H1-11/88-96

Development of the Ray Finder Electronics for the z-Vertex Trigger for the H1 Detector at HERA

May 1988

Diploma Thesis of
Stefan Eichenberger

made in the group of
Prof. Dr. P. Truöl

Institute of Physics
University of Zurich
Schönberggasse 9
8001 Zurich

Contents

List of Tables

List of Figures

Abstract

For the first trigger level of the H1 detector at HERA, a z-vertex trigger has been suggested. Using fast multi wire proportional chamber signals, it should reconstruct the vertex location of an event along the beam axis. This report is concerned with the realization of the so called ray finder card, which recognizes rays in the data coming from six pairs of parallel planes of MWPCs. Together with the (not yet developed) vertex finder card, it forms the z-vertex trigger.

Chapter 2 describes various simulations made to test different setups for the ray finder and the influence of several suggestions for the exact geometry of the MWPCs. Input for these simulations were data obtained from Lund Monte Carlo simulations of neutral as well as charged current events.

Chapter 3 is concerned with the realization of the ray finder and shows that an application specific integrated circuit (a gate array) provides the best solution. This was dictated not only by technical but also economical reasons. Indications are given on how the gate array could be used for other applications as well.

Chapter 4 deals with timing constraints for the gate array and presents test results of measurements made on the engineering samples, which met all requirements well.

Zusammenfassung

Als Teil der ersten Trigger Stufe für den H1 Detektor am HERA wurde ein z-Vertex Trigger vorgeschlagen. Unter Verwendung schneller Proportionalkammersignale soll er die Lage des Vertex auf der Strahlachse rekonstruieren. Diese Arbeit befasst sich mit der Entwicklung der sogenannten 'Ray Finder Card', die die Daten von sechs Kammerpaaren zu sogenannten 'Rays' verknüpft. Zusammen mit einer später zu entwickelnden 'Vertex Finder Card' bildet sie den z-Vertex Trigger.

Kapitel 2 beschreibt verschiedene Simulationen, die dazu dienen sollen, Varianten für den 'Ray Finder' und für die exakte Geometrie der Kammern zu testen. Die Simulationen beruhen auf Daten aus Lund Monte Carlo Rechnungen für Ereignisse mit neutralen und geladenen Strömen.

Kapitel 3 befasst sich mit der Planung des 'Ray Finders' und zeigt, dass die Entwicklung einer kundenspezifischen integrierten Schaltung (eines Gate Array) die beste Lösung des Problems darstellt. Hierfür sprechen sowohl technische wie wirtschaftliche Gründe. Gezeigt wird ebenfalls, wie der schlussendlich entwickelte IC in anderen Applikationen Verwendung finden könnte.

Kapitel 4 behandelt Ueberlegungen betreffs des Zeitverhaltens des Gate Arrays und präsentiert Resultate von Messungen, die an den Entwicklungsmustern vorgenommen wurden. Sie entsprechen allen Anforderungen.

1. Introduction

1.1 HERA and H1

HERA is a new electron-proton collider currently under construction at DESY in Hamburg and due to be operational in 1990. The storage ring with a radius of about 800 m lets 30 GeV electrons collide with 820 GeV protons. Their respective tracks cross at two places, where international collaborations of about 25 universities and institutes each build complex particle detectors. One is called ZEUS, the University of Zurich collaborates in the H1 group.

With these particle energies a momentum transfer up to $Q^2 \approx 3*10^4$ GeV^2 can be reached. This allows to study possible substructures of quarks and leptons, since actually just one of the three quarks of the proton collide with the electron [1]. The electroweak interaction can be carefully studied and measurements done at e^-e^+ colliders can be extended. Currently, it is a point of discussion, whether HERA can be used for heavy quark production. The discovery limit for the mass of the top quark is around 70 MeV [2].

Fig. 1.1 shows the H1 detector layout. Collisions that occur on the beam axis produce secondary particles, which should be observed by the detector. It consists mainly of two parts: calorimeters, which are used to determine energies deposited in them, and tracking detectors, which should help to examine particle tracks. Part of the latter are multi wire proportional chambers (MWPCs). The detector is not symmetrical around the interaction point, since, due to the given e^- and p energies, the particle showers are mostly boosted into the forward region.

MWPCs deliver fast information (time resolution of about 20 ns) on the existence of a charged particle and give a rough estimate on one coordinate of its track with a reso-

lution of a few millimeters. Therefore, they are good candidates for first level triggers. About 2 μs later, drift chamber data will give more accurate information on the actual track with a resolution of 100 μm.

