeCroy (module 4300). Its advantages are its conversion speed ($\sim$10 μs), zero supression, channel density (16) and fast read out mode (ECLine). Its disadvantage is that it does not have the resolution (11 bits with 10 to 20% differential non linearity) that other ADC's have. The FERA has several modes of operation, including zero suppression, pedestal subtraction and CAMAC and/or ECLine readout. ECLine readout in a non CAMAC ECL data transfer port that allows transfer rates at 1 word per 100 ns. In the 4π the FERAs are used for all signals from plastic scintillators and time signals from LeCroy 4303 FERETS. In the 4π data acquisition system the setup and control is done in software and does not require intervention. However, there are several basic things one should be aware of when using or adding FERAs to the system.

3.6.2 Initializing the FERAs

The FERA setup configuration is done via CAMAC commands. Commands such as reading status, setting pedestals can only be done when the FERA is NOT busy. Therefore, when reading or writing to the FERA (except for reading *data*) the FERA must be cleared and gating inhibited. The FERA can be configured in several modes. In the 4π system the FERA is set up for zero supression, pedestal subtraction and either CAMAC or ECLine readout depending on the readout system. In the FERA Status Register one can assign a Virtual Station Number (8 bits). This appears in the data stream in the FERA header word (bit 1 to bit 8) when running in zero suppression mode. In the 4π system the VSN is assigned the slot *and* crate number of the module in the following manner: VSN = CCCSSSSS where CCC is the crate number (bits 6 to 8) and SSSSS is the slot number (bits 1 to 5). This is done automatically in the 4π software when setting up the configuration mode. One should note that this is done at the beginning of every run and any changes made manually during a run will be reset.

Pedestals are setup in the software also via the software Integers. One should refer to section 10.4 to set these integers. These values are used by the front end computer to set the pedestals at the beginning of every run and therefore permanent changes must be done to the Integers. Note that these pedestals are applied individually to each of the 16 channels.

One final setup procedure is that new FERAs should have their 'O' pot adjusted for pedestal offset. This is done by placing the module on an extender and adjusting the 'O'

pot to minimize the digitized output. Note the input should be cabled up, although with no input signal, and the gate used should be the normal gate. In other words, wire up as in running condition.

3.6.3 Gating and Clearing the FERAs

Gating and clearing the FERA is generally done via the ECLine command bus from a FERA Driver (LeCroy 4301). If ECLine readout is done this is a necessity. The gate is supplied via NIM input to the Driver and all FERAs connected are then gated. Note the gate must precede the data signal by 20 ns. In the 4π system the time, ΔE and E signals all get an individual gate. If additional FERAs are added then they should (or must) be supplied with a Driver for each subsystem with a common gate.

Clears are supplied either by CAMAC commands to each Driver or module, or by the FERA FAUCET in the case of the ECLine readout mode. Alternately one could supply a NIM signal to the Drivers but, for historical reasons, this is not done in the 4π system.

3.6.4 Supplying analog data signals to the FERAs

The FERAs used in the 4π system have 100Ω termination. FERAs also come in 50Ω termination – one should always check. The 100Ω will have a 'Model 210' sticker on the back. Some 50Ω FERAs have been modified to 100Ω. Signals for 100Ω FERAs can be supplied via 100Ω ribbon cable.

3.7 Discriminators

3.7.1 Introduction

The discriminators are primarily Phillips Scientific with some LeCroy 4413. Functionally the two types are the same, but there are some important differences in the initialization. The purpose of the discriminators is to generate signals for timing, multiplicity and pattern logic. Each device in the 4π detector is assigned a channel in the bank of discriminators.

3.7.2 Initializing the discriminators

The discriminators are set up by the software with the exception of the front panel switches. The 4π system uses the Phillips in UPDATE mode and REMOTE mode. The LeCroys

require a CAMAC command to put them in remote mode. The discriminators have two sets of conditions to set up with CAMAC commands; the masks and the thresholds. These values are set up via software at the beginning of every run using values stored in the INTEGER table (see section 10.4). Note that the *masks* of the Phillips and the LeCroy have *opposite logic*. The 4π system normally does not mask channels.

3.7.3 Supplying signals to the discriminators

The input signals are 50Ω Lemo connectors normally from the splitter boxes of the 4π system.

3.7.4 Output signals from the discriminators

One of the ECL output signals from the discriminators is used for TOF signals, *i.e.* input to the FERETs. Additionally, the SUM output from the discriminators is fed to the SUMMERs and used for multiplicity discrimination. Optionally the other ECL signal is used for scaler input.

3.8 Time-to-FERA converters (FERETs)

3.8.1 Introduction

The FERET is a commercial time to charge converter to supply time signals compatible with a FERA ADC. Like the FERA the module is a LeCroy product (4303, Time to FERA Converter). The module has 16 channels and can be operated in either common start or common stop mode.

3.8.2 Initializing the FERETs

The first setup step required for the FERET is the START/STOP mode switch on the back. It has no CAMAC functions. In the 4π system the FERETs are used in the common stop mode. The only other set up procedure is the output amplitude. The output is of fixed voltage(current) and variable length. Adjust the STOP signal to the maximum time range that you want and then adjust the TFC amplitude(on front panel) to the FERA range you want.

3.8.3 Supplying signals to the FERETs

Timing start signals are supplied via 100Ω ribbon cable from the discriminators. Other signals are supplied via a ribbon cable to all the FERETs using common signals to all. The start signal must be supplied to *enable* the system even if in common stop mode and it must precede any individual *start* signal. Because the 4π uses ribbon cable, the stop and test signals both must be used and supplied from a common source. This will prevent ringing on the wire. Normally, if the FERETs are not ganged together, one or the other of the input signals is left open. One should note that the GATE signal of the FERET is an OUTPUT which is not used in the 4π configuration. Because the FERETs are ganged together all but one have the termination removed. The last one should be the terminated one. Note that some additional terminated FERETs could be used but after about six the signals will be degraded to the point signals may not work.

3.8.4 Output signals from the FERETs

A ribbon cable is strung between the FERET and a FERA for the analog signals. The gate to the FERA is supplied to encompass all possible timing signals.

3.9 Silenas

3.9.1 Introduction

The Silena 4418/v ADC is a commercial peak sensing ADC. Its advantages are its conversion speed (4 μs/channel), zero supression and fast read out mode (ECLine). It disadvantages are that its channel density is only 8, it is expensive, it requires a 1 μs risetime and the fact that it triggers on the first peak it registers. Its resolution is 3840 channels (4096 minus 256 channels for a sliding scale). Like the FERA it has several modes of operation which include zero suppression, pedestal subtraction and CAMAC and/or ECLine readout. ECLine readout in a non-CAMAC ECL data transfer port allows transfer rates of 1 word/100 ns. In the 4π the Silenas are used for signals from the Bragg curves to measure the energy and 'Z' of the particle. In the 4π data acquisition system the setup and control is done in software and does not require intervention. It is similar to the FERAs but there are several basic functions that are different. Note carefully that the format of the header word from the Silena differs from the FERA in that the word count (or 'valid data count') appears in bits 9 to 12, as opposed to bits 12 to 15 for the FERA. This difference means that data processing

programs *must know the location* (crate and slot number) of Silena modules used in a given experiment.

3.9.2 Initializing the Silenas

The Silena setup configuration is done via CAMAC commands. Commands such as reading status, setting pedestals can only be done when the Silena is NOT busy. Therefore, when reading or writing to the Silena (apart from reading *data*) the Silena must be cleared and gating must be inhibited. The Silena can be configured in several modes. In the 4π system the Silena is set up for zero supression, pedestal subtraction and either CAMAC or ECLine readout depending on the readout system. In the Silena Status Register one can assign a Virtual Station Number (8 bits). This appears in the data stream as the Silena header word in zero suppression mode. In the 4π system this number is assigned the crate and slot number of the module in the following manner: VSN = CCCSSSSS where CCC is the crate number and SSSSS is the slot number. This is done automatically in the 4π software when setting up the configuration mode. One should note that this is done at the beginning of every run and any changes made manually during a run will be reset. Pedestals are setup in the software via the software Integers and one should refer to section 10.4 to set these values. These values are also set at the beginning of every run and therefore permanent changes must be done to the Integers. Note that these pedestals are applied individually to each of the 8 channels. *Unlike* the FERAs there is an upper and lower threshold along with a common threshold. There is also an offset for each channel.

3.9.3 Gating and Clearing the Silenas

Gating and clearing the Silena is generally done via the ECLine command bus from a LeCroy Driver (LeCroy 4301). If ECLine readout is done this is a necessity. The gate is supplied via NIM input to the Driver and all Silenas connected are then gated. Care should be taken to set the gate time so that it does not include much other than the desired peak. The peak shape is also critical and one should insure that it has at least a 1 μs risetime. If additional Silenas are added then they should(or must) be supplied with a Driver for each subsystem with a common gate.

Clears are supplied either by CAMAC commands to each Driver or module, or by the FERA FAUCET in the case of the ECLine readout mode. Alternately one could supply a NIM signal to the Drivers but for historical reasons this is not done in the 4π system.

3.9.4 Supplying analog data signals to the Silenas

The Silenas used in the 4π system have 1000Ω termination and expect positive polarity. The Silenas "trigger" on the first peak NOT the largest peak in the gate. Basically you should have clean, well shaped signals and a gate as narrow as possible. It is a very good idea to read the Silena manual(A very, very good idea).

3.10 Octal CFDs

3.10.1 Introduction

The Octal Constant Fraction Discriminators (OCFDs) are commercial modules produced by EG&G. In the 4π one is used used for timing, and another for the master trigger. These modules have 8 NIM inputs and 2 – 8 channel ECL outputs along with a sum and ORed output. One discriminator is used to set the separate timing for the ball and the forward array. The other discriminator is used to select which trigger mode will be used: Ball, Forward Array or both. Also the multiplicity is selected with this trigger.

3.10.2 Initializing the discriminators

The software at the beginning of every run sets up the discriminators via the Integer table (see section 10.4). Any changes to the module not done via this table (*e.g.* done through `BCNAF` commands) will be overwritten at the beginning of each run. There are four sets of values to be setup via CAMAC commands. They are the individual thresholds for the eight inputs, the width for both the A and B output channels and the mask for the input channels. Normally the 4π system has the timing discriminator set for minimum multiplicity with the mask set to allow both Ball and Forward Array triggers to pass through. The other discriminator is set up as to the required running conditions, *i.e.* the source and multiplicity of the trigger. The output width is set to a minimum.

3.10.3 Supplying signals to the discriminators

In the 4π system the inputs to the discriminators are the NIM outputs of the Summer Boxes. These are the Ball multiplicity, the Forward Array multiplicity and the sum of both which goes to the trigger selection OCFD only.

3.10.4 Output signals from the discriminators

The NIM OR output of the trigger selection OCFD is used as the MASTER gate for the system. The B ECL output of the trigger timing OCFD is used to trigger the Ball and Forward Array ADCs separately.

3.11 Trigger Electronics

The are many other possible ways of triggering the the 4π. One such way is phoswich multiplicity. The multiplicity of phoswiches registering a hit during a given event is determined by adding the 'sum' outputs from the Phillips discriminators. This output generates a negative voltage level equal to -50 mV for each channel that has exceeded the threshold level. The 'sum' outputs from the twelve Phillips used for the standard 'ball' detectors are added in a linear voltage adder while the outputs from the three discriminators for the High Rate Array are added separately. The linear voltage adders have three outputs each. One output goes directly to a constant fraction discriminator (CFD) which will generate a trigger signal based upon a multiplicity level for either the 'ball' or the 'High Rate Array'. A second output from each linear voltage adder is sent to a secord CFD that generates the individual gate triggers. The third signal is sent third summer. The two signals are then summed to form an overall system multiplicity, which also goes to the CFD. Thus the experimenter has a choice of three standard phoswich triggering schemes:

1. 'ball' multiplicity
2. 'High Rate Array' multiplicity
3. 'system' multiplicity

The remaining CFD inputs are used for PPAC, Bragg, MFA or any other trigger of choice. These triggers are produced in the same manner as above with discriminator outputs and summers.

Having generated this triggering signal, a coincidence must be made with the livetime signals from the computer hardware. If the computer is busy handling previous events, this event will be ignored. If the coincidence is made, we have a *Master.Live* signal. This signal is fanned out many times. One output flags the computer to read this event, another goes to a scaler, others go to the individual gate generators that create the various gates for the banks of FERAs and SILENAs.

3.12 Gating and Timing

blah blah blah

Figure 3.3: Ball Electronics.

Figure 3.4: Phoswitch Timing.

Chapter 4

Voltage Supplies

4.1 LeCroy 1440 HV Power Supply

The high voltage for the 170 ball phototubes and the 45 High Rate Array phototubes are powered by a LeCroy 1440 High Voltage Power Supply located in the first rack of the 4π electronics. All voltages are positive and range from 1100 volts (High Rate Array tube) up to 2000 volts (ball tube) Communication to the LeCroy is via a RS232 interface on the 1445 Controller (9600 baud). This can be connected directly to a terminal using a null-modem cable or, as presently used, connected to the terminal server for network access.

The 4π LeCroy terminal line is assigned the logical name 4PI_RAP, which has an alias 4PI_LECROY. It can be accessed from the terminal server by typing `CONNECT 4PI_LECROY` at the `Local>` prompt, or from any VAX terminal session by typing `$ SET HOST/DTE 4PI_LECROY`. The built-in software for the LeCroy allows simple voltage manipulations (See the LeCroy manual for a description). In addition, a 4π–LeCroy interface language was written to allow control of the voltages using 4π terminology. (*i.e.* detector names, module names, *etc.*) For a complete description see section 4.2.

The high voltage outputs from the LeCroy are separated in groups of 16 channels called modules. The HV modules are located in the 16 available slots in the back of the LeCroy (thus 256 channels max.). For the 4π Array, each module controls a 'bank' of detectors. The HV modules in slots 0-4, 6-10, and 12-14 each contain an unused channel (15). Since the pentagon modules have no 'F' detectors, slots 5 and 11 each have 6 unused channels. For a listing of the channel number assigned to a given detector see Table 4.1.

Table 4.1: Lecroy Module Map

HV module slot #	Detectors Controlled	HV module slot #	Detectors Controlled
0	1A, 3A, ... 29A, MFA1	8	2C, 4C, ... 30C, MFA11
1	1B, 3B, ... 29B, MFA2	9	2D, 4D, ... 30D, MFA12
2	1C, 3C, ... 29C, MFA3	10	2E, 4E, ... 30E, MFA13
3	1D, 3D, ... 29D, MFA4	11	2F, 4F, ... 30F, MFA14-16
4	1E, 3E, ... 29E, MFA5	12	HRA1, HRA2, ... HRA15
5	1F, 3F, ... 29F, MFA6-8	13	HRA16, HRA17, ... HRA31
6	2A, 4A, ... 30A, MFA9	14	HRA32, HRA33, ... HRA45
7	2B, 4B, ... 30B, MFA10	15	ZDD1, ZDD2, ... ZDD 16

4.2 LeCroy Power Supply Interface Language

4.2.1 Introduction

A software package exists that creates a user-friendly interface to talk to the 4π LeCroy 1440 Power Supply. It is capable of performing all of the built-in operations of the LeCroy "firmware", plus additional features tailored to the 4π detector configuration. Additional routines allow, *e.g.* , running a diagnostic check of all voltages, monitoring the LeCroy status during experiments, reading and writing voltage files, and scaling voltages on-line. All available commands are fully documented here as well as in an on-line HELP menu. Please make use of it.