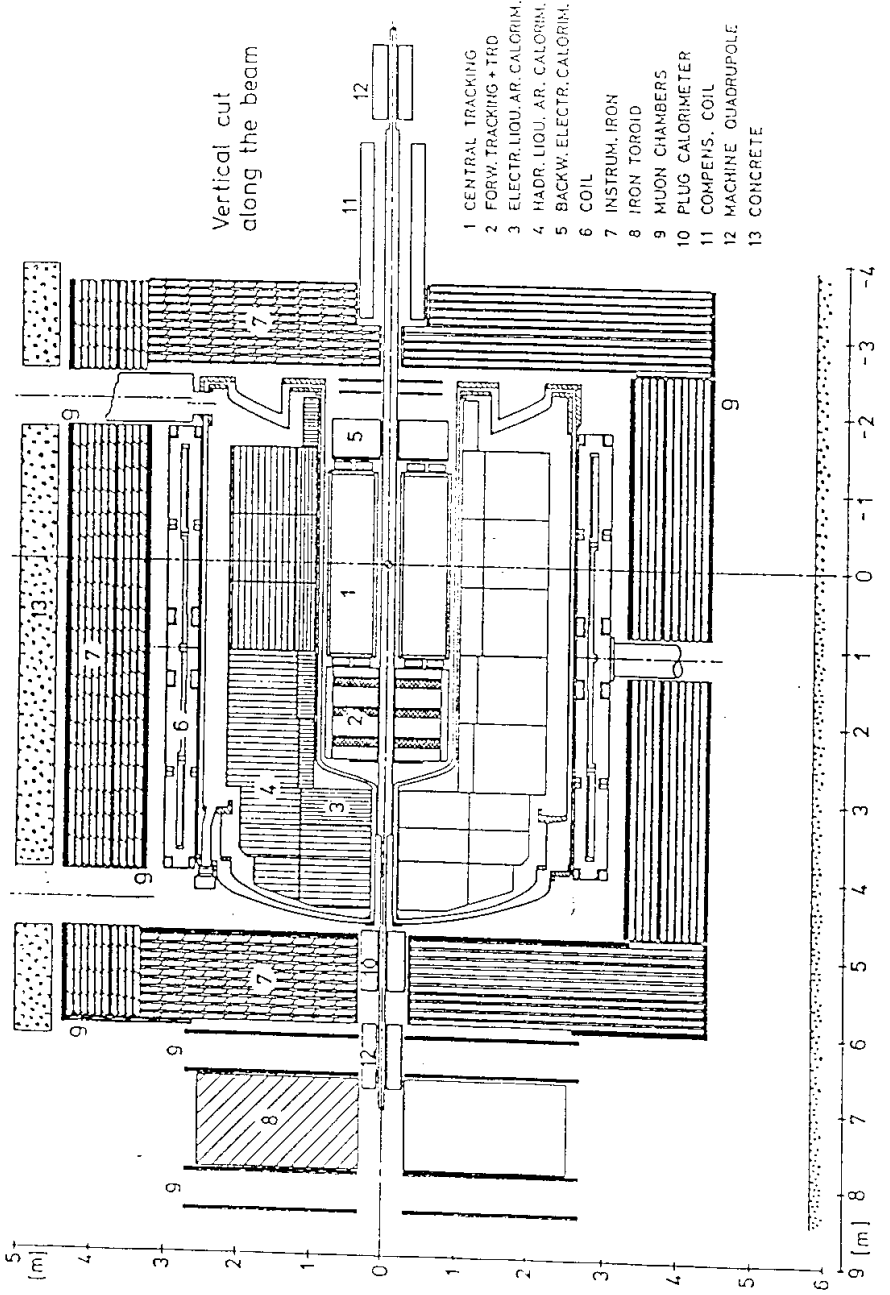


Fig. 1.1: H1 detector layout [3]

1.2 Trigger Considerations

Both electrons and protons are not spread equally over the entire ring, but kept in bunches, so that every 96 ns collisions may take place. Of course, collisions are much less frequent. The task of any trigger is therefore to determine, whether something interesting happened during a bunch crossing or not. This is done in several stages at different trigger levels with increasing selectivity.

The first trigger level, which should reduce the data rate from the initial 10 MHz to anything below 10 kHz [3] has just about 2 μs decision time and should work 'dead-timeless'. Part of it is the z-vertex trigger, which is more extensively discussed in [4] and chapter 2.1. Once this first level accepts an event, more sophisticated examinations take place, most of which are software based. Each trigger level i+1 only decides after level i has accepted the event. Otherwise, it aborts. This procedure minimizes dead time of the detector - no further data can be taken, while trigger levels 2 and 3 analyze data. Trigger levels higher than 3 operate on multi event data buffers and do not cause further inefficiency. Finally, data are written to tape for offline processing.

The first level trigger electronics can not rely, due to its short decision time, on software based algorithms but must be hardwired. However, to allow flexibility, one should be able to control the major parameters influencing the trigger algorithm and therefore load them into the electronics. Furthermore, computers should be able to monitor trigger performance and find defective modules.

2. The z-Vertex Trigger

2.1 Overview

Fig. 2.1 shows a block scheme of the first level trigger logic with special emphasis given to the z-vertex trigger. Signals of at present five pairs of parallel planes of MWPCs (in future six, if a foreseen backward chamber is included) are preamplified and transferred out of the detector. Receiver cards[1] shape and synchronize these signals against the HERA clock, which has a period of 96 ns. Then, they are made available to the z-vertex trigger electronics as well as to a forward ray finder, which will not be treated here[2]. Furthermore, the signals are all kept in a pipeline, so that they can be made available to data acquisition computers on request.

The z-vertex trigger consists of two major parts, the ray finder, which will be discussed in detail in chapter 3, and the subsequent vertex finder. The goal is, to decide, whether particles seen by the MWPCs originate from the interaction zone of the accelerator - and are therefore candidates for interesting events - or from somewhere else. In the latter case, they should be considered as background, which could, for example, be caused by a collision of beam particles with some remaining gas atoms in the beam tube. The z-vertex trigger decides on the location of the vertex of a possible event along the beam axis.