4.2.2 How to run the LeCroy Interface Program

To run the routine type: `RUN DATAQ:[4PI.LECROY]LECROY`
You will see ...

```
Connection made to LeCroy
M1
Type HELP for assistance
LECROY> enter commands here
```

4.2.3 Description of Commands

The following commands are defined ...

COPY, CURRENT, DIFF, EXIT, HELP, INCREMENT, M1(M2), MONITOR, OFF, ON, READ, SCALE, STATUS, TEST, WRITE

- **COPY**
  Format: **LECROY> COPY** (abbr. "CO")
  The copy command calls up a menu that has five options...

  1 **Write** the voltages that are currently in the demand, backup, and actual memories of the LeCroy supply to disk. The file VOLTAGE.DAT will be created in the default directory.

  2 **Swap** the demand and backup memories of the LeCroy. This is the same as the LeCroy "SWAP" command.

  3 **Copy** the voltages in the demand memory to the backup memory while leaving the demand memory unchanged. This is the same as the LeCroy "COPY" command.

  4 **Read** in a voltage data file that was created using option #1. If the entered file is not found a search routine is ran that allows the user to search any account for the file using DCL wildcards. The user is asked whether to read in the demand or backup voltages from this file, and can choose to read the voltages for only certain sub-arrays. The voltages read in will be placed in the demand memory location of the LeCroy.

  5 **Exit** the copy routine and return to the LECROY> prompt.

- **CURRENT**
  Format: **LECROY> CURRENT** (abbr. "CU")
  This command first displays the present values of the current limits and then prompts the user for any desired changes.

- **DIFF**
  Format: **LECROY> DIFF** (abbr. "D")
  This command compares all demand voltages currently in memory to

either the backup voltages or a voltage file (Demand or Backup voltages), at the user's request. This is particularly useful in identifying any voltages that have been changed in memory, and in comparing the voltages in memory to those in a file. Any differences found are typed out to the screen with the current demand voltages listed under "DEMAND" and the compared voltages (*i.e.* from Backup memory or file) listed under "BACKUP".

- **EXIT**
  Format: **LECROY> EXIT** (abbr. "E" or "Q")
  Closes the connection to the LeCroy Mainframe and returns the user to the DCL level.

- **HELP**
  Format: **LECROY> HELP** *[command]* (abbr. "H")
  Calls up the on-line help menu.

- **INCREMENT**
  Format: **LECROY> INCREMENT** (abbr. "IN")
  This command allows the user to increment voltages in either the mainframe or a voltage file, on-line. The user is prompted for the desired change in voltage, by sub-array. The increments entered will be interpreted as additive changes, and will be applied to the DEMAND voltages. The resulting voltages can be written to the mainframe, a voltage file, or both.

- **M1(M2)**
  Format: **LECROY> M1(M2)** (no abbr.)
  These commands allow the user to address mainframes number 1 and 2, respectively. The M1 command is issued at startup. These address commands should produce the return message: "Mainframe x responding" ($x = 1$ or 2)

- **MONITOR**
  Format: **LECROY> MONITOR** (abbr. "M")
  This command starts up a monitor process which runs a continuous test on the LeCroy. The user is alerted if the high voltage has been turned off, or if a particular channel is not biasing properly. The number of

minutes between each successive test must be entered at the beginning of the routine.

- **OFF**
  Format: **LECROY> OFF** (no abbr.)
  Turns the high voltage off for all channels of the LeCroy.

- **ON**
  Format: **LECROY> ON** (no abbr.)
  Turns the high voltage on for all channels of the LeCroy.

- **READ**
  Format: **LECROY> READ** *[detector-list]* (abbr. "R")
  The read command is used to read the present voltage of a particular channel or group of channels.
  detector-list: Is a list of the detector names to be read. (If no value is entered, the READ will be performed on the detector that was last READ or WRITTEN.)
  Some defined words are...

Detector names:	1A,1B,23C,30F,FA1,FA45 *etc.*
LeCroy channels:	0,236,10–34,123,100–200 (ranges allowed)
Banks:	oddA,oddB,oddC,oddD,oddE,oddF
	evenA,evenB,evenC,evenD,evenE,evenF
	FAslot1,FAslot2,FAslot3
Modules:	MOD1,MOD2,.....,MOD30
Generic:	BALL,FA,MFA,ZDD,ALL

- **SCALE**
  Format: **LECROY> SCALE** (abbr. "SC")
  This routine is similar to the INCREMENT command, except that it allows the user to scale voltages by a multiplicative factor, entered as a percentage change.

- **STATUS**
  Format: **LECROY> STATUS** (abbr. "S")

The STATUS command returns the present condition of the LeCroy high voltage, *i.e.* ON/OFF, HV ENABLED/DISABLED.

- **TEST**
  Format: **LECROY> TEST** (abbr. "T")
  This routine reads the voltages on LeCroy channels 0-255 and performs 3 tasks ...

  1 Lists all detectors that have demand voltages less than 500 volts. (Those with zero demand voltage are listed separately.)

  2 Lists all detectors whose actual and demand voltages are different by more than 25 volts.

  3 Lists those LeCroy channels that are biased but do not correspond to any detector.

- **WRITE**
  Format: **LECROY> WRITE** *voltage [detector-list]* (abbr. "W")
  The WRITE command is used to write a given voltage to a particular channel or group of channels.
  detector-list: Is a list of the detector names as defined above under READ. If no value is entered for *detector-list* the WRITE will be performed on the detector that was last READ or WRITTEN to.
  voltage: Is the number of volts that is to be applied to all detectors in detector-list. A voltage must be supplied.

4.2.4 Program Structure

All required subroutines are stored in DATAQ:[4PI.LECROY]. For a brief description of the individual routines see

DATAQ:[4PI.LECROY]LECROY.DOC

4.3 C.A.E.N. SY127 HV Power Supply

The high voltage for the 30 Bragg curve counter anodes and cathodes is supplied by a C.A.E.N. model SY127 40-channel High Voltage Power Supply located in the middle rack

of the 4π electronics. Remote communication to the CAEN is via a RS232 interface (9600 baud). This can be connected directly to a terminal using a null-modem cable or, as presently used, connected to the terminal server for network access.

The 4π Caen terminal line is assigned the logical name 4PI_CAEN. It can be accessed from the terminal server by typing `CONNECT 4PI_CAEN` at the `Local>` prompt, or from any VAX terminal session by typing `$ SET HOST/DTE 4PI_CAEN`. The built-in software for the CAEN allows quite sophisticated voltage manipulations (See the manual for a description).

4.4 PPAC Power Supplies

The PPACs require two power sources, a (negative) high voltage supply, and a ± 12 V supply for the pre-amplifiers. The -HV power is supplied by NIM mounted (*e.g.* Ortec) individual power supplies, while the power for the pre-Amps is supplied by five rack-mounted supplies, which distribute the power via bundles.

Chapter 5

Pumpdown Procedure

5.0.1 Pumping and Venting Procedure

A Pfeiffer combination roughing and turbo pump is used to evacuate the 4π vacuum chamber (main ball). The following description and instructions apply to the ball and its vacuum system. The ZDD chamber, due to the presence of the exit gate valve, is equipped with its own independent vacuum chamber. For information on the ZDD chamber (pumping procedures *et al.*) refer to section 2.4.3.

The pumps operate at the rate of 2000 liters per second and achieve pressures in the ball down to about $8x10^{-6}$ Torr. The controls for the pumps and valves are on the PanelMate terminal in the vault. Note that there are separate mechanical pumps for the gas system exhaust and an auxiliary pump for roughing down the beam pipe leading to the dump. Do not confuse these with the large roughing pump that is used in the initial stages of the pumpdown operation and thereafter as a backing pump for the turbo. **WARNING: Make sure you know what you are doing. Mistakes can destroy the detectors inside or even cause mechanical damage to the vacuum chamber.**

- **Pumping Down**
  Before pumping down several things must be checked...

  1. All detector voltages are turned off.
  2. The HV Interlock Bypass (on PanelMate) must be OFF (or else you won't be able to start the pumps).
  3. Verify that the ball is sealed properly and that support bolts on the beamline bellows are tightened. Make sure the venting valve

on the beam box is closed. Also check the vent valve on the exit beam pipe at the beam dump.

4. Verify, by looking at the OPT-22 LED indicators and the FERMI console, that the Bragg Curves and PPACs are vented to the chamber properly and that all other gas handling valves are open (High Vacuum Bypass, IN, OUT, Exhaust Bypass, Supply/Return). See Secs. 6.1 and 6.2.
5. Make sure the carousel gate valve is in the correct position. It can be open or close. Generally it's left open and pumped out with the ball.
6. The manual valve on the roughing pump is closed.
7. Verify from the PanelMate terminal that the following valves are CLOSED:
   - The gate valve between the injection beamline and the 4π Array
   - The gate valve between the ball and the exit beamline to the dump
   - The Roughing Isolator valves for the Bragg and PPAC lines.
8. Check that the power cable is connected to the turbo pump (this may get disconnected if the ball is rotated).

To start the pump down, toggle ON the '4Pi Pump' button on the PanelMate terminal. The rough pump should start. Slowly open the manual valve. Verify that there is no major problem by monitoring the 0–30 inch dial gauge mounted on the ball entrance flange. Listen carefully for hissing sounds if you suspect a leak. If everything appears normal, toggle the '4Pi Turbo Pump' button on the PanelMate to Vacuum AutoStart. This means that the turbo pump will start automatically when the pressure reaches about 0.5 Torr. Once the 0–30 inch dial gauge is close to full scale, the pressure will be in range of the thermocouple gauge mounted on the ball. This is read out on the multiplexed digital display near to the PanelMate terminal. The read-out is in Volts – the conversion to pressure is displayed on the graph taped to the multiplexer panel. The turbo pump will time-out after 30 minutes (approximately) if the turbo has not come up to speed. The

needle in the rotor speed gauge will not have completely reached the green section. If the ball is still at an acceptable vacuum restart the turbo. Otherwise wait and then restart the turbo. This should not happen a second time unless there is an excessive load on the pump (such as a leak). After the vacuum has reach the 10-4 region, the ionization gauge can be turned on. The lowest pressure achieved in the ball ranges from $8x10^{-6}$ are achieved if the ball has not been up to air for less than 6-10 hours.

- **Letting up to Atmosphere**
  Great care should be exercised in letting the ball up to air. If possible it should always be done with two people, at least one of which is experienced. Take the following steps:

  1. Turn off detector voltages (PPAC, LeCroy and CAEN).
  2. Turn off BCC preamp power by unplugging the +/-12V supply.
  3. **CLOSE** the gate valve between the 4π vacuum chamber and the injection beamline . Override the exit beamline interlock to exhaust the exit pipe, otherwise close the exit gate valve.
  4. Remove the gas from the Bragg Curves and PPACs. This is done by setting the pressure to zero, setting the manual override on, and opening the exhaust bypass. When the roughing pump has pumped out the lines close the Supply and Isolator valves from the PanelMate terminal. A good indication of the pressure is the vacuum reading in the exhaust lines.
  5. Check Bragg Curve and PPAC valves are in "pump down" state (at the Fermi console).
  6. Park the target at HOME
  7. Shut the 4π turbopump off, close the roughing pump valve and turn off the roughing pump. (PanelMate).
  8. Without delay gradually open the vent valve on the entrance beam box near to quadrupole.

Chapter 6

Gas Handling

"What's the groove, Man"
— Prof. Roy Lacey

6.1 Introduction

The gas handling system is composed of two parts; one being the ON/OFF (OPEN/CLOSE) valves for pressurizing, venting, *etc.*, and the other part is for regulation of the pressure. The ON/OFF section is handled by a FERMI system digital output board through OPTO–22 I/O modules. The regulation is done at base level with a commercial system developed by MKS. This system is interfaced to the FERMI system, such that, in normal operation, the FERMI system is used to set the pressure through the MKS controllers which operate in 'external' mode. In case of problems with the interface, the MKS system can be used standalone ('internal' mode) to set the pressure directly. Both modes are described under section 6.3, *'Pressure Control'*.

Valve control and pressure setting are arranged in banks. Each bank has one gas handling system (*i.e.* one pressure and flow controller) to govern 6 detectors. The grouping of detectors within a bank is arranged horizontally along the ball (this is the geographical layout of the gas lines). The table below shows which Bragg curve counters are within each gas handling bank.

Bank	BCC module numbers
1	1, 11, 21, 6, 16, 26
2	2, 12, 22, 7, 17, 27
3	3, 13, 23, 8, 18, 28
4	4, 14, 24, 9, 19, 29
5	5, 15, 25, 10, 20, 30

Each detector has its own IN and OUT valve mounted on the detector flange. These valves are used to isolate individual detectors. During normal operation (gas flowing through detectors) they are open.

A diagram of a gas handling system is shown in Fig. 4.1. Each gas handling system has three ON/OFF valves (High vacuum, supply/return, and exhaust bypasses), a manual flow meter/controller, a pressure transducer, and a gas inlet control valve. The flow meter/controller works in any orientation but the setting can only be read correctly when it is vertical.

- During **normal operation** (chamber under vacuum, gas in the detector), all the bypass valves on the gas handling system must be closed.
- During chamber **pump-down** or **venting**, all the valves on the gas handling system should be open.
- When **pumping out the gas from a detector** (*e.g.* prior to venting the chamber), the exhaust and supply/return bypasses should be open. Do *not* open the high vacuum bypass until all the gas is removed!

6.2 Setting solenoid valves ON or OFF

6.2.1 Introduction

Control of the ON/OFF solenoid valves could, in principle, be done from any FERMI station which has the required page set up. In practice, this will usually be the FERMI station in the N2 vault. There are individual FERMI pages for various operations. There are 3 pages dedicated to the gas handling system and one page for general application that is also used to set valves. See the main menu for the page numbers. These pages are:

- The gas pressure page can be used to see all the gas line setting and readings. It also can be used to set individual settings. This does not control valves.

- The Bragg Curve page is used to set exclusively Bragg curve valves and pressures. The options on this page will apply to all 5 Bragg lines.
- The PPAC page is used to set exclusively PPAC valves and pressures. The options on this page will apply to all 5 PPAC lines.
- the BINARY page.

In order to make changes to the status of any valves, the safety key on the FERMI panel must be turned anti-clockwise (the setup can still be *viewed* with the key in the locked position). When you have finished, please remember to turn the FERMI station key clockwise to the locked position. This will avoid accidental changes to the valve settings. Changes are generally made by placing the cursor over the line or field in question and hitting the `<Kbd Int>` key.

6.2.2 The Bragg page

On the Bragg page there are options to set ALL the Bragg valves in PUMPDOWN state, i.e. all open or in RUN state, i.e. system valves closed. Care should be taken to insure that no damage will occur with a change of state. In other words you should never go to pump down state unless the Braggs and the Ball are at the same pressure. Also on this page is a quick dump option. This option will open the appropriate valves to dump the cells to the roughing pump and set the pressure to zero. Another option resets the valves but not the pressure. To use these options place the cursor over the line in question and hit the `<Kbd Int>` key.