Only the inner and outer barrel chambers and the first pair of forward chambers are analyzed in the z-vertex trigger, as shown in Fig. 2.2. A particle originating from somewhere along the beam axis always crosses four chambers:

---

[1] developed by Mr. H. Krehbiel, DESY Hamburg [6]

[2] The forward ray trigger is developed by the University of Orsay, Paris

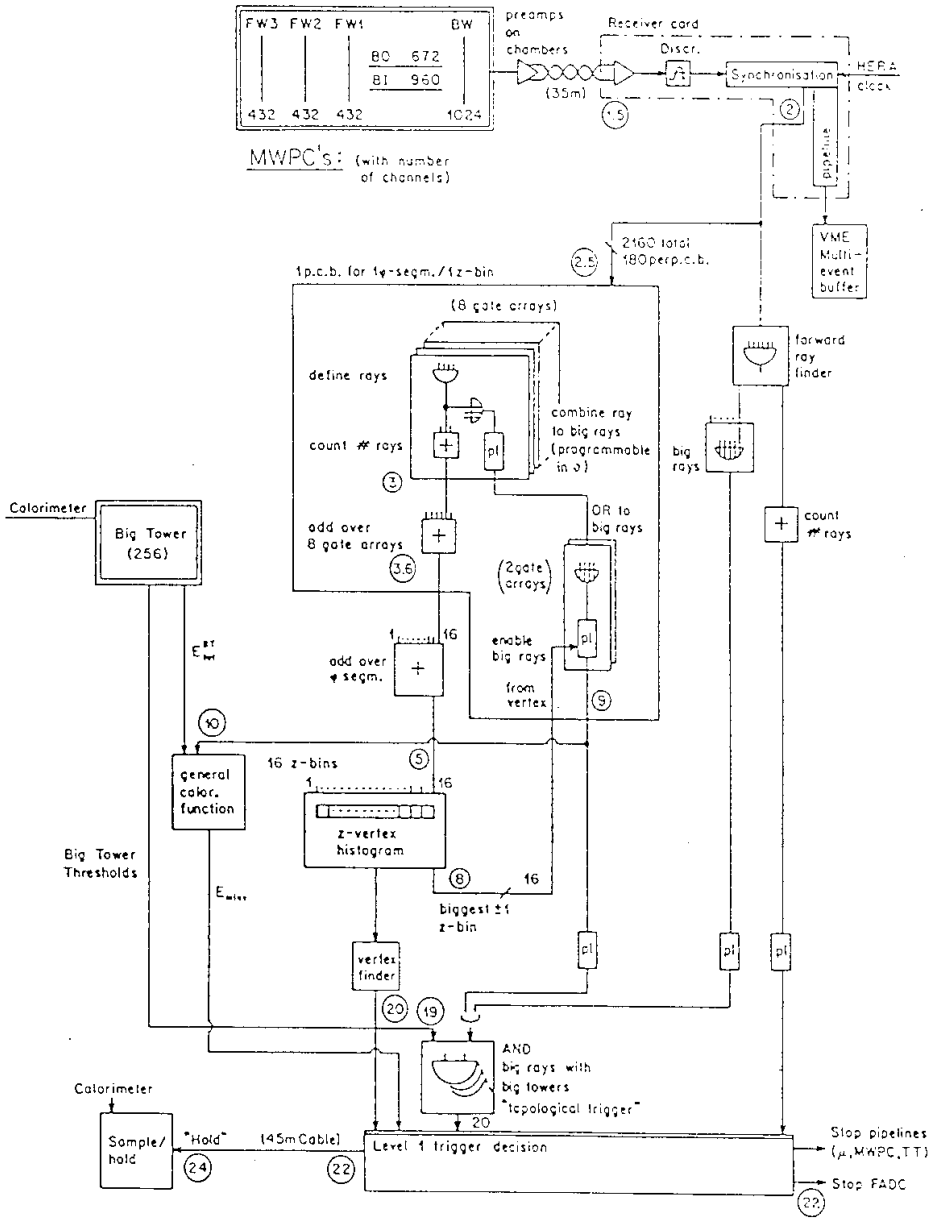


Fig. 2.1: H1 first level track trigger [4]

the two innerbarrel and either the two outer barrel or the forward chambers. The projection of its track to the r-z-plane is a straight line. The hits on the four chambers can therefore be connected and tracked back to the beam axis. Of course one does not know which hits belong to a given

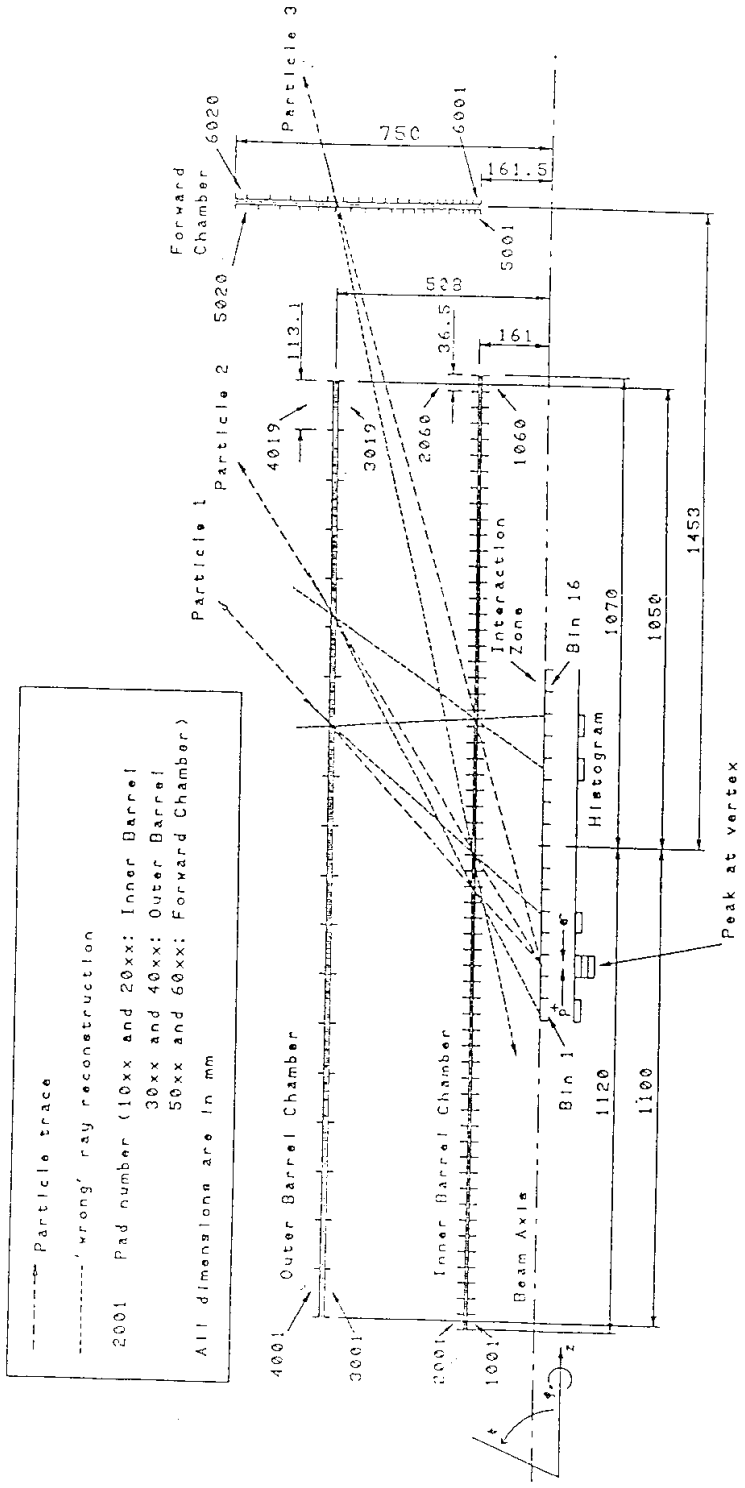


Fig. 2.2 MWPCs used in the z-vertex trigger

particle, so the task gets complicated. The ray finder will simply find all possible combinations of four hits, and construct a histogram of the corresponding z-coordinate for these ray candidates. In this histogram, the interaction zone along the beam axis is divided into 16 approximately 50 mm wide bins. The false rays - combinations of hits not belonging to the same particle - will produce a background in the histogram. The bin with the true vertex however will contain a peak.

It is the task of the vertex finder, to analyze, whether there is actually a peak in the histogram, indicating, that the vertex was somewhere within the interaction zone. This information will be made available to the first level trigger box, which will then make a decision based on this information as well as on data from the forward ray trigger and the calorimeter.

The z-vertex trigger can furthermore help the calorimeter trigger electronics because it indicates, into which parts of the calorimeter the rays are pointing: Once the vertex finder decided on one peak within the 16 bin wide histogram, many wrongly combined rays can be discarded. The remaining ones originate from the correct interaction area and are therefore likely to represent particles. They can be traced to the calorimeter, which is divided into 16 phi segments around the beam axis and 16 theta divisions, thus resulting in 256 so called big towers. This procedure allows to read out just a few instead of all big towers, which is hoped to substantially reduce background noise, thus allowing for lower energy thresholds. Similar combinations with a possible drift chamber trigger as suggested by H.-J. Behrend from DESY [8] are currently under consideration.

The encircled numbers in Fig. 2.1 give approximate signal propagation delays since the HERA beam crossing. The first level trigger box should decide within 22 beam

crossings (2.1 μs = 22 x 96 ns) and produce a 'L1 keep', if the desired trigger conditions are met. However, the first trigger level differs insofar from all other levels, that it should work 'deadtimeless', which means, that a decision has to be made every 96 ns, although each one belongs to a event 2.1 μs ago. This calls for a fully synchronously working electronic, signals belonging to the same event are nowhere allowed to have propagation delays which differ by more than 96 ns one from another.

2.2 Simulations of the z-Vertex Trigger

2.2.1 Event Data

Data available to me came in four files. They contained three different types of e-p-interactions, which were based on Lund Monte Carlo simulations by J. Gayler and background data from R. Eichlers code [4]. The detector has been simulated, using GEANT routines, by R. Eichler. An overview of the four event classes is given in table 2.1. The abbreviations will later be used to refer to the respective groups. NC stands for 'Neutral Current', CC for 'Charged Current'. The number indicates the lowest momentum transfer occuring (Q^2). Due to the few events available to me, no effort has been made to analyze different areas of the Q^2-X_{BJ} (X-Bjorken[3]) diagram separately. Background events came from the area between z = -4000 mm to z = 4000 mm and were evenly distributed. Fig. 2.3 shows some examples of histograms for the four classes of events.

---

[3] Momentum of the quark (of the proton) interacting with the electron divided by the momentum of the entire proton.