6.2.3 The PPAC page

On the PPAC page there are options to set ALL the PPAC valves in pump down state i.e. all open or RUN state, i.e. system valves closed. Care should be taken to insure that no damage will occur with a change of state. In other words you should never go to pump down state unless the PPAC and the Ball are at the same pressure. Also on this page is a quick dump option. This option will open the appropriate valves to dump the cells to the roughing pump and set the pressure to zero. Another option resets the valves but not the pressure. To use these options place the cursor over the line in question and hit the `<Kbd Int>` key.

6.2.4 The BINARY page

Bring up the binary page (if not already displayed) by pressing the <HOME> key, then type B followed by pressing either the <ESC> key on the keyboard or the blue <Kbd Int> key on the front panel (these two are equivalent). On the lower half of the screen, valves and their status are displayed, either with the labels HIVAC BYPASS, EXHAUST BYPASS and SUP RET INT (gas handling system), or IN and OUT for each module. The bank number is displayed in column 9, the module number (in the case of the IN and OUT valves is in columns 14 and 15. The valve status toggle bit is in column 22. To illustrate this more fully, we give below two example lines from the FERMI display:

```
F33<B19 4 IN 9   BCC>1 <0000>   0
F32<B18 4SUP/RET INT>0 <0000>   0
```

The lower line shows the supply/return bypass valve in bank 4. The valve is closed (toggle bit is 0). The 'B18' indicates that the OPT–22 I/O module controlling this valve is the 19th on panel B (the labeling starts from 0, rather than 1). The upper line shows the IN valve for Bragg curve counter 9, which is in bank 4. The valve is open (toggle bit is 1).

If the desired valve is not on the screen use the blue buttons on the FERMI panel to move through the possible valves. Once the wanted valve is found, use the arrow keys on the keyboard to position the cursor over the toggle bit next to the name of the valve. A '1' on the toggle bit indicates the valve is ON (OPEN), a '0' indicates OFF (CLOSED). The red LED lights on the OPT–22 I/O modules should reflect these settings. Change the state of the valve with the <ESC> key on the keyboard or the blue <Kbd Int> key on the front panel.

6.3 Pressure Control

,

The operation of the MKS gas handling system via the FERMI control system is done through three I/O ports. The first is an Analog to Digital port which is used to set the voltage which in turn is translated to gas pressure. The second is the voltage readback which is translated to a gas pressure reading. The third port is binary I/O which is used to control the OPEN/CLOSE state of the gas input valve much like the front panel switch of the MKS. The binary I/O is also used to determine the state of regulation of each of the MKS subsystems via a binary input bit from each of the subsystems. The interface between the MKS and the FERMI system is done with OPTO-22 on the "H" panel of the gas control system. When the MKS is "wired" to the FERMI only the MKS manual switch is operative. The MKS pressure display is hardwired to the analog input so both the FERMI and the

MKS can display the pressure readings. There is a FERMI manual in the 4π library for those wishing more details on the FERMI system.

6.3.1 Hardware Setup of the FERMI-MKS Interface

The MKS can operate in either "External" or "Internal" mode. The selection of mode is done via dip switches in each of the MKS modules. See page III-6 of the MKS series 260 manual (a copy may be found in the 4π library). The system is normally set for external-direct. The signals for this mode come via the two plugs at the rear of the backplane housing. NOTE these two connectors appear the same but they are not interchangeable. Be careful to hook up the correct connector to each. See the same section in the MKS manual as noted above for the pinout. In the external mode the cables are connected to the "H" panel. This panel contains the digital and analog I/O.

The display signals from the MKS are wired into the analog input via the upper breakout box. Note the the display signal are also feed back to the MKS for display for channels 1-4. channels 5-7 are hardwired in connector 2. All display signals for the MKS go to the MKS connector two.

The set points come from the lower breakout box and go via the cables to the MKS. Digital I/O are wired to the OPTO-22 modules. The large ribbon cables interface the panel to the FERMI system.

To revert to the MKS standalone mode modify the dip switches noted above and replace the external cables with the short cables stored in the electronics cabinet located in the vault. NOTE the cables must be connected with the correct connector to each plug even though they appear interchangeable.

6.3.2 MKS Standalone mode

The gas pressure may be set with the MKS series 260 control system directly (without the FERMI interface). This is *not* the normal mode of operation, but we discuss this first, since it is useful to familiarize oneself with the MKS hardware first.

The entire MKS system consists of the Type 260 controller which accepts an input signal from the Baratron Type 223B pressure transducer and opens or closes a Type 248A/B solenoid valve to maintain a set pressure. The measured pressure is displayed by a Type 247B multiplexed LED display unit.

There is full documentation on the MKS system in the 4π library. Users unfamiliar with the system are encouraged to read this first. The front panel of each controller has a 10-turn

set-point potentiometer, an "AUTO/CLOSE" switch, a zero adjust, and 3 LEDs labeled "ON", "DEVIATION", and "OVERRIDE".

- The zero potentiometer may be used to offset zero errors in the transducer. With zero pressure, the zero pot is adjusted such that the display unit reads zero.
- The set-point potentiometer is the user's input (requested set pressure). When in AUTO mode, the controller will attempt to maintain the set pressure by opening or closing the control inlet valve. For the Bragg Curve Counters, the full scale pressure on the dial is 1000 Torr.
- If the AUTO/CLOSE switch is in the CLOSE position, the gas inlet valve will be closed, regardless of the set-point pressure. Switching to AUTO will turn the controller on.
- The "ON" LED indicates whether the controller is supplying a signal to the control inlet valve. The intensity is indicative of the level of the output.
- The "DEVIATION" LED indicates when the measured pressure is within the 'process limit' of the requested set-point. The indicator is green when the measured pressure is within limits, and red when outside limits. The limit itself is set by an internal pot.
- The "OVERRIDE" LED indicates whether the controller is in manual (switch in CLOSE position) or automatic (switch in AUTOMATIC position) operation.

In the standalone mode, the dip switches on the MKS modules must be set to 'internal'. See section 6.3.1, *Hardware Setup*, for details.

6.3.3 Options for Setting the Pressure through the FERMI System

The MKS system should be in "external' mode and powered up. The system is divided into two parts; the Bragg Curve and the PPACs. The lefthand power supply is for the Bragg Curves, the right for the PPACs. Section 6.2 tells one how to use the FERMI system, read this first if necessary. To set the pressure there are two options. One is to go to page 2,

labeled Gas Pressure, on the FERMI index page. Put the display in "SET" mode by pressing the the appropriate button on the FERMI console panel. The value displayed will be the setting. Place the cursor over the line labeled with the gas line in question (MKS01 thru MKS14). Use the black knob to dial in the setting. The settings are in *volts* if in voltage units, and the same voltage corresponds to different pressure settings for the BCCs and PPACs. If the display is in engineering units (press the "ENGR" key on the FERMI front panel), the values are in *torr* and are correct units for either type of detector. Toggle to the "READ" mode to get the reading. The reading can be in voltages or engineering units (torr). Again the buttons on the console are used to select the desired mode. If you wish to type in the settings, place the cursor over the displayed pressure field and type in the number. Use the interrupt key to enter the value. Note that the pressure displayed on the FERMI station and the MKS display are not quite the same. The MKS display appears more stable. This might be due to an averaging done in the MKS. The FERMI also seems a little higher in its reading by 0.012 volts. The system can be monitored for regulation by noting the upper 16 bits of the "H" panel on the binary page. To force the control value to close, (*i.e.* to put the MKS in manual mode), use the MKS front panel "OVERRIDE" switch or set the binary bit on the lower 16 bits of the "H" panel. See section 6.2 for details on setting binary bits.

The other option is to go to the Bragg (or PPAC) page and place the cursor over the line and field labeled appropriately. Type in the desired pressure and hit the `<Kbd Int>` key. This will set the pressure for all the Braggs(or PPACs) at once. Care should be taken in that the FERMI system takes the contents of the field not just what is typed in and that the cursor is still on the line when the key is hit.

6.3.4 Procedure for Setting the Pressure

Since a group of 6 BCC (or PPAC) detectors share the same gas handling system, the same gas pressure is common to all within the bank. However, individual detectors may be isolated by closing both the IN and OUT solenoid valves for a module with a gas leak.

The procedure to set the BCC gas pressure after the chamber has been pumped down is now discussed.

1. As precautionary measures in case of gas leaks in the detectors, the ball should be isolated from the beam line by closing the entrance gate valve and the high voltage should be turned off the phototubes.

2. The high vacuum bypass, exhaust bypass, and supply/return bypass solenoid valves on the gas handling system should all be closed (see

Sec. 6.2). Turn on the MKS 15 V power supply for the controllers if it is not already on.

3. See below for details on how to prepare the gas lines and gas bottle.

4. Gas can be admitted to all detectors within a bank at once, or to each detector, one-by-one. If the latter procedure is desired, close all the IN and OUT valves from the detectors you wish to isolate initially.

5. Use the FERMI console to set the pressure to 10 Torr(1 Torr for PPAC). The pressure should rise on the FERMI and MKS displays and soon settle to the requested value. The chamber ion gauge should be watched carefully during this procedure, in case a leak has developed in any of the counters.

6. If you suspect a leak, turn the controller OFF then open the EXHAUST BYPASS solenoid valve. The gas should be pumped out of the counters. You may want to put gas in the counters one by one, in order to locate the problem.

7. If no problem appears, raise the set-point up to the desired operating pressure.

8. If you are putting gas in the counters one-by-one, isolate the counter you have just filled by turning off the IN and OUT solenoid valves, and proceed with the next counter in the same bank until all are filled. Then open all the IN and OUT valves for the bank.

To change the pressure after it has been set, simply enter the new requested pressure. The constriction of the manual flow meter means that large *reductions* in pressure take a long time. To avoid this, temporarily open the EXHAUST bypass solenoid valve.

To pump all the gas out prior to venting set the pressure to zero. Switch the controller from AUTO to CLOSE for added safety. Open the EXHAUST bypass and SUPPLY/RETURN bypass solenoid valves until all the gas is pumped away (at least down to the 1-2 Torr level). With the high voltage off the phototubes and chamber isolated from the beamline, open the HIGH VACUUM bypass valves.

6.4 Procedure for Setting Up Gas Supply

One should be completely familiar with this part of the system before attempting to do anything. It is also advisable for two people to be present. The procedures are similar for both the Bragg curves and the PPAC(MWPC) with the exception of the recycle tank and the supply pressure.

6.4.1 Bragg Curves

The Assuming that the supply lines are up to atmosphere or contain contaminated gas and the Bragg curves are pumped out up to the Roughing Isolator prepare the gas supply system by taking the following steps :

1. Make sure the gas filter is closed(the filter bypass is open), and the Roughing Isolator bypass valve is closed. If either of these conditions is not correct seek assistance. You DO NOT want to expose the filter to atmosphere or the Ball vacuum to the exhaust roughing pump.
2. Close off the MKS override valve.
3. Close off the manual value to the recycle tank and the valves on the tanks itself. If you feel the gas in the line between the valve and the recycle pump is contaminated seek additional assistance.
4. Open exhaust pump valve.
5. Open bypass ball side and use thermocouple A to monitor the pressure.
6. Pump until the pressure reads in the millitorr range.
7. Close bypass ball side.
8. Open bypass bottle side and use thermocouple A to monitor the pressure.
9. Pump until the pressure reads in the millitorr range.
10. Close bypass bottle side.
11. Open manual SUPPLY valves on recycle tank.

12. Open supply bottle to recycle tank and set pressure at 3 psi.
13. Open gas filter.
14. Close filter bypass.
15. Open up supply valve.
16. Place Bragg curves in Run state.
17. Open Roughing isolator AFTER checking pressure using thermocouple B.
18. Open MKS override.
19. Set pressure at nominal setting and let run for a few minutes.
20. Close off exhaust pump.
21. Open up manual valves to recycle pump and tank.

6.4.2 PPAC(MWPC) Curves

1. Make sure the Roughing Isolator bypass valve is closed. Seek assistance if it is not. You DO NOT want to expose the Ball vacuum to the exhaust roughing pump.
2. Close off the MKS override valve.
3. Open bypass ball side and use thermocouple A to monitor the pressure.
4. Pump until the pressure reads in the millitorr range.
5. Close bypass ball side.
6. Open bypass bottle side and use thermocouple A to monitor the pressure.
7. Pump until the pressure reads in the millitorr range.
8. Close bypass bottle side.

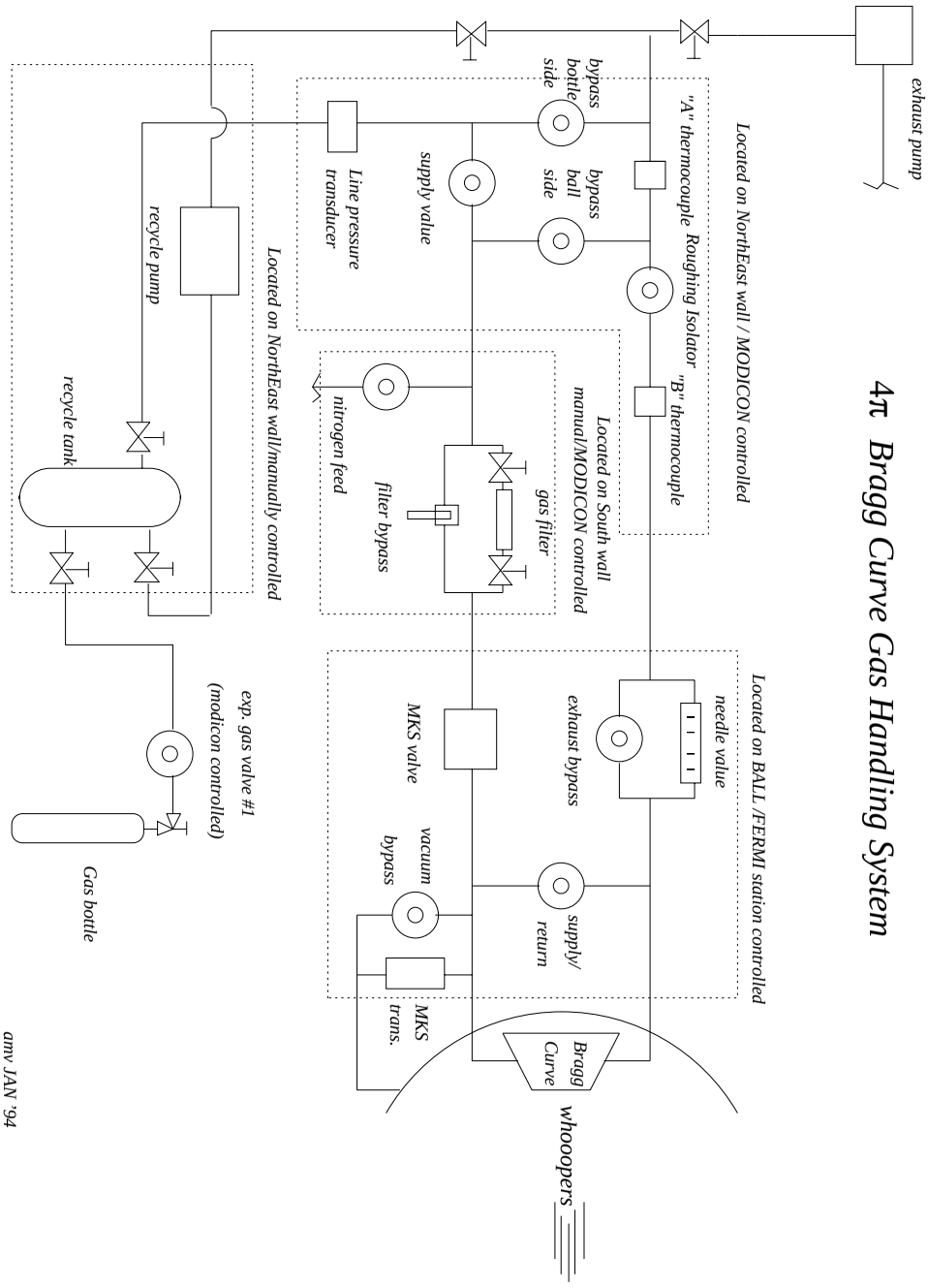


Figure 6.2: Bragg Curve gas system.

9. Close off exhaust pump.
10. Open supply bottle and set pressure at 3 psi.
11. Open up supply valve.
12. Place PPACs in Run state.
13. Open Roughing isolator AFTER checking pressure using thermocouple B.
14. Open MKS override.

Figure 6.3: PPAC(MWPC) gas system.

Chapter 7

Target Control

Freddie
Lives!!

7.0.3 Swinging-arm target.

The new-and-improved target mechanism consists of a swinging arm which puts a single disk target into the center of the ball, and a carousel which holds up to eight targets. The carousel can be valved off from the ball, which permits exchange of targets in the carousel without venting the ball. In addition, a target exchanger can be attached to the carousel, allowing the use of targets which cannot be exposed to air (such as calcium). The carousel-to-ball and carousel-to-exchanger valves are manual valves with interlocks. The interlocks only warn you if you are about to do something foolish. They do not prevent you from doing it.

Pump down and venting of the carousel is controlled from the PanelMate. In the situation that the ball is under vaccuum and you want to replace or add a target you should follow this procedure:

1. Unload all targets and position the arm at HOME.
2. Close the gate valve between the Carousel and the Ball.(Hand operated)
3. On the PanelMate terminal go to Page 0.
   Open the Carousel Vent Valve.
4. Remove the plastic cover on the carousel.

The targets are mounted on rings of ID 5/8" (approx. 1.6 cm) and OD 7/8" (approx. 2.2 cm). The rings are supported by two thin rods which connect to a base. When you mount a new target you must take great care in aligning the frame and rods. Use the guide mechanism for the correct height and alignment. After placing the target holder in the carousel tube insert the tube in an open slot in the carousel. You must rotate the tube clockwise until it clicks and snaps back a couple of mm's. Note that if the arm is not at "HOME" there may be a target on it. Make sure you are not filling a slot that will be used.

In the situation that the ball is under vaccuum and you want to pump down the carousel you should follow this procedure:

1. Replace the plastic cover on the carousel.
2. On the PanelMate terminal go to Page 0.
   Close the Carousel Vent Valve if not already closed.
3. Open the Carousel Pump Valve.
4. Turn on the Auxiliary Roughing Pump.

You can view the pressure on a thermocouple gauge. The setting is labelled Target Carousel (#2). When the vacuum reaches around 100 millitorr:

1. Close the Carousel Pump Valve.
2. Open the hand valve between the carousel and the ball.
3. Shut down the Auxillary Pump.

You will see a slight increase of the vacuum in the ball but it should be minor.

The pickup and replacement of targets in the carousel is controlled via a computer program 4PI_TARGET. To run this, type:

```
$ RUN DATAQ:[4PI.TARGET]4PI_TARGET
```

Chapter 8

Beamline

"You can tune a piano, but you can't tuna fish"
— Title of LP by Spike Jones

8.1 Beam Transport at NSCL

The beam transport system at NSCL allows beams from either cyclotron (K1200 or K500) to be delivered to any of the experimental devices in the various vaults. This is accomplished by a 'switchyard' of beam pipes and dipole magnets in the Transfer Hall. In addition, beams from the K1200 cyclotron pass through the A1200 analysis device before entering the Transfer Hall. The slits and apertures in the A1200 enables beams from the K1200 to be better defined in momentum and divergance. Primary beams can be stripped or degraded at the A1200 target 'pot' and then 'cleaned-up' (by removal of haloes or tails) in the remainder of the device. Secondary radioactive beams can also be produced.

Figure 8.1shows the beam transport layout in the Transfer Hall and the N2 vault.

8.2 Magnets, Wall plugs, and Gate valves on the 4π Line

The line from the A1200 to the 4π array is a straight shot along the 'B'-line. After the switching magnet A073DS (which is at zero field for the B-line), there are 4 quadrupole doublets before the 4π ball: B080QA-B082QB, B100QA-B102QB, B115QA-B116QB, and B131QA-B132QB. The last doublet is in the N2 vault, the others being in the Transfer Hall. In between the second and third doublets, there is another switching magnet that is also

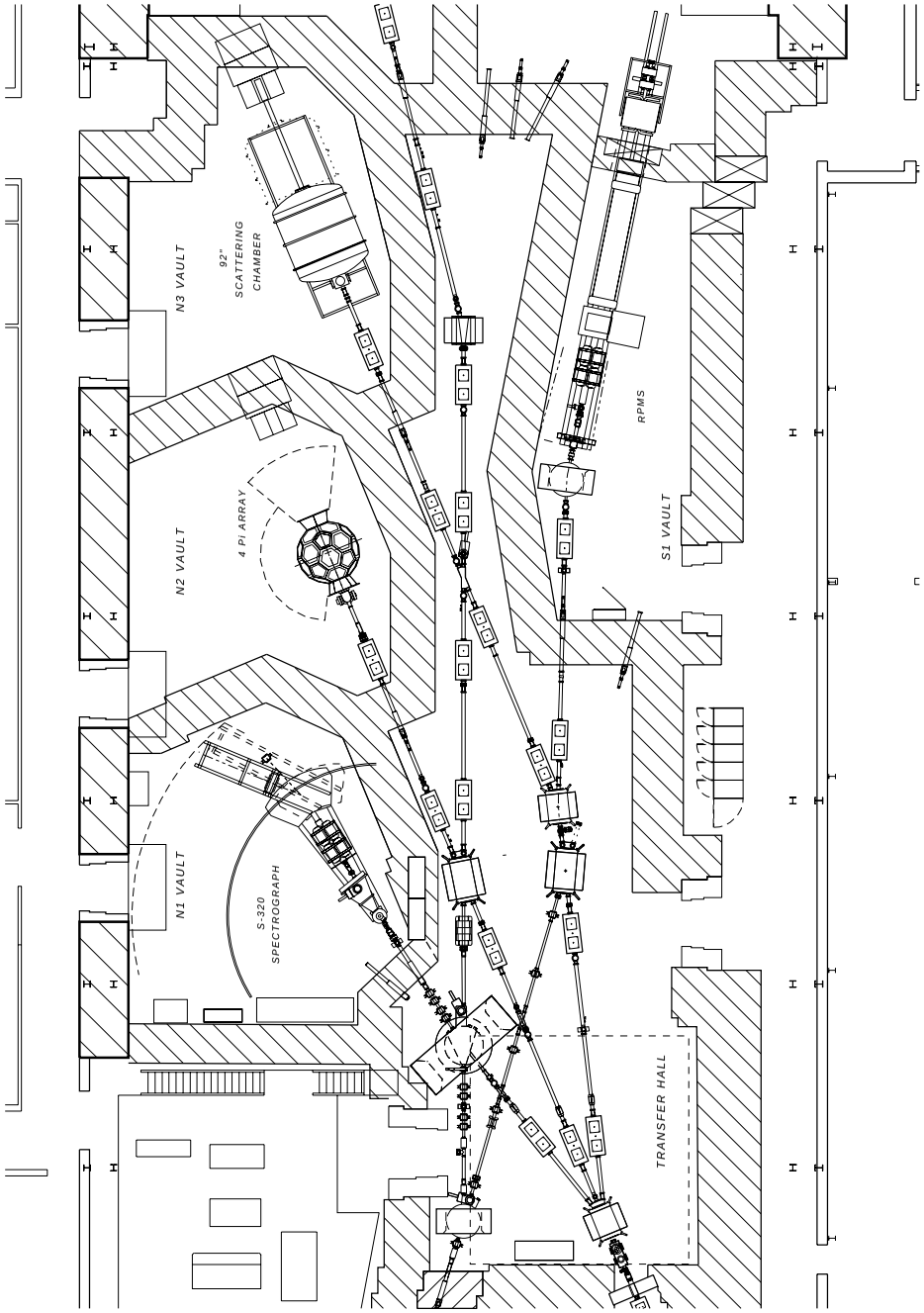


Figure 8.1: The Central High Bay.

at zero field for K1200 to 4π operation. There are several small room-temperature steering magnets on the K1200 beamline in the Transfer Hall. These are V085DV, B123DV, and B124DH. They are powered by the Kepco supplies in the racks in the South corridor.

The wall plug for the N2 vault is B125WP. The purpose of the wall plug is primarily to prevent neutrons entering the vault through the hole in the vault shielding wall for the beampipe. The wall plug must be inserted before entering the vault when the cyclotron is running. As part of the radiation protection system, the cyclotron rf will be interlocked off if the vault door is opened when the wall plug is open. The wall plug itself is not intended as a primary beam stop. The beam should be stopped by inserting a Faraday cup or the cyclotron Beam Blocker *before* running in the wall plug.

The beamline vacuum system is divided into numbered sections, each section being isolated by gate valves and having an independent pumping system. The vacuum sections and associated gate valves relevant to the line from the A1200 focal plane to the 4π array are listed in the table below:

Gate valve	From Section	To Section
A068GV	3	4
B093GV	4	6
B104GV	6	8
B133GV	8	4π
B148GV	4π	Dump

The gate valves listed above are all controlled from the Metra terminal on page 35. The vacuum status (thermocouple set point) for each section is displayed next to the control button.

8.3 Tuning the 4π Beamline

A table of elements for tuning beams from the A1200 to the 4π array is given below. (Note: the camera assignments, in particular, are fragile. They were accurate as of 18th March, 1991). Except for the A1200 (up to and including the A1200 focal plane) all the Scintillator and FC in/out controls and the switchable AC for lights are controlled from the Metra page 35. All the relevant wall plugs are also shown and controlled from this page.

Description	Location	Scint. & FC	Faraday Cup current	Light	Camera assignment
A1200 Focal plane	A071	A071SC†	A075FR	A071S1	14
B-line after 1st doublet	B093	B093FC	none	B093S1	02
B-line after B102DS	B111	B111FC	B111FR	B111S1	03
B-line before N2 vlt	B118	B118FC	none	B118S1	04
Entr. scint (N2 vlt)	B134	B134SC	–	B134LT	25
Target scint.	–	–	–	–	26
Exit scint. (N2 vlt)	B154	B154SC	–	B154LT	27

†Faraday cup control, A088FC, is separate.

Faraday cup currents are indicated on the KIM analog meters. The appropriate device name must be assigned to one of the top boxes on the graphic display.

Camera video output signals are patched to the multiplexer input board in the old magnet power supply room (outside the Transfer Hall). Camera displays can then be selected at the control console monitors.

Chapter 9

Alarms and Interlocks

"Awfully sorry, Roger. Seem to have pranged the kite."
— Monty Python's Flying Circus

9.0.1 Overview of Alarms and Interlocks

The alarms related to the 4π fall into three major catagories:

- Electronic hardware
- Modicon
- Software generated

Not all of these alarms and interlocks will or can prevent a catastrophe. One must not rely on these to prevent and accident.

9.0.2 Electronics

The electronics has three forms of alarms:

- Some CAMAC crates alarm on power and temerature. A Buzzer/horn will sound. Generally these imply a crate power supply needs to be looked at.
- The LECROY and CAEN both have an logic input that will turn off or prevent the high voltage from coming on. This is connected to the MODICON and linked to the ball vacuum. There is a MODICON override. Use this instead of removing the input as if you forget to replace

it the phototubes would be vulnerable to vacuum fluctuations. The LECROY and CAEN both have internal limiters for current. Check the manuals for details.

- The A/C power for the data acquisition racks is interlocked to the rack air temperature. There are two set points that are not accessable to the users. The first(around 75 F) will result in an alarm via MODICON for the operators. The second set point(around 80 F) will turn off the power to the DAQ hardware. A RESET will be found in in back of rack 3.

9.0.3 Modicon

9.0.4 Software

FERMI parameters can be accessed on the VAXs and therefore software alarms can be created on any of parameters. Three programs exist for this purpose. One monitors the Bragg cells and alarms on a 10 % varience or the MKS DEVIATION condition. A similar program exist for the PPACs. The Third program monitors the ball vacuum and alarms if it drops below 1x10-5 Torr. All of the alarms are audio and visual. The FERMI interface has some unused inputs can these can be used for special inputs if required.

Chapter 10

Data Acquisition and Support Programs

"His sister Pam / Works in a shop,
She never stops / She's a Go-Getter"
— Lennon/McCartney *(Abbey Road)*

10.1 DAQ Overview System

10.1.1 Overview of Data Acquisition System

The transputer based system is similar to the MASH based system in the way it appears to operate e.g. commands. However there are some major differences in hardware, software and philosophy such as additional frontend commands like the TRIGGER command and the way INTEGERS are loaded and saved. The hardware consists of a host system; a UNIX based SUN clone (with node name fourpi.nscl.msu.edu) linked to a VME transputer board. The links are transputer links not ethernet. Run control must be run from the host but it does not have to be at the main console. The transputer code is downloaded to the transputer from the host via `link0` and all communications transact over this link to/from a host resident program called ISERVER. Data is transfered back via `link1` to host based memory. There must be a program to receive this data which is generally TRANSRC. A routing program (ROUTER) is then started to send buffers to one or more consumers such as TAPE. Taping in done on the host. Note that buffers can be routed over ethernet to other workstations for displays, etc. This system can handle data rate above 300 kilobytes/sec. The major restrictions to this speed are the long (30 meter) links ($\approx$400 kb/sec.) and the tape drive (250 or 500 kb/sec.).

Figure 10.1: FrontEnd DAQ System.

Figure

10.2 Data Event Structure System

10.2.1 Introduction

The event structure for the MSU/NSCL 4π array is based on the standard NSCL event structure with modifications for zero-suppressed data from modules such as LeCroy 4300 FERA's and Silena 4418/V ADC's and for the FERA-Faucet-Maier (FFM). The standard NSCL event structure is described in the *"NSCL Data Acquisition System Reference Guide"*. Data (which here refers to run control information as well as scaler and "physics" event data) are transmitted in 8-kbyte buffers. Run control, scaler and physics data are kept separate in their own buffer types. Each buffer begins with a 16-word header, which contains information such as the buffer type, the number of used words in the buffer (note the buffer is *not* padded out with zeros and thus the unused words are old data) and the run number.

For the 4π array data acquisition system, the run control (*i.e.* begin-run, end-run, pause-run and resume-run buffers) and the scaler buffers are the same as for the standard NSCL acquisition. The structure of the physics data buffers is described in the following subsection.

10.2.2 Physics Data Buffer Structure

Buffer Header

At the beginning of the buffer is a 16-word header. This is very similar to the standard DAQ header apart from words 8, 9 and 10 which are not used in the 4π DAQ. For data taken with the FERA Faucet Maier, word 7 in the header has a different meaning than usual – in the transputer system if should always be 1. The used words in the buffer header are:

WORD	USAGE
1	Number of words in buffer with meaningful data
2	Buffer type (1 = Physics data)
4	Run number
5,6	Buffer sequence number within a run
7	1

Event structure

The FFM introduced another level of buffer structure. Within every buffer there is an additional 16 word FFM header. The contents of this buffers is used for error checking and may change over time. The buffer contains many events (or 'physics events') which are separated by a unique word: FFFF in hexadecimal. After the FFM header follows the first FFM separator. The FERA (and Silena) headers and data follow thereafter until the next FFM separator signals the end of that event and the start of the next.