Table 2.1: Simulated events data

NCG100

Total events	: 499	
Events with less than 4 hits	: 51	(10.2%)
with less than 8 hits	: 69	(13.6%)
Avg. number of hits per event	: 34.8	

Q^2 \ X BJ =	0.20	0.40	0.60	0.80	1.00
667.. 1000 GeV²	17	2			
333.. 667 GeV²	56	2	1		
0.. 333 GeV²	358	32	3	2	

NCG1000

Total events	: 407	
Events with less than 4 hits	: 5	(1.2%)
with less than 8 hits	: 8	(1.6%)
Avg. number of hits per event	: 44.6	

Q^2 \ X BJ =	0.20	0.40	0.60	0.80	1.00
6667..10000 GeV²	8	1			
3333.. 6667 GeV²	27	5	5		
0.. 3333 GeV²	286	59	10	2	

CC100

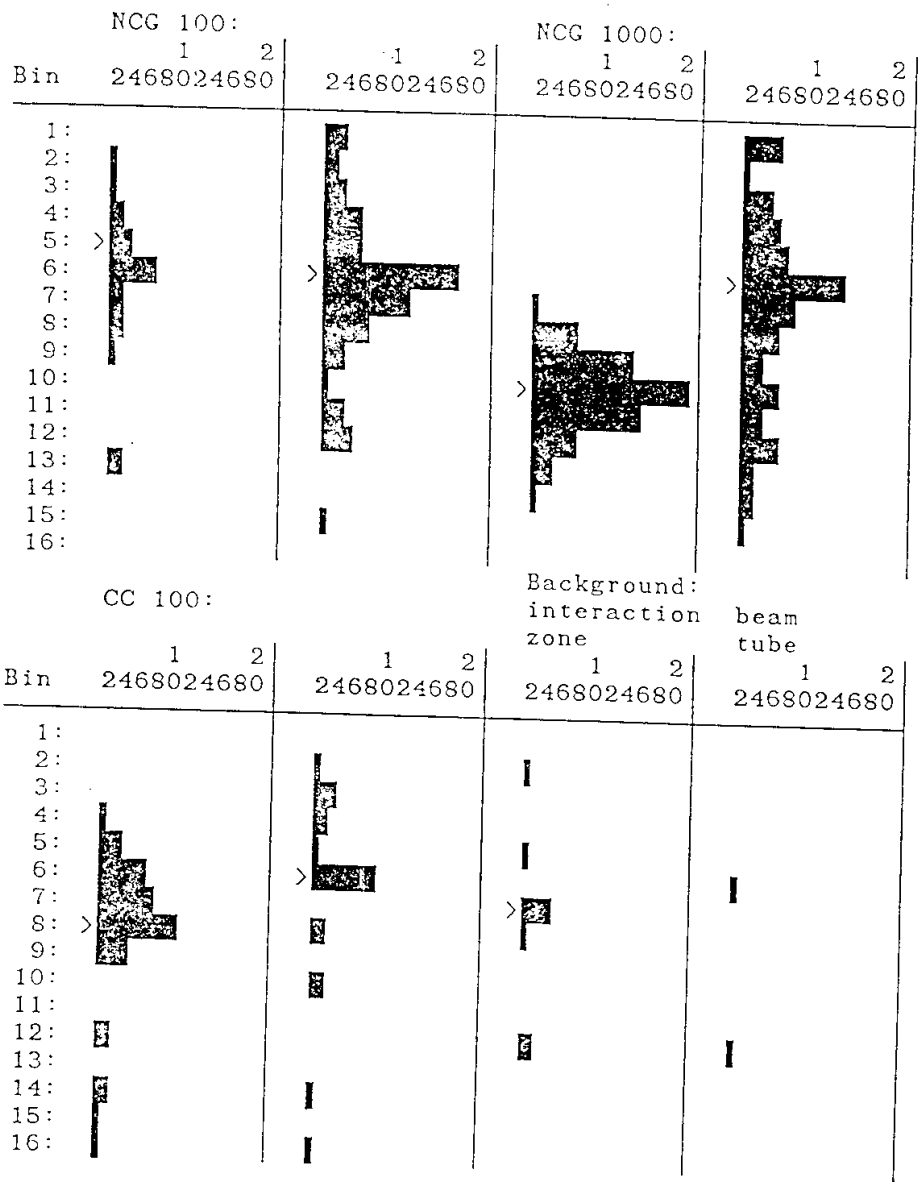
Total events	: 499	
Events with less than 4 hits	: 11	(2.2%)
with less than 8 hits	: 20	(4.0%)
Avg. number of hits per event	: 46.1	

Q^2 \ X BJ =	0.20	0.40	0.60	0.80	1.00
6667..10000 GeV²	19	9	5	1	
3333.. 6667 GeV²	56	19	8	2	
0.. 3333 GeV²	298	42	11		

Background

Total events	: 898	
Events with less than 4 hits	: 304	(33.9%)
with less than 8 hits	: 438	(48.8%)
(1) Events from interaction zone	: 106	(11.8%)
from beam tube	: 792	(88.2%)
Avg. number of hits per event	: 14.8	

(1) interaction zone: 16 histogram bins, 50 mm each, centered around z = 0.



'>' indicates the true vertex location.

Fig. 2.3: Examples of z-vertex histograms

2.2.2 Trigger Algorithms

The aim of the simulations discussed below was more to check the performance of the ray finder than that of the vertex finder. Nevertheless, at least one algorithm was necessary to decide, whether a simulated histogram contained a peak or not. The most obvious way to determine the statistical significance of a peak over the background is to require:

$$c_j - b >= h_1' * \sqrt{b}, \qquad (1)$$

where c_j is the content of the highest histogram bin,
b the sum over all other histogram bins and
h_1' a free parameter.

The algorithm has to be extended in at least two ways, to handle special cases: If there are two or more bins with a content of c_j, b should be the sum over the remaining bins. If b = 0, one should require

$$c_j >= h_2$$

with h_2 as a second parameter. More complex exception handling will have to be discussed before designing the vertex finder.

$$c_j >= h_1 * b, \qquad (2)$$

would be easier to implement than (1) since it does not involve a square root. In all the simulations discussed here, (1) and (2) showed nearly the same results as far as trigger efficiency is concerned. Differences were smaller than statistical errors for $h_1' = h_1$. All quoted results are therefore based on condition (2).

2.2.3 Definition of a Ray

Of course, none of the chambers have an infinite resolution. Fig. 2.4 gives a cross section of one chamber. Electrons set free from ionizing the molecules of the chamber gas by a high energy particle drift to the anode

wires. Near the wires, the electrical field is so high, that the accelerated electrons ionize further molecules, resulting in an avalanche of electrons. This signal induces also pulses on the cathode. The cathode cylinder of the central chambers consist of a capton foil, with a resistive graphite coating on the front surface and pads of a thin aluminium layer on the back side, which are connected to the readout electronics. For the forward chambers, a similar construction is used. The pad size is responsable for the chamber resolution in z (for the barrel chambers) and r (for the forward chambers).

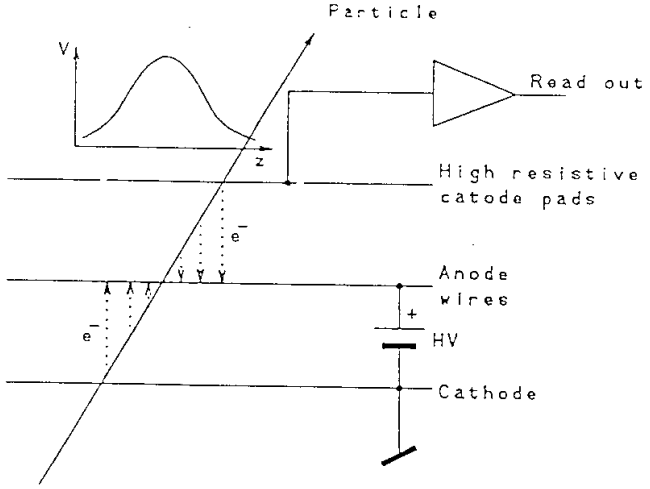


Fig. 2.4: Cross-section of a MWPC

Signals on the cathode cylinder (or plane) are centered around the crossing point of the particle track with the anode wire cylinder (or plane). Therefore, the relevant radius (z position for the forward chambers) is that of the anode wires. Table 2.2 gives an overview of the detector geometry as used in the following simulations. This geometry is not exactly identical to the currently available parameter sheets for the chambers - which are still not completely finalized. The influence of geometrical changes is discussed in chapter 2.3.2.

Table 2.2: Chamber geometry

	Old Geometry (1)	Current Geometry (2)
Inner Barrel Chamber		
Active length	z = -1120 .. 1070 mm	z = -1120 .. 1070 mm
Anode position	r = 157 , 166 mm	r = 157 , 166 mm
Number of pads	2 x 63	2 x 60
Pad size	34.8 mm	36.5 mm
Outer Barrel Chamber		
Active length	z = -1120 .. 1070 mm	z = -1100 .. 1050 mm or -1085 .. 1035 mm
Anode position	R = 501.1, 514.1 mm	R = 501.6, 514.6 mm
Number of pads	2 x 20	2 x 18 or 2 x 19
Pad size	109.5 mm	111.6 .. 119.4 mm
Forward Chamber		
Anode position	z = 1504, 1516 mm	z = 1447, 1459 mm
Number of pads	2 x 21	2 x 20
Pad boundaries	r = [mm]	r = [mm]
1st Chamber	165.0, 177.4, 190.8, 205.2, 220.7, 237.3, 255.2, 274.4, 295.1, 317.3, 341.2, 367.0, 394.6, 424.4, 456.3, 490.7, 527.7, 567.5, 610.3, 656.3, 705.7, 750.0	167.5, 180.5, 194.6, 209.7, 226.1, 243.7, 262.7, 283.1, 305.1, 328.9, 354.5, 382.1, 411.9, 443.9, 478.5, 515.8, 555.9, 599.2, 645.9, 696.1, 750.0
2nd Chamber	171.1, 184.0, 197.9, 212.8, 228.8, 246.1, 264.6, 284.6, 306.0, 329.1, 353.9, 380.5, 409.2, 440.1, 473.2, 508.9, 547.2, 588.5, 632.8, 680.5, 731.8, 750.0	167.5, 187.4, 202.0, 217.8, 234.7, 253.0, 272.7, 293.9, 316.8, 341.5, 368.1, 396.7, 427.6, 460.9, 496.8, 535.5, 577.2, 622.1, 670.5, 722.7, 750.0
z-Vertex Histogram		
Bin size	50.0 mm	53.5 or 54.6 mm

(1) Geometry used for simulations
(2) Status April 1988, the active length and the number of pads of the outer barrel chamber are not yet fixed.