Data Format - FERAs

The FERA header contains the information necessary to decode the data from that FERA: The word count is encoded in bits 12 through 15; Bits 1 through 8 are the Virtual Station Number, with the slot number encoded in bits 1 through 5, and the crate number in bits 6 through 8. Note that the VSN choice is implemented by the Front End code. Each FERA data word has the sub-address encoded in bits 12 through 15.
The (sub) event structure may be pictured as:

EVENT WORD COUNT or FFM SEPARATOR
1st FERA header
Data word 1
....
Data word n
2nd FERA header
Data word 1
....
Data word n
....
Last FERA header
Data word 1
....
Data word n
NEXT EVENT WC or FFM SEPARATOR
1st FERA header
Data word 1
....

Data Format - Silenas

The event format for the Silenas is similar to that for the FERA's except that an additional "Pattern Word" is present after the header word. This pattern word is a bitmask of the sub-addresses which fired for that particular module. Since the individual 4π Event Structure sub-addresses are encoded in each data word, this pattern word may be skipped over if desired.

10.3 Starting up the Acquisition System

10.3.1 Logging In

To log into the Sun system `fourpi.nscl.msu.edu` use the user name "user4pi". Ask a current 4π group member for the current password. In UNIX character case is important (case sensitive). "/ultrixusers/transputer/sources>" is the prompt. The prompt always tells you your current working directory (cwd). Type at the prompt:

/ultrixusers/transputer/sources> openwin

(The prompt will be abbreviated as "...>" from now on.) This command will start the windowing system OpenWindows (sort of like DECWindows or XWindows).

The procedure will create all the necessary windows. If you need (for some reason) an additional terminal window, double click with the left mouse button on the console icon (lower left corner) and type "xterm" in the window that comes up (usually at the top left corner). This will create an xterm terminal window somewhere in on the screen. You can move it around by clicking on the title bar and dragging the window and resize it by clicking on the corners of the window (if you move the cursor to one of the four corners it turns into a little disc).

10.3.2 Starting the data acquisition system

1. After the windowing system is running, the screen should look like fig. 1. The terminal window named "xterm" can be used as a generic vt100 terminal. From here (and only here) you can do things like printing, using telnet to connect to other nodes, etc. In the open terminal window (xterm) type at the prompt:

...> rctl

This opens a second window, the DAQ control window (DAQCW) in the lower right section of the screen. This window will have the name RCTL in the title bar and the icon (if you iconize it). From here you will start all the required DAQ software.

Note: In UNIX like in VMS there are directories and subdirectories. All the DAQ stuff is in the directory

/ultrixusers/transputer/sources/

Subnote: Subdirectories in UNIX are separated with forward slashes as opposed to dots in VMS, i.e.

VMS device:[user.sub1.sub2]*.*

corresponds to

UNIX /ultrixusers/user/sub1/sub2/*

Note that all users directories in our UNIX system are under

/ultrixusers/

which is the subdirectory ultrixusers/ in the root directory /.

The home directory for the user4pi account is

/ultrixusers/user4pi/

To switch between the home directory and the transputer directory type

...> home

to go to the home directory (there is rarely a reason to do that), and
...> trans
to go to the transputer directory.

2. The routing system consists of a source of data for the data router and the router itself. One program gets the data from the frontend interface and transfers it to the router as a data source. The router will then route the data to one or more sinks such as XTAPE. If you have data sinks such as ROUTER running on other workstations then a third program has to be started which acts as an external data sink for this ROUTER.

After having started the DAQCW the screen should look like fig. 2. In this window (RCTL) you start all the DAQ software by typing:

. ...> frontend – terminal window for front end program

...> transrc – data source for router

. ...> router – the routing program

. ...> bufserv – external sink program

. ...> xtape – the SUN/UNIX taping program

Note that this is UNIX so these commands are case sensitive.

The first command opens another window (name: frontend) in which the frontend program will be started automatically. This window is dedicated to the front end program. Don't do anything else in it.

The second command starts up transrc, which runs transparently in the background and supplies the SUN host with the data from the transputer.

The next command starts up the SUN router, which takes data from transrc and supplies it to whoever wants it on the SUN (e.g. bufserv, xtape and other SUN consumer programs).

The fourth command starts up bufserv, which is the link from the SUN to the outside world.

The last command starts the SUN taping program that opens two windows. The one in the upper left corner (/daq/bin/xterm) is an interactive tape control window and the one below (TAPECTL) tells you what its doing (double click on the title bar to see more). In the /daq/bin/xtape window you will find buttons to select the function you wish to perform. Generally you want to:

(1) Select the tape density ("Size" pulldown menu). Note: This does NOT determine the density the tape is written with; the switches on the tape drive do this.

2) If necessary set the tape number using the "+" and "-" buttons under the "Tape Number" box. You can also click on the "Tape Number" box and type in a number.

(3) Start the tape process by pushing the "Start Tape" button. Note: This is generally done once and not for each tape change. Use the "Change Tape" button for changing tapes.

Note: If the taping program is busy (mounting, dismounting or initializing tapes) the frontend will not respond to begin/end run commands even if taping is off!!!

This is not a problem. The frontend will respond as soon as the taping program is done with whatever is was doing. Stay calm.

From all those programs only the first (frontend) and last (xtape) need any user intervention. The other three are completely transparent to the user (except in case of FIRE, see below).

In the DAQCW (RCTL) window you can check the status of the DAQ system by typing at the prompt:

...> jobs

A list similar to the following should display

[1] + Running xterm -geometry 80x24-0+0 #64x64+960+836 -title frontend -n frontend

[2] - Running /daq/bin/transsrc

[3] Running /daq/bin/router

[5] Running /daq/bin/bufserver

[6] Running xterm -geometry 58x4+0+296 #64x64+896+836 -title TAPECTL -n TAPECTL -e

...

Here the number in the square brackets is the number of the jobs that runs (in the background, much like VMS spawn). The next field ("+", "-", " ") is not important. The next tells the status of the job. It should always be "Running". The field that follows is the command that started the job. There should always be an xterm with the option "-title frontend" (which is the frontend terminal window) and one with the option "-title

TAPECTL" (the TAPECTL window). Note that there is no "xtape" job in the list. This one is started from the TAPECTL window so you don't see it here. The other three are the transputer data source "/daq/bin/transsrc", the SUN routing program "/daq/bin/router" and the external data sink "/daq/bin/bufserver".

Another check of the DAQ system is

...> shpr

This can be typed from any xterm window. It looks at all processes on the system and finds the ones that contain the string "/daq/bin". There should always be (only the last filed, i.e. /daq/bin/... is relevant)

. user4pi 639 0.0 1.6 16 488 p3 S 11:58 0:20 /daq/bin/router
  user4pi 640 63.0 2.7 40 820 p3 R 11:58 28:49 /daq/bin/transsrc

. user4pi 650 0.0 4.0 568 1212 p5 S 12:23 0:00 /daq/bin/xtape

and if bufserv was started

user4pi 646 0.0 0.0 32 0 p3 IW 12:03 0:00 /daq/bin/bufserver

plus one more of those for each external router running.

The line

user4pi 661 0.0 0.7 32 204 p3 S 12:54 0:00 grep /daq/bin

can be ignored.

10.3.3 Running the FrontEnd

Running the frontend is similar to the previous system. Typing "help" will give you the syntax of the commands. Commands to the frontend are not case sensitive. The file names used in INTeger commands are though. Generally you will want to:

. Load the run parameter(INTeger)
  Reconnect the front end to the host (REConnect)

. Set the title (STItle)

. Set the run number (SRUn)

. Set the scalers (SSCa)

. Set the trigger (TRIgger)

. Set tape on (TAPe)

Type "STATUS" to verify the state of the frontend.

An example is :

. Please enter command :int d master_def.ped (a single character in place of the file name will get you the default file.

Please enter command :rec

. Please enter command :sti 150 mev/n 20ne plus 27al ball 2

. Please enter command :sru 1

. Please enter command :ssc 10 10 256

. Please enter command :tri ball 2

. Please enter command :tap on

. Please enter command :stat

. Please enter command :beg

Remember if the frontend is in the "connected" state then ROUTER must be running on the host.

10.3.4 In Case of Fire

1. No data coming back

a) One possible cause is that the FFM is hung. The result is that the Master.Live ... on the scaler page are zero, but the scalers are still updating (every 10 sec or whatever the scaler updating time is set to). In this case you should be able to end the run. If not see 2). If the run is ended and the frontend prompt is back type

...: rsv

to reset the VME crate. To see if that took care of the problem turn the taping off, i.e.

...: tape off

begin a run (not put to tape) and see if the scalers are OK. If they are end the run turn taping back on and resume taking data (either on a new tape or on the old one).

b) Another possible cause is that there is no beam! Check the current integrator scaler.

2. Cannot end the run

The result is that after typing

...: end

at the frontend prompt you don't get the prompt back.

This problem can have several sources. Check to make sure no ROUTER is starting or stopping or some consumer is hanging on to buffers. If so correct the problem or wait in the case of ROUTERs and the problem should then clear up. If not then :

a) The frontend program is hung.

The following procedure should take care of the problem.

- type CTRL-C in the frontend window (this kills the frontend window)
- bring up the frontend window by typing

...> frontend

- use the tape window option to put on an EOF on the tape

- take the tape out of the drive
- at the prompt in the RCTL window type
- in the /daq/bin/xtape window set the correct tape #, size, ... and restart
the taper by clicking on the Start Tape button
- put a new tape into the drive
- wait until the TAPECTL window tells you that the tape is initialized
- in the frontend window set the correct run #, trigger, etc. make sure taping is off and begin a run
- see if the scalers are correct

10.4 Run Time Parameters

The runtime parameters on the transputer based system can be changed individually or loaded from file stored in on fourpi.nscl.msu.edu in ultrixusers/transputrer/sources.

All the CAMAC setting, e.g. discriminator levels, pedestals, etc are set from values stored in memory. This block of memory is referred to as the INTEGERS. At the begining of each run or when the INItialize command is used the values stored in the INTEGERS are used to set the modules.

To change a Trigger use the TRIgger command. There are 11 possible trigger names of which three are redundent. You can have a BALL, FA or SYSTEM trigger or you can select the subaddress(0-7) you want to use along with the trigger value(multiplicity). The STAtus command will display the actual values used both for the multiplicity and the threshold. These values are programmed in for now. If you wish to change them you have to edit and recompile, etc. or use the INTeger command.

To change a pedestal use the PEDestal command. Its syntax is like a BCNAF where F would be the value , e.g.

Please enter command :PED 0 4 1 2 20

This will set crate 4 , slot 1 subaddress 3 to 20. (time pedestal for 5A). Note this sets the INTEGER value so it will only take effect after you begin the run or use the INItialize command. You can save these new settings on disk by using the INTeger uploading command, e.g.

Please enter command :int u new_file.ped start_integer end_integer

Another way is to edit the file on the host and download the new setting using the INTeger command, e.g.

Please enter command :int d edited_file.ped

If you need or want to edit the INTEGERS you can use the INTeger command to read or write individual parameters or download and upload parameters from/to disk.

. Please enter command :int w 1700 20 – write to INTEGER 1700 20
  Please enter command :int r 1700 – read INTEGER 1700

. Please enter command :int d edited_file.ped – download a file FROM the host to the frontend

. Please enter command :int u edited_file.ped 1700 2000 – upload a file FROM the frontend to the host for INTEGERS 1700 thru 2000

10.5 XSARA Histogramming

10.5.1 Motivation for XSARA

The histogramming program XSARA was developed to deal with the large number of parameters (*c. 500*) to be read in Phase I of the MSU 4π array. Conventional histogramming programs available at MSU (SARA and SMAUG) would be cumbersome for this task because the histogram binning is looped over all defined histograms. XSARA was modified from SARA such that the zero-suppressed parameters from each event are stored in a list, and a loop over parameters in this list is used to check conditions and increment histograms. In addition, the binning of histograms through a parameter list allowed the possibility of multiply occurring parameters within a given event. This in turn leads to histograms that may be incremented multiple times within a given event. An example of such histograms that have been used for the 4π experiments is the 'ADC-value versus ADC-slot-number' type.

Another adaptation of XSARA to the 4π array is the use of automatic labeling of parameters and histograms. Since the 4π setup is (more-or-less) permanent, and the cabling of modules into the FERA channels is done in a regular, well-defined order, it is possible to

generate labels for parameters and histograms based on crate, slot number and input sub-address. Similarly, simple applications programs have been written to generate histogram lists for the ball and forward array.

Since the program is based on SARA, the command language is similar, and most of the commands do similar things as those in SARA, if not the same. Some routines are used unmodified from SARA. Thus the SARA user manual and interactive help is mostly applicable, particularly for the operational command description and syntax. The same histogram display task (AEDTSK) is used as SARA.

10.5.2 How to Set Up XSARA for an experiment

The current version of XSARA (for the FFM) is kept on

DATAQ:[4PI.XSARA]

There is also a version located on

DISK$FOURPI:[FOURPI.XSARA]

In principal, the code is "standard" for experiments that use the plastic scintillators from the ball and the 45-element forward array. In practice, quite a few experiments have been performed with some additional detectors (*e.g.* silicon telescopes).

The following procedure is used to set up XSARA for "non-standard" experiments. (You will need to have a copy of the XSARA user's guide, be familiar with CMS libraries, and have access control (ACL) to DATAQ:[4PI.XSARA] to do most of this).

1. A new version of the subroutine DATAR may need to be made if, for example, the event structure changes, or changes need to be made to the 'event counter' pseudo parameters, or the 'word count' pseudo parameter. DATAR is the main condition checking and histogram incrementation routine. It is presently (July 1990) called XDATA_SUN.FOR

2. If the pseudo parameter numbering gets changed, *e.g.* due to the addition of a large number of additional parameters, the MULT_ENTR routine may require modification. This routine generates the pseudo parameters used to make multiple entry histograms. It is called XMULTENTR.FOR; there is also a FORTRAN INCLUDE file called PSEUDO_TABLE.INC which contains the raw parameter numbers in an encoded form. The first two digits of these encoded numbers flag either time (02), ΔE (04), Energy (06), or undefined (08). The upper three digits are the computed parameter number (see sub-section 10.5.3)

3. If additional pseudo parameters are required (*e.g.* particle identification with silicon detectors), a routine PSEUDO should be implemented. This can have any (non-conflicting) file name. A skeleton routine should be found in XPSEUDO_4PI.FOR; other examples may be found with file names such as XPSEU_93%%%.FOR, where 93%%% is a 5-digit NSCL experiment number. At present, the MWPC position calculation in done in the pseudo code. The MWPC file is called XPSEU_PPAC.FOR – additional pseudo parameters may be added on to the end of this code if desired. See section 10.5.5 below and the XSARA user's guide for information on how to set up user-defined pseudo parameters.

4. The number and location of Silena ADC's has to be defined. This is done in the FORTRAN INCLUDE file MODULE_LOCATIONS.INC (if many Silenas are used, you may need to change the parameter number mapping in DATAR).