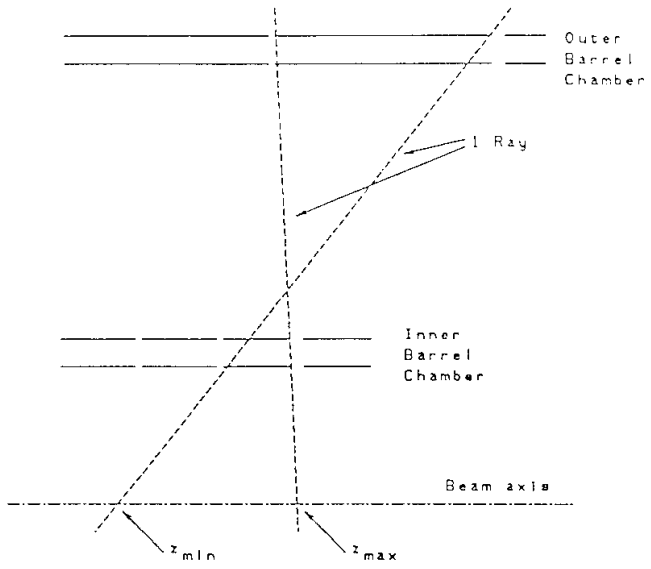


Fig. 2.5: Ray definition

A ray is given by a straight line in the r-z-plane. It crosses four MWPCs and fires (at least) one pad in each of them. These four pads therefore define the ray. Due to the finite resolution, there exists more than one line, which results in the same ray. These different lines originate from different z coordinates on the beam axis within the range $z_{min} < z < z_{max}$, as shown in Fig. 2.5. The ray is counted to the bin, that includes $z_{Ray} = (z_{min} + z_{max})/2$.

Currently, only the single pad, which is vis-à-vis of the point, where the particle track intersects the anode wire surface is considered to have fired. No provisions have been made for those cases, where signals are induced on two adjacent pads by one particle. This effect is however expected to be unimportant, since it results in both a decrease of z_{min} and an increase of z_{max} for a ray and normally not in a change of the bin associated. In any case

a ray should not be defined by more than four pads, since this would substantially complicate the ray finder design.

2.2.4 Simulation Results

Table 2.3 gives an overview over simulation results as far as various ray finder setups are concerned. If not otherwise mentioned, parameters for the peak finding algorithm as discussed in chapter 2.2.2 were:

$h_1 = 2.0$ and $h_2 = 4$

Statistical errors of the trigger performance are estimated by the formula

$$\sigma^2(h) = n * p * (1-p)$$

where n = number of simulated events

h = number of events that triggered

p = probability to trigger $\approx h/n$

(The formula holds for binominal distributions.)

2.2.4.1 Phi Segmentation

One can suppress background in the z-vertex histogram by dividing the ray finder into several axial segments. A separate z-vertex histogram is produced for each of them, based on hits in one phi segment only. These histograms are finally added up and the result is delivered to the vertex finder.

Under the assumption of fully efficient chambers and no additional hits induced by synchrotron radiation, different phi segmentations for the z-vertex trigger have been tested. A global '3 out of 4' option allowed to accept rays even if just 3 of its 4 defining pads have fired. We actually realized a more subtle option later, as discussed in chapter 3.4.2, which has not been simulated.

Table 2.3: z-vertex trigger simulations

Phi Seg. (1)	Trigger Condition 3/4 (2)	Pre. (3)	NCG1000	NCG100	CC100	Background beam tube	inter-action zone	tot.
Simul. Events			407	499	499	792	106	898
σ (approx.)			1.6%	2.0%	1.5%	0.3%	4.6%	0.7%
100% Efficiency								
16	F	NA	91.2%	71.5%	89.4%	0.5%	34.0%	4.5%
8	F	NA	92.4%	75.2%	91.0%	1.0%	59.4%	7.9%
4	F	NA	92.4%	75.6%	90.6%	2.8%	67.9%	10.5%
1	F	NA	87.5%	72.3%	84.8%	10.0%	83.0%	18.6%
16N	F	NA	92.4%	76.6%	91.2%	1.5%	67.9%	9.4%
16/8	F	NA	92.4%	72.9%	89.8%	0.5%	39.6%	5.1%
100% Efficiency								
16	T	NA	93.1%	82.6%	92.0%	8.1%	89.6%	17.7%
8	T	NA	89.9%	82.8%	89.6%	13.5%	95.3%	23.2%
4	T	NA	86.5%	80.0%	86.2%	20.2%	96.2%	29.2%
1	T	NA	58.0%	56.9%	57.1%	28.3%	88.7%	28.3%
16N	T	NA	88.5%	82.6%	88.2%	17.7%	96.2%	26.9%
16/8	T	NA	88.9%	83.4%	89.6%	13.4%	95.3%	23.1%
High Efficiency (4)								
16/8	F	F	90.4%	68.5%	88.4%	0.5%	34.9%	4.6%
16/8	F	T	91.6%	72.9%	88.4%	0.9%	43.4%	5.9%
16/8	T	F	90.2%	82.8%	89.2%	14.3%	92.5%	23.5%
16/8	T	T	88.5%	83.6%	89.0%	45.8%	96.2%	51.8%
Low Efficiency (5)								
16/8	F	F	85.3%	60.5%	83.2%	0.6%	17.9%	2.7%
16/8	F	T	86.5%	68.1%	87.0%	0.6%	28.3%	3.9%
16/8	T	F	90.2%	82.0%	91.2%	12.9%	88.7%	21.8%
16/8	T	T	89.9%	85.4%	89.4%	41.0%	97.2%	47.7%
High Efficiency (4)								
8/8	F	F	91.2%	73.9%	88.6%	0.9%	55.7%	7.3%
8/8	F	T	92.1%	74.9%	90.8%	2.4%	57.5%	8.9%
8/8	T	F	90.7%	83.6%	89.6%	19.6%	96.2%	28.6%
8/8	T	T	87.7%	84.4%	87.8%	54.5%	98.1%	59.7%