5. Array sizes for the maximum number of parameters, histograms of conditions may need to be changed. These are defined in the INCLUDE files COMBLK%%.INC

6. Once all the changes to the code have been made, a rebuild of the program is done with the command procedure REBUILD.COM

7. Any special parameters should be labeled *via* the file
   SPECIAL_LABELS.DAT
   (or some other filename that is logically assigned to 'SPECIAL_LABELS').

8. Additional histograms must be put in the one of the "standard" definition files. An example to start from is FATED.DEF

10.5.3 Parameter Numbering Scheme

XSARA computes parameter numbers from the crate, slot number, and sub-address of the FERA words. The algorithmis:

$$368 * (cn - 4) + 16 * (slot - 1) + sa + 1$$

where *cn* is the crate number, *slot* is the slot number, and *sa* is the subaddress. This formula gave unique parameter numbers only for crate numbers greater than 3, slot numbers to 231, and sub-addresses up to 15 (assuming *sa* starts from 0). ADC's in other locations could be mapped to "empty" locations in the valid parameter space.

10.5.4 How to Restart XSARA during an Experiment

If XSARA crashes during data taking, please write down any traceback error messages (or make a screen dump if on a WorkStation), try to record exactly what circumstances led to the crash. To recover, the DCL command `RESTART` or `@SYS$LOGIN:GOSARA` should get XSARA going again. You will need to know the name of the current definition file in use.

10.5.5 Implementing New Pseudo Parameters

New pseudo parameters should be implemented in the subroutine PSEUDO. There are example FORTRAN examples and skeletons on [4PI.XSARA.TEST]. The file names begin with the string XPSEU... Copy one of these to another filename, and use your favorite editor to add new parameters as detailed below.

New parameters may be created within the PSEUDO routine by adding the new parameter number, *p_list*, and the new parameter value, *p_value*, to the respective lists. The counter that holds the total number of parameters in the list, npar, must be incremented *before* the new numbers and values are added. Since pseudo parameters generally are constructed from existing (real) parameters in the list, which may occur at different places in the list for each event, one has to first loop through the list to pick the required real parameters up (if they exist at all). An example piece of FORTRAN code to illustrate this is included in the XSARA documentation, along with a list of presently-defined pseudo-parameters.

The PSEUDO routine has an argument "reject", which is passed back to the calling routine DATAR. If reject is returned .TRUE. by PSEUDO, then DATAR will skip condition checking and histogram binning for the event being processed. Note that the reject flag should be set .FALSE. for each "good" event; DATAR will otherwise remember the last setting.

Once you have created or edited a new source for the pseudo routine you need to compile it and link with the rest of XSARA to make an executable image.

It is preferable to:

1. Set default to DATAQ:[4PI.XSARA.TEST]
2. COPY or RENAME your pseudo file to this directory.

3. Type `@[-]REBUILD` *name-of-pseudo-file.*

4. COPY or RENAME the executable XSARA.EXE back to your own directory.

However, you need access privilege to DATAQ:[4PI] in order to do all this. If you don't have the required privileges, stay on your default directory and type:

`@DATAQ:[4PI.XSARA]XSARAMENU.LNK` *pseudo-name*

where *pseudo−name* is the name of the object file containing your pseudo routine. This will only work if all required object files happen to be on [4PI.XSARA.TEST] and if XSARAMENU.LNK has been kept up-to-date.

10.5.6 Auxiliary Programs

In DATAQ:[4PI.XSARA] (or sub-directories thereof) there are the following auxiliary support programs for XSARA:

- CRATE_LAYOUT.PAS
  This Pascal program generates a table of XSARA's parameter number for each slot and subaddress of either crate 4, 5, or 6. Phoswich element (ball and forward array) are indicated by 3-character labels.

- DATA_GEN.FOR
  This program is used to generate pseudo data as disk file, that can be used to debug XSARA via the DISK playback option.

- MAKE_DEF.FOR and MAKE_DEF_FA.FOR
  These programs generate the histogram section of XSARA definition files for the ball and forward array. They are kept in the XSARA CMS library.

- MAKE_PSEUDO_TABLE_FA.FOR This program makes the Fortran DATA statement for XMULTENTR.FOR.

- TEST_CRK.FOR This is used to decode FERA and Silena header and data words. The user enters the header word (in decimal or hexadecimal) and the program types out the crate, slot, and word count. The user is then prompted for the data words, which are decoded into subaddress and data.

10.5.7 Additional References:

- *"XSARA — A parameter driven version of SARA for the 4-pi array"* by John S. Winfield. This document is kept as 4PI-SARA.MEM on DATAQ:[4PI.XSARA.DOC]
- *"The SARA data analysis program"* by Brad Sherrill and John Winfield. Copies of this are kept in the NSCL computer documentation library.
- *"VAX DEC/CMS Code Management System — User's introduction and reference manual"* (Digital Equipment Company).

10.6 Scaler Program

The on-line SCALER process receives scaler and control buffers from ROUTER. It displays the scaler values either in counts per second or as a running total. The total is cleared at the start of a new run. The 4π uses 32-channel LeCroy 4434 scaler modules. Depending on the number of scaler modules available scaler inputs exist for almost all detectors and in addition, the Master, Master.Live, Current Integrator and other important scaler sums. A 'generic' scaler definition file is kept on DATAQ:[4PI.CAMAC] as GENERIC.SCA

Scaler References:

- Scaler Buffer Format
  See *"DAQ Event Tape Buffer Structure for NSCL"* by Ron Fox.
- Data Positioning in Scaler Buffer
  See the scaler definition file "xxxxx.SCA" from the desired experiment.
- Starting up the Scaler Process
  See section 10.3 *"Starting up Data Acquisition"*.
- Creating a New Scaler Definition File
  See *"DAQ Program Generator Guide"* by Ron Fox and Alan Grover.

Chapter 11

Data Reduction and Analysis

"That's not a bug, it's a feature!"
— multiple authors

11.1 Data Reduction

11.1.1 Introduction

This chapter of the 4π User's Guide addresses the off-line reduction of data. Analysis of the data will be discussed briefly in section 11.8. Due to the complexity of the detection system and the large number of individual devices, a highly structured routine must be developed. The process proceeds as follows: First, the raw data words which were written to tape during the experiment, are *sorted* and written to spectra, organized by detector number. Second, the sorted spectra are *matched*, one at a time, to a master template specific to each sub-array. In this step, the conformal mapping parameters are determined and stored in a *parameter* file (*e.g.* .LNS, .BLNS, etc.). Such master templates are created during a *calibration* process, which should happen when a new detector is built, and whenever significant changes in the operation of the detector are made. Third, the raw data tapes are *reduced* and translated onto *physics tapes* which contain values of physical interest. And finally, physics tapes are *checked* and verified to assure that no errors have been introduced during the data reduction. At the completion of these four steps, the raw data provided by the acquisition system will have been translated into *physics tapes*, and will be ready for analysis (section 11.8).

Having completed the reduction of the data and created the new set of *physics tapes*, the data can now be analysed for content of interest to the physics community. Details concerning the physics analysis that can be performed upon a set of experimental data is beyond the scope of this User's guide. The pre-analysis described here is a necessary step for any more

detailed study and is general enough not to introduce biases in addition to those already introduced by the detector acceptances. The pre-analysis is very CPU intensive, therefore, by taking an intermediate step (creating physics tapes) one saves in overall computing time. Generally, multiple analysis will be performed on any one set of experimental data; the pre-analysis is only performed once.

Most of the programs described in this documentation are best run via command files. In these cases, the user will, with a very high probability, find sample command files in the same directory with the executables.

11.1.2 Software Modifications

Since the 4π is a growing and active research group, there is continual change in the hardware and software configurations, both on the detector side and the analysis/computing side. For this reason, changes are inevitable. If and/or when modifications to any of the programs described in this document are necessary, the user should copy the current version to his/her own working area, make the necessary modifications (preserving retro-usability where practical), and then return the new version to the "public" area. On such an occasion, the user should also give appropriate warning to the rest of the group to any such changes.

11.2 Sorting Raw Data : Creating Spectra Files

11.2.1 SORT_4PI

Input : Raw Data (tape)

Output: An unformatted spectra file, I*2 (128,128,170)

Tentative Location: DAQ cluster, [FOURPI.RAW]

The sorting program appropriate to most 4π applications (*e.g.* PIDMAKER_4PI, BRAGGMATCH) is currently named SORT_4PI. Due to changes in record structure of the raw data tapes, the current version of the SORT program may require small changes (*e.g.* to SORT_4PI.INC) before sorting older data sets (*i.e.* before 1993). Several specially modified (and obsolete) versions exist for analysis of tapes from earlier experiments (*i.e.* experiment 88012, 88014). These versions have names such as: SORT_88012, SORT_BRAGG,...

The first step in the pre-analysis of the data is the production of spectra of the raw parameters. For each of the phoswich detectors, there are three parameters that should have

been recorded to tape, the *fast* QDC, the *slow* QDC, and the TDC. The gate on the fast QDC is set up to maximize the integration of the signal from the ΔE scintillator. The gate for the slow QDC optimizes integration for the stopping (E) scintillator. The TDC signal is related to the time during the course of an event that an individual detector fired. Two dimensional histograms of the fast versus slow QDC signals produce spectra that resolve the different particle types. Figure 7.1 displays a typical fast versus slow histogram.

The program is familiar with the record structure of the data words recorded on the raw data tapes. Each data word on the tape is proceeded by a header word that identifies which FERA channel was read. The FERA channel is then associated with the proper detector number. For each event, a check is made requiring that signals be present from the TDC, the fast QDC, and the slow QDC for each detector that registered a hit. If this requirement is met, the histograms for that detector are incremented. SORT_4PI can sort phoswich ΔE vs. E spectra, BCC E vs. phoswich ΔE spectra, BCC E vs. Z spectra, and Time spectra.

The dimensionality of the output spectra file can change according to specific demands. To be specific, BCC/Phoswich sorting will produce a 170-spectra file (128 resolution), while sorting for other specialty applications (*e.g.* PIDMAKER_4PI) can produce any size file the user defines via modifications to SORT_4PI.INC. PIDMAKER_4PI requires, for example, 512 resolution, and can read files with up to 24 spectra.

Subtleties of SORT_4PI

• Due to its flexibility, which should be preserved, users must be aware of specific settings in the include file, SORT_4PI.INC . Changes in resolution or number of spectra, described above, are implemented by changing the appropriate parameter values found in the include file SORT_4PI.INC.

• SORT_4PI can be given time cut (.TGATE) files as input, if the user desires to implement these cuts at the matching level. These files are specified in 512 resolution.

11.2.2 ADF

Input : Raw Data (tape)

Output: A compressed 2D spectra file

Tentative Location: USER_LOAN2:[JETS.DALI.VAX/AXP.ADF_LIB]

While SORT_4PI is appropriate to most 4π applications, matching with MTK requires a specific sort routine. This is due to the fact that DALI takes as its input compressed

2-D (.C2D) spectra files. ADF is a specialized sorting routine which will create these files. It is part of the larger DALI package, as is MTK, which is already well documented. Its documentation is located on the World Wide Web (URL http://nuchm2.chem.sunysb.edu/-dali/dali_l.html).

ADF is a menu driven program, which should be run, like most other tape-intensive applications, in batch mode, via a command file (*e.g.* SORT_DALI.COM). Such command files are available in various forms on the VAX and DAQ clusters; in fact, the user can most likely find one on the DAQ cluster in the [FOURPI.RAW] area. For a further description of ADF, the user is directed to view the WWW documentation.

11.3 Determination of the Response Function

FIT

Input : Calibration data file(s)

Output : Response parameter file(s), TOPDRAWER calibration plots

Tentative Location: USER_LOAN3:[FOURPI.CALIBRATION.ZDD/MFA]

Calibrations are a standard step necessary in the addition of any new detector. Following the addition of the ZDD, a program called FIT and its companion COMPARE were written to assist in this process. As with any calibration, the necessary inputs include a known and an unknown set of data points.

The purpose of FIT is to find the response parameters that best fit the data. It does this for one function at a time (*e.g.* ΔE or E). To do this, the user must execute the following steps:

- Create the input .DAT files containing the calibration data
- Program the desired function into a subroutine (*e.g.* DE_FUNC.FOR or E_FUNC.FOR)
- Give FIT an initial guess for the parameterization

The successful completion of the above steps does not, unfortunately, guarantee a successful calibration. The engine that drives FIT is an IMSL library subroutine called BCONF.

It is a bounded minimization routine that manipulates a set of parameters (the response parameters) to minimize the value of a user-defined function (χ^2, as calculated by CALC_CHI). The boundaries on the parameters will affect the resulting fit, as will the initial coordinates. If BCONF finds a local minimum, it will stop (that counts...). It is the experimenter's duty to verify that the output of FIT is the BEST fit of the response function.

The output from FIT comes in two forms. The resulting parameters and corresponding χ^2 value are written to a file called FIT_PARAM.TMP. In addition, FIT is also programmed to create a TOPDRAWER file for displaying the results of the fit (via the subroutine TD_PLOT).

11.3.1 Some subtleties in FIT

Fitting over ranges (excluding some points) has been achieved by the insertion of the necessary conditions in CALC_CHI. The variance for values that are to be excluded are simply not added to the running sum.

The initialization of the REAL and INTEGER parameters for BCONF is done by another IMSL routine UMINF. These parameters control qualities of the fit, such as Maximum Number of Iterations, Step Size, Fit Tolerance, et al. Changing these parameters will also change the results of the fit.

11.3.2 Companion Programs

COMPARE

Input : Response parameter file

Output : χ^2 value, and a TOPDRAWER calibration plot

Tentative Location: USER_LOAN3:[FOURPI.CALIBRATION.ZDD/MFA]

COMPARE calls the same subroutines (CALC_CHI, TD_PLOT, DE_FUNC, E_FUNC), but it does not call the IMSL libraries. It simply reads response parameter files (in the FIT_PARAM.TMP format) and generates the χ^2 value and plots. The purpose of this program is to aid the user in finding initial coordinates for FIT, and also in visualizing the effects of the various parameters in the response function.

This routine has been used twice successfully for calibrations (ZDD and MFA), and is also currently (July 1995) being used to compare data to theoretical predictions of radial flow. The code is extremely versatile, as it is in modular form, so that new applications of FIT require modifications to only select modules (*e.g.* DE_FUNC, CALC_CHI).

11.4 Making Calibration Tables

11.4.1 PIDMAKER_4PI

Input : Sorted spectra file (512 resolution), stripped energy loss files

Output : Ztable.TMP, Atable.TMP, MeVtable.TMP, ZPID.TMP, PUNCH.TMP

Tentative Location : USERLOAN3:[FOURPI.PIDMAKER]

In order to create physics tapes, one must first create calibration tables (Z/A/MeV tables and ZPID files) which the PhyTape code will use to assign PID and energy to the particles. The creation of these tables (for phoswich type detectors) is done with PIDMAKER_4PI. PIDMAKER_4PI is (basically) menu-driven, and can be used to create new tables, or modify existing tables. The menus are fairly self-explanatory.

The input to PIDMAKER_4PI is a sorted spectrum file which is generated by SORT_4PI. Currently (July 1995) PIDMAKER_4PI will only run on VAX stations, due to its embedded UIS calls, although translation work of this type is already underway for BRAGGMATCH (these are in fact similar programs). The basic task of the user is to map a generic spectrum to some chosen standard form, and then draw lines on that standard map which will be used to denote the particle species. From these *gatelines* that the user has drawn, PIDMAKER_4PI will determine Z and A for each pixel in a 512 x 512 array, which corresponds to the locations in a mapped spectrum. It will also read stripped energy loss (*e.g.* DONNA) files to create an energy table, which is referenced by Z, A, and X (mapped E)- coordinate. The user can create such stripped files from standard energy loss files via the program STRIP, described in section 11.4.2.

With the advent of new detectors, serious changes to PIDMAKER_4PI will hopefully only be necessary in the subroutine TABLEMAKER, where PID is assigned according to a pixels position between drawn ZPID lines. However, minor changes will certainly be necessary elsewhere (*e.g.* everywhere Nz is transformed to Ng or vice versa).

The output of PIDMAKER_4PI is in the form of 5 files:

- ZPID.TMP : Unformatted, written from the array ZPID, declared as REAL ZPID(100,0:512). Carries the Y coordinate of each of the ZPID lines (100 lines max), given its X coordinate.
- ATABLE.TMP : Unformatted, written from the array ATABLE, declared as BYTE ATABLE(0:512,0:512). Carries the atomic mass assigned to each pixel.

- ZTABLE.TMP : Unformatted, written from the array ZTABLE, declared as BYTE ZTABLE(0:512,0:512). Carries the charge assigned to each pixel.

- MEVTABLE.TMP : Unformatted, written from the array MEVTABLE, declared as INTEGER*2 MEVTABLE(0:18,0:40,0:1028). Carries the Incident energy assigned to a particle with a specific Z, A, and X (mapped E)- coordinate. PUNCH.TMP : Formatted, written values are extracted from the array MEVTABLE. This contains a list of punchin and punchout energy values, based on the premise that these occur at channels 1 and 1028, respectively.

Companion Programs

ADD_SPEC

Input : Two (or more) sorted spectra files

Output : One sorted spectra file

Tentative Location : USER_LOAN3:[FOURPI.PIDMAKER]

The purpose of this program is to add the contents of two or more sorted spectra files into one file. This is useful for viewing multiple energies at once.

READ_SPEC

Input : One sorted spectra file

Output : A PAW histogram (.HST) file

Tentative Location : USER_LOAN3:[FOURPI.PIDMAKER]

The purpose of this program is to create a histogram file viewable in PAW. This is useful for creating "pretty" pictures of spectra, and otherwise viewing them when one does not have access to a UIS compatible machine.

READ_TABLES

Input : Calibration tables (Atable.TMP, Ztable.tmp, MeVtable.TMP, ZPID.TMP

Output : A PAW histogram (.HST) file

Tentative Location : USER_LOAN3:[FOURPI.PIDMAKER]

The purpose of this program is to create a histogram file viewable in PAW. This is useful in that, without such a program, there is no other way to view the contents of a calibration table. Once inside PAW, the user can verify the values stored in these arrays.

There is (likely) a very useful .KUMAC file in the same directory with this program called TABLES.KUMAC, which makes use of PAW's STYLE GP interface for the purpose of viewing these tables.

STRETCH_ZPID

Input : One ZPID file

Output : A modified ZPID file

Tentative Location : USER_LOAN3:[FOURPI.PIDMAKER]

The purpose of this program is to apply a linear stretching factor to the contents of a ZPID file, thus allowing the user to, *e.g.* , add more ZPID lines to an existing template.

TRANSFORM

Input : One MeVtable (any size/resolution)

Output : An MeVtable of a different resolution

Tentative Location : USER_LOAN2:[JETS.DALI.TOOLS]

The purpose of this program is to transform low resolution (512) MeVtables into high resolution (1024/1028) MeVtables, and vice versa. This is useful since programs like PID-MAKER_4PI and TMAKER create, by default, 1028 resolution MeVtables, which are fit for use by the DALI package. However, PHYTAPE, and other programs, require 512 resolution tables.

RESOLUTION

Input : A sorted spectra file (512 resolution)

Output : A modified spectra file (lower resolution)

Tentative Location : USER_LOAN3:[FOURPI.PIDMAKER]

The purpose of this program is to condense sorted spectra into lower resolution images. This is useful since such a change would otherwise require running another batch job. This allows the user to sort once (at high resolution) and view the file in any resolution thereafter.

11.4.2 TMAKER

Input : BCC gateline (.GBC) files

Output : BCC calibration (Z/A/MeV) tables, gateline (ZPID) file, punchin/punchout file

Tentative Location : USERLOAN3:[FOURPI.TMAKER]

The purpose of this program is to create calibration tables for BCC E vs. phoswich ΔE spectra. The nature of these tables is the same as those created by PIDMAKER_4PI (section 11.4.1. The principle is also the same as that described for PIDMAKER_4PI, with one primary exception: drawing gatelines is done automatically in TMAKER. The locations of the gatelines are read from the .GBC files, so that TMAKER simply copies those files into a LINES file, and generates the Z/A tables according to the locations of these lines. The values written the MeVTABLE are also read from the .GBC files, with the small exception that TMAKER must call an interpolation routine in order to assign energy values to each channel appropriately.

Companion Programs

GATESPEC

Input : Stripped energy loss (.EBC) files

Output : BCC gateline (.GBC) files

Tentative Location : USERLOAN3:[FOURPI.TMAKER]

GATESPEC reads stripped energy loss files, generated for media including the BCCs and the phoswich "fast" plastic, and then applies the two respective response functions to determine the signals produced (measured in channels) in the BCC and ΔE detectors. GATESPEC thus does "all the work" in locating the gatelines and determining the incident energies corresponding to the points along those gatelines, writing those values to the .GBC files.

Input consists of several stripped energy loss files, and output consists of two gateline (.GBC) files per particle type (one "integer" and one "half-integer" type for each particle). The "integer" files are as one might expect: straightforward energy loss calculations for that particle. The "half-integer" files are simply energy loss calculations done for "half-integer" charges and masses (*e.g.* Z=2.5, A=5.5; Z=3.5, A=8.5; etc...). The half-integer lines are the lines that eventually become the gatelines, the logic being that Z=2.5 should fall right between Z=2 and Z=3, which it does fairly well.

The output files contain the X (ΔE) and Y (BCC E) coordinates of points along the gatelines, in addition to the corresponding incident energies for those particles.

When running GATESPEC, the user will be prompted for the location of the stripped energy loss (.EBC) files, and his/her disposition towards errors. To explain, while converting energy deposition into channel numbers, errors often occur, sometimes resulting in bad sequences (energy momentarily decreases) and even negative values (yes, even negative values). The user will be notified when errors like these occur (both interactively and in a .ERR file), and GATESPEC can even fix these errors automatically, if the user so desires.

STRIP

Input : One (master) energy loss file

Output : Stripped energy loss (.EBC) files

Tentative Location : USER_LOAN3:[FOURPI.TMAKER], USER_LOAN3:[FOURPI.PIDMAKER]

The program STRIP will read an energy loss file and break it up into all of its constituent pieces. Specifically, one generally creates an energy loss file for multiple ions. STRIP will search for the ION headers, identify the ion, and write the data for that ion to a file named after its charge and mass. The user can choose to add "padding" zeroes in the name of the output files, and also must choose the output file specification. Typical names are of the form Z010A010.EBC with "padding", Z1A10.EBC without.

11.5 MATCHING

11.5.1 MTK

Input : Sorted, compressed 2-D (.C2D) spectra file(s)

Output : Parameter (.LNS / .BLNS) file(s), average energy (.AVG) files, energy cutoff (.ECUT) files, BCC gateline (.PNCH) files

Tentative Location : USER_LOAN2:[JETS.DALI.VAX/AXP.MTK_LIB]

The process of determining the parameters necessary to map the raw spectra for each detector into a standard template, created in the calibration process, is called *matching*. The matching of the sorted phoswich spectra is currently (June 1995) done with MTK, part of the DALI package developed primarily at SUNY-Stony Brook by Jérôme Lauret. It already has extensive documentation available on the World Wide Web (URL http://nuchm2.chem.sunysb.edu/dali/dali_.html) and online help as well, accessible by running the desired program with the option "-help".

Matching for the 4π data here at the NSCL is primarily concerned with two programs in the DALI package: MTK and ADF. MTK is the matching program itself, which can match phoswiches, draw BCC gatelines, and match BCCs (not yet available), while ADF is the sorting program which creates the compressed 2-D histogram files (.C2Ds) that MTK displays while matching (section 11.2.2). MTK is run via the DCL symbol RMTK. Thus, as mentioned above, for online help, the user can type $ RMTK -help .

While matching, there are some basic diagnostics that the user has at his/her disposal. There are two particularly helpful options at the user's disposal while matching. The first is the "Energy Projection" option. By choosing this option, then choosing a PID band in the matched spectrum, the user will be shown an energy spectrum for particles in that band. MTK will pop up a window that contains the PID for that particle, the energy spectrum for that PID band, and the average of the energy spectrum. This is useful in checking for correct assignment of PID, and also for reasonable qualitative energy distributions. One specific problem this projection helps to identify is contamination of the proton line, which will be addressed below in a discussion of ECUT energies.

The second helpful online option at the user's disposal is the "Project All Energies" option. By choosing this option, MTK will calculate the average energy for p, d, t, and α particles (α, Li, Be, B particles for the MFA and ZDD) and write them both on the screen, and to a file. In this way, the user can ensure that detectors in the same azimuthal ring have similar average energies, and that detectors in successive rings have smoothly decreasing

energies with increasing Θ. Upon exiting the program, these average values will be written to an average (.AVG) file.

MTK gives the user the ability of choosing energy cutoff (ECUT) values while doing the phoswich matching. Some raw spectra, once matched, show contamination of the protons by the neutral line. Above the energy at which this intersection occurs, the proton energy spectra would (without any further action) become distorted. By placing an ECUT at this point, the user allows the PHYTAPE program to assign a specific label to protons whose energy is above the ECUT energy, thus eliminating such neutral particle contamination.

Subtleties of MTK

• Running MTK, and any of the other DALI routines, requires an expansive number of symbol and logical definitions. This is done via two calls to DCL command files. The first is called DALI_LOGIN.COM, which currently lives on USER_LOAN2:[JETS.DALI]; this sets up the basic symbols and logicals for the DALI environment. The second reflects the date of the most recent changes to the calibration tables (ZTABLE, ATABLE, etc.); this tells DALI the directory specifications where it can find these tables. These secondary setup files are located in the CENTRAL area (USER_LOAN:[JETS.DALI.CENTRAL...]), and they typically will change the definition of three logicals: CENTRAL, which contains detector maps and such; PIDMAKER, which holds phoswich calibration tables; and BCSTABLES, which holds BCC calibration tables. The current file of this type is called APR95.COM.

• MTK assumes a default location for the calibration tables (Z/A/MeV tables, and ZPID files). If those files are not in the default location, or if they do not have the same (generic) names as the defaults, then the user must create a configuration file to tell MTK where to find the right ones. This can be done by running ADF, and choosing the options (M)ISCELLANEOUS, then (G)ENERATE. This will create a file called ADF_CONFIGURE.DAT in the current directory. The user is then free to change any information (names only, not numbers) necessary to set the record straight. Partial configuration files are also permissible, if the user has only changed *some* of the calibration files...

• The .C2Ds created by ADF, although not so large on disk, are very large in memory, and thus require a VERY large pagefile quota to run MTK (especially to do BCC matching, in which case the phoswich .C2D **and** the BCC .C2D are both loaded simultaneously). Phoswich matching with MTK requires roughly 60K pages, while BCC matching with MTK requires roughly 200K pages !!! GREAT CARE must be taken when running MTK (especially for BCC matching) not to overdraft the pagefile.

The MTK user should type SHOW MEMORY at the DCL prompt to get a report on the current status of the pagefile for the current node. The statistic of interest is the

RESERVABLE amount across from the label DISK$SYSTEM...[]PAGEFILE.SYS . That number is the number of pages that any NEW applications can reserve on that node. It can go negative with no observable ramifications. The number of FREE pages (also across from ...PAGEFILE.SYS) is the current actual use on that node. This number CAN NOT go negative, and a 0 here cripples the node.

Companion Programs

C2D_TOOLS

Input : One or more compressed 2-d spectra (.C2Ds)

Output : Modified .C2Ds, or unformatted spectra files

Tentative Location : USER_LOAN2:[JETS.DALI.TOOLS]

C2D_TOOLS is a utility that was written to manipulate .C2Ds, the compressed 2D spectra created by ADF. Its features include the following:

Add .C2Ds : Add the contents of two .C2Ds, resulting in one with greater statistics, and/or different energies.

Splice .C2Ds : Join part of one .C2D with part of another. The user determines the ranges to merge in each file.

Copy a .C2D : Copy part (or all) of an existing .C2D to a new .C2D.

Extract a spectrum : Extract a 512 x 512 spectrum from an existing .C2D and write it to an INTEGER*2 unformatted file.

CHECK_PUNCH

Input : BCC Punchin (.PNCH or .BLNS) and Phoswich (.LNS) parameter files

Output : Data (.DAT) file listing the "Offenders", possibly a modified punchin (.PNCH) file

Tentative Location : USER_LOAN2:[JETS.DALI.TOOLS]

The PHYTAPE code reads as input two particular parameter files of concern here. They are the phoswich matching (.LNS) files, created by MTK (or MATCH), and the punchin (contained in either .PNCH or .BLNS format) files. The concern lies in the fact that the decision to identify particles in the ball via BCC E vs. phoswich ΔE or phoswich ΔE vs. E is made based upon the "gateline", whose slope and intercept are contained in the .PNCH file, while particles which fall to the left of the "punchin" line, whose slope and intercept are contained in the .LNS file, are unidentified by the phoswich ΔE vs. E method. It is for this reason that the program CHECK_PUNCH was written.

CHECK_PUNCH reads both the gateline and punchin line parameters and checks to see if the gatelines are to the right of the punchin lines. For those cases where the lines cross, it reports which end (top/bottom) is misaligned, and for those cases where the places of the lines are completely interchanged, it reports these as "Dual Offenders". Upon execution, the user will be given the option of 1) reporting problems only, or 2) reporting and automatically fixing problems.

The user will also be prompted for whether the .PNCH file was generated by MTK or BRAGGMATCH, as the format is somewhat different, and whether the .LNS file is for 215 or 300 detectors, so as to read the file with the appropriately sized array. The .LNS file is assumed to be created by MTK.

A summarized report is shown on the screen, while an itemized report is written to the file CHECK_PUNCH.DAT .

TWEEK

Input : Phoswich (.LNS) or BCC (.BLNS) parameter files

Output : Modified parameter files, or ASCII (.SAV) files

Tentative Location : USER_LOAN2:[JETS.DALI.TOOLS]

The output of the MTK matching program is in the form of unformatted .LNS (and .BLNS) files. This can be frustrating to those of us who like to see and touch everything :) . TWEEK was created to allow the user to "tweek" or view the values in the .LNS files by performing a few basic operations:

1. Merging two parameter files
2. Tweeking (changing) individual parameters via direct keyboard input
3. Viewing parameters in unformatted files

4. Converting small (215 det) parameter files to large (300 det) ones

5. Splitting/Merging .BLNS files and .PNCH files

Option 1 is useful, for example, in joining work that has been done one separate occasions. Option 2 is useful for those occasions when the user knows exactly what matching values he/she wants (MTK does not allow direct entry of parameter values... only mouse matching). Option 3 is not just for curious people, although they will certainly like it, but it allows the user to execute a DCL level DIFFERENCE command on two .LNS files to see clearly any differences between the two. Option 4 simply enlarges small parameter files by adding zeroes at the end, while preserving the unformatted structure of the file. Option 5 allows the user the freedom to specify BCC gateline parameters in either the .BLNS file or a .PNCH file.

AVGPARS

Input : Average energy (.AVG) files

Output : Averaged (by ring) average energy (.AVGP) files

Tentative Location : USER_LOAN2:[JETS.DALI.TOOLS]

The average energy (.AVG) files created by MTK list the energies of p, d, t, and α for every detector. Since it is the *matcher*'s job to ensure that these are the same (if not similar) for all particles in a ring, it is reasonable to want to see a single average of those energies for each ring. This makes it easier to identify the presence (or absence) of a smooth decrease in energy with increasing Θ. AVGPARS simply reads the .AVG file, calculates a single set of averages for each ring, and writes them to the .AVGP file.

11.5.2 BRAGGMATCH

Input : Sorted spectra files (phoswich and BCC)

Output : BCC parameter (.BLNS) file

Tentative Location : USER_LOAN3:[FOURPI.BRAGGMATCH]

Bragg Curve spectra matching is currently (July 1995) done via BRAGGMATCH. Although MTK is advertised to handle BCC matching, the bugs have not yet been worked out, in addition to the fact that it has rather unfriendly pagefile quota requirements.

The task of matching BCCs via BRAGGMATCH is similar to that of phoswich matching, but the software is (obviously) different. BRAGGMATCH is predominantly a mouse-driven program, although occasional keyboard input is necessary. Input to the program consists of two sorted spectra files: one for the phoswiches and one for the BCCS. These files are created via SORT_4PI. Output from the program is a BCC parameter (.BLNS) file, also similar to the phoswich matching.

The current software uses UIS calls, and thus, can only be run on VAX stations. However, the task of converting this into a MOTIF-based program for VAX and ALPHA is underway. (Thanks John!).

11.5.3 MATCH (Obsolete)

Input : An unformatted data file, I*2 (128,128,215)

Output: A parameter file, real (6,215)

The old gain matching program is called MATCH. In addition, a companion program that is used in tandem with MATCH is called PRM_CHECKER. Since MATCH reads in the output from the SORT program, it is not affected by changes that are made to the buffer structure of the raw data tapes. Modifications to the output of SORT will, however, necessitate similar changes in MATCH.

The program reads in the large data file created by SORT_4PI (this file reading operation takes about five minutes). Operation of the program is almost completely controlled through the mouse attached to the VAX workstations. Only when the program requires the input of a filename, or when the normal flow of analysis is disrupted, is the operator asked to provide input from the keyboard. Generally, each question asked by the program will have three possible responses which correspond to the three buttons on the mouse. The LEFT button generally moves one forward through the analysis, while the RIGHT button generally allows one to undo the last operation. Verification is required at each step before the program proceeds, this tends to slow the analysis down a bit, but it reduces the number of mistakes.

The program starts by asking the name of the parameter file. If the input name is not found, the program will ask if this is to be a new file. The program next asks for the name of the energy cut file – as before, if it can not find the file, it asks if it should create a new file by that name. The program then asks for the name of the data file. After the data file is read in, the analysis can begin.

MATCH assumes that one wishes to start with detector number 1. Pressing the LEFT button will allow one to proceed with analysis of detector number 1, pressing the MIDDLE

button will allow the operator to input a detector number from the keyboard, while pressing the RIGHT button will tell the program to analyze the detector number one lower than it expects (normally, upon completion of a detector the program will advance to the next detector number, therefore the RIGHT button tells the program to go back and redo the last detector. For the case where one has just started the program, this then tells the program to analyze detector number 0. Detector number 0 is a flag for writing out the current state of the parameter file and asking if the operator wishes to terminate the analysis session.) Assuming one tells the program to analyze detector number 1, the program will first display the previous *punch-in line* and ask if that line is OK.

Checking the Lines File

A new program was written for the 88012 experiment to check the lines file and provide feedback for improvements to the person running the MATCH program. The program is called PID_CHECK and is located in [FOURPI.88012.SPECTRA]. It reads in 36 spectra at a time that have been created by the SORT_88012 program in the area [FOURPI.88012.MATCH]. This SORT program is run in exactly the same way as the other versions of SORT. The spectra are of size (512,512) which matches the size of the lookup tables used in PHYTAPE routines. The program is run on a workstation and creates two windows; one with the raw spectrum, the other with a colour-coded PID map superimposed on the raw spectrum that has been multiplied by a user-input scale factor. Depending on the quality of this superposition, the user can OK the match, or suggest various changes. The user can re-examine the spectrum with a different multiplicative factor. At the end of 36 detectors, the program lists the comments made by the user in a file called COMMENTS.DAT. This must be renamed if the user wishes to keep it. The process is repeated for the next 36 detectors until all 215 have been checked.

A further check is to create a physics tape and then analyze the energy spectra of a particular fragment. The code [FOURPI.CRAIG.SPECTRA]FOURPI.FOR does this and then sorts the histograms into groups that have constant Θ but different Φ. By checking that the energy spectra match, the user has an internal consistency check on the calibration.

TOF spectra

Time-of-flight spectra can be created for all 215 detectors with the program SORT_TOF in the area [FOURPI.88012.MATCH]. This program is run in the same way as all other SORT programs. The TOF histograms it creates can be selected to include all particles, those near the neutral line or those near the punch-in line. The user is strongly advised to check the

source code to control the selection. The program requires a TDC calibration which is stored in the file TDC_FIT.DAT. This has been obtained by changing the delay on the common stop in the timing circuitry of the 4π electronics.

Once the histograms are created the user can automatically place lower TOF cuts with the program TDC_GATE in the area [FOURPI.88012.SPECTRA]. The program finds a peak in the spectrum and places the lower cut a certain number of channels lower. The program produces a file TDC_CUT which must be renamed with an extension .TOF for use in applying the TOF cuts in PHYTAP89. The routine PID_RF in PHYTAP89 applies an upper cut a set number of channels above the lower cut. Again, users are strongly advised to check the source code and fine tune the placement of the cuts for their experiment. Note also that the time calibration should be checked for each experiment.

11.6 Translation of Raw Data to Physics Tapes

11.6.1 PHYTAPE

Input : Raw data (tape), calibration tables, matching parameter (.LNS, .BLNS) files, punchin (.PNCH) file, energy cutoff (.ECUT) file, time cut (.TGATE) file

Output: Physics data (tape)

Output: Error checking (.PID) file (obsolete)

Tentative Location : DAQ cluster, [FOURPI.PHYTAPE]

The information on the raw data tapes is written in an encoded manner. It is difficult for all but the most trained users to be able to look at the data words from a raw event and to be able to know which detectors fired. For an average user to understand the information contained within these data words, the words must be decoded, translated, and interpreted. This is all done within PHYTAPE. As with the SORT program, because the record structure of the raw data tapes changes, minor changes may be required to run the current version of PHYTAPE with older data sets.

11.6.2 Buffer Structure

The physics tapes created by PHYTAPE consist of buffers currently (June 1995) containing 8192 I*2 words. For a raw buffer size of 8192 words, and a physics buffer size of 16384 words,

filling one physics buffer will typically require 4 to 5 raw buffers. The buffers are structured as follows:

/ Buffer Header / Event 1 / Event 2 / ... / Event N /

Buffer Header: The first 6 words of the buffer make up the header and contain the following information:

Word 1 : Buffer Type

1. Data Buffer
2. Begin-of-Run Buffer (which also contains data)
3. End-of-Run Buffer (which also contains data)

Word 2 : Run Number

Word 3 : Physics Buffer Sequence Number

Word 4 : Number of events contained in this buffer

Word 5 : Position of the last word in the buffer

Word 6 : A running total of the normalization information from:

$$\frac{master.live * current\_integrator}{master * 100} \tag{11.1}$$

(Note: this gives the live-time corrected accumulated charge in units of *ten* times the BIC current integrator full scale.)

Event Structure: The event is structured as follows:
/ Event Header / Detector 1 / Detector 2 / ... / Detector N /

Event Header : One word, the event multiplicity (number of detectors which fired)

Detector Structure: Each detector contains 6 data words:

Word 1 : Z, the charge of the detected fragment

Word 2 : A, the mass number of the detected fragment

Word 3 : Θ, the polar angle in degrees

Word 4 : Φ, the azimuthal angle in degrees

Word 5 : E, the particle's energy in MeV

Word 6 : I, the detector ID number

11.6.3 Modifying PHYTAPE

The PHYTAPE program will have to be modified for any experiment that uses detectors other than the standard 4π detectors. The user will have to add in sections that handle the additional detectors. Additionally, users may wish to modify PHYTAPE if they are interested in modifying the selection criteria placed upon detectors during the pre-analysis.

As with any "standard" 4π program, PHYTAPE has many users. If any changes to the program are necessary, the user should copy the current version into his/her own area, make the necessary revisions, DOCUMENT them (inform all of the other group members, as well) and then put the new, working, TESTED version back into the public domain.

Modifications to PHYTAPE introduced for 88012

There were two main groups of changes introduced for this experiment;

1) a) The particle-identification tables used by PHYTAPE were extended up to Z=18
   b) The isotopic identification for Z=1 particles in the forward array was dropped.
   c) Tighter limits were placed on the isotopic labelling of the fragments.
2) Timing cuts were introduced;
   a) The event-master-RF time signal had to be within certain limits.
   b) A cut was placed on the time-of-flight for Z=1 fragments to reduce the contribution of the strong neutral line. This needs a TDC calibration and TOF cut file for all 215 detectors.

Timing cuts can be deactivated in the command file that runs PHYTAPE89.

Modifications to PHYTAPE introduced for 93033

1. The MFA and ZDD were included.

2. The number of phoswich-type detectors (Nphos) was "generalized" (made into a FORTRAN parameter), and subsequently set to 300. Nphos is the number of positions in the relevant arrays (*e.g.* matching parameters, etc.); Nphos_actual was also generalized, and represents the *actual* number of phoswich detectors

3. A "NO Ecut/Tgate"-file option was added.

4. The input resolution of parameter files was generalized, and the "working" resolution, in INCREMENT_BUFF, for all parameters was standardized to 512.

5. Particles that are transformed off-scale in the vertical direction are now kept, and assigned Z/A as though they occurred at a value of IdE = 512.

6. Diagnostic lines to test all inputs and to track all parameter assignments (identified particles or unidentified) were introduced.

7. Tabulated (by subarray) output of "bad" particles was added to the .LOG file.

8. Particles below the first gateline (Z=0, A=-1) are no longer kept (written to the physics buffer)

9. PHYTAPE.INC, an include file containing FORTRAN parameter definitions, was introduced to facilitate easy parameter changes.

11.7 Error Checking with PID_TEST: (Obsolete)

Input : Error checking (.PID) file

Output : A list of the detectors that failed and a measure of each failure level

Former Location : USER_LOANA:[FOURPI.PHYTAP]

NOTICE:
This stage is associated with the former PHYTAPE program (PHYTAPE91028). This option is not in place in the new PHYTAPE code.

Once generated, the physics tapes must be checked to verify that the values of the matching parameters have not changed since the last tape that was gain matched and that the inputs in the command file that generated the new physics tape were correct. Gain shifts, timing shifts, adjustments to the electronics between runs, and simple typing errors

in the command file will all cause the physics tape generating program to create physics tapes containing errors. The physics tape program itself has no internal error checking and therefore, the operator must check each tape as it is created. Physics tapes should not be used unless the error checking information is recorded on the tape label. To check a newly generated tape, the operator runs the checker program, unless that tape has *Master* status. *Master* status means that the gain matching was done for this raw data tape and not simply for a data tape for the same vicinity. The checking that is done with this program is a comparison of the basic results of the newly generated physics tape to the results from the *Master* tape.

The program determines a pass/fail evaluation for every particle type for every detector. The failure criterion is that the normalized number of counts from the physics tape being tested does not differ from the number that was present in the *Master* tape by more than three standard deviations. This is a failure criterion that should be satisfied 99.9% of the time for identical normal distributions. The program checks over 2000 quantities, therefore one expects a few failures through statistical fluctuations. A systematic error between the two files will, however, generate copious numbers of failures. Individual detectors that have drifted in gain will display multiple failures.

11.8 Analysis

Input : Physics data (tape), Impact parameter file (maybe)

Output : Histogram (.HST) file

Tentative Location : USER_LOAN3:[FOURPI.ANA]

In order to process physics data, one needs an analysis code. ANA is one of several analysis codes. ANA makes use of HBOOK/CERN libraries in order to facilitate analysis with PAW. Since every analysis code is different (and changes by the day), only a brief description of its most basic features is worth including here.

Following are some basic subroutine names, and their general functions

- ANA_MAIN : Parse user input from the command file, open the physics tape file, read physics buffers, and calls the appropriate routines in sequence.
- HBOOK_SETUP : Initialize HBOOK memory and define relevant histograms, which will be filled elsewhere.

- CHAR_MAP : Parses the physics tape file name to extract system characteristics (Z_{proj}, A_{proj}, E_{beam}, etc.).
- PROCESS_EVENT : Calculates quantities of interest for each event, and increments histograms.
- IMPACT : Opens an existing impact parameter (.BS) file to determine centrality thresholds, and assigns centrality "bin" classifications to each event.
- OUTPUT : Writes booked histograms to disk in a histogram (.HST) file, and creates impact parameter (.BS) files, if so directed.

Other routines may also be used in some configurations, but, once again, this is only a basic introduction to the most common parts of ANA. Input to ANA is a physics data file, and output from ANA is a histogram file (and possibly an impact parameter file).

Appendix A

Updating This Text

A.1 How to modify this document

This document is written using LaTeXversion 2.09 in a subroutine-like structure. That is, each section is contained in a separate .TEX file and the macro *input* is used to include these sections into the main framework file USERSGUIDE.TEX. All files are located in DATAQ:[4PI.USERSGUIDE].

To make modifications of this document you must be familiar with LaTeXand have DAQ_PROGRAMMER privileges. After the desired .TEX files are modified the document should be recompiled using the commands:

```
$ LATEX USERSGUIDE
$ LATEX USERSGUIDE
$ DVI2PS USERSGUIDE
```

(The `LATEX` step needs to be repeated so that the table of contents is up-to-date). The above commands will allow other users to print out the most recent version of this manual (as described in the next section.)

If you do not have DAQ_PROGRAMMER privileges and wish to makes changes you may copy the desired files to your working area and make modifications. Then notify the person in charge of the documentation (currently A. Vander Molen, who can copy the updated files back to the DAQ area and recompile the document.

A.2 How to get the most recent version

You do not need any special privileges to issue the following commands:

- To get the date of the most recent version of this document:

  ```
  $ DIR/DATE DATAQ:[4PI.USERSGUIDE]USERSGUIDE.*
  ```

- To get a printout of the most recent version of this document, type:
  `$ PRINT/QUE=west_print guide:USERSGUIDE.PS`
  Or some other`queue` that is a print queue for a PostScript printer. For appropriate printers use the two_sided option.

A.3 Forthcoming additions

- Blue LEDs
- Phototubes and Bases
- Vault Safety