(1) Phi Segmentation:
- 16N : 16 phi segments, ORed over neighbours in phi in the outer barrel and forward chamber
- 16/8 : 16 phi segments, inner barrel and forward chamber with 8 segments, staggered in phi
- 8/8 : 8 phi segments, inner barrel and forward chamber staggered in phi by $2\pi/16$

(2) '3 out of 4' option: F = disabled, T = enabled

(3) Presetting of defective pads: F = false (not fired), T = true (fired), NA = not applicable

(4) Chamber efficiency: 97%, defective pads: 2%, pads fired due to synchrotron radiation: 0.2%

(5) Chamber efficiency: 90%, defective pads: 2%, pads fired due to synchrotron radiation: 0.2%

If the '3 out of 4' option is disabled, trigger rates for physics events (NCG100, NCG1000, CC100) are roughly the same for all setups (16, 8, 4 and 1 fold). They slightly decrease for the simplest situation without segmentation due to the many wrong combinations the ray finder generates, which fill the histogram with background.

A full 16-fold setup is worse than a 8- or 4-fold solution, since low momentum particle tracks, which are, bent in the r-phi-plane by the magnetic field of the detector, often cross segment boundaries and are therefore rejected. This effect becomes especially obvious for low multiplicity events. It could be remedied by forming a logical OR over neighbouring phi segments of the outer barrel and forward chambers. Such a setup has been tested: 16 phi segments were assumed, but for the outer barrel and forward chambers, signals of the phi segment i were derived by

$$x_i := x_{i-1} \text{ OR } x_i \text{ OR } x_{i+1}$$

where x_k = fire status of pad x in phi segment k. Results were comparable to a 8-fold segmentation, but background rejection was worse.

The background acceptance is approximately proportional to the phi segmentation, so that a 16-fold solution is clearly preferable. However this is not possible in its true sense, since the inner barrel and the forward chambers are only 8-fold segmented with the segments of each of the chamber pairs staggered by $2\pi/16$ in phi. Such a situation is comparable to a full 16-fold setup.

If one enables the '3 out of 4' option, no phi segmentation proves to be clearly worse than any other. The trigger rate as well as the background rejection is best for a 16 segment geometry. Forming an OR of neighbouring phi segments shows again counter-productive results. Also, the actual chamber geometry, with staggering, results in a de facto 8-fold phi segmentation.

On a first glance, it is surprising, that the '3 out of 4' option increases the trigger rate, especially for low multiplicity events. If one analyzes these events, it becomes clear, that they have not triggered in the '3 out of 4 disabled' simulation because their histogram had just a couple of bins filled with a single entry. The 'wrong' rays produced by the '3 out of 4' option allow the histogram to contain a 'better' peak. Very similar results were achieved, if $h_2 = 1$ was used instead.

2.2.4.2 Real Chambers

To study the influence of chamber efficiency and synchrotron radiation, the following assumptions have been made: 2% of all pads were assumed to be defect and preset to either true or false. 0.2% of all pads fired due to synchrotron radiation, which agrees with estimations done in [3]. These hits were randomly scattered over the chambers. Finally, the chamber was assumed to have an efficiency of 97% in an optimistic case and 90% in a less optimistic one. All simulations were made with the actual phi segmentation of the chambers, that is, staggering the inner barrel chambers and the forward chambers.

One finds, that in the optimistic case, the trigger rate decreases by about 2% if '3 out of 4' was disabled and stays about the same if '3 out of 4' is enabled. However, if one presets defect pads to true, the background acceptance increases dramatically by a factor of 2 to 3. About as many pads have now been set to true than have effectively fired.

In the less optimistic case of only 90% chamber efficiency, the trigger rate decreases especially for low multiplicity events, if '3 out of 4' is disabled. Rates as low as 60% are possible.

2.2.4.3 Eight Phi Segments

If one considers an 8-fold phi segmentation, one wishes to have the opportunity, however, to upgrade it later to a 16 segment solution. Since the chambers can not be altered, this requires the inner and forward ones to be staggered in phi from the beginning. This has been simulated, assuming optimistic values for their efficiencies.

Compared to the 16 segment setup, the trigger rate is about 2% better and compareable to the one of ideal chambers, if '3 out of 4' is disabled. However, the acceptance for background events is about twice as 'good' as well! Therefore, the 16-fold solution is clearly preferable. It also allows better big tower resolutions in phi, thus giving improved support for combined triggers.

2.2.4.4 Parameters of the Peak Search Algorithm

With a trigger algorithm as described in chapter 2.2.2 (2), the performance of the z-vertex trigger depends on two parameters h_1 and h_2, which is shown in table 2.4. Listed are the results of the high efficiency situation in the 16/8-fold setup as shown in table 2.3, where the '3 out of 4' option was disabled and presetting was made to false.

The value h_2 has significant influence on the background rejection rate because background is generally of low multiplicity, whereas h_1 is not critical and should best be around 2.0.

Table 2.4: Influence of peak search algorithm parameters

Peak search algorithm: Parameters	NCG1000	NCG100	CC100	Background beam tube	Background inter-action zone	Background tot.
h_1 = 0.5	90.4%	67.7%	87.8%	0.5%	33.0%	4.3%
1.0	90.4%	67.7%	87.8%	0.5%	33.0%	4.3%
1.5	90.4%	67.7%	87.8%	0.5%	33.0%	4.3%
2.0	90.2%	67.7%	87.8%	0.5%	33.0%	4.3%
2.5	89.4%	67.3%	86.6%	0.5%	33.0%	4.3%
3.0	88.2%	66.7%	85.4%	0.5%	33.0%	4.3%
3.5	85.7%	65.9%	82.6%	0.5%	33.0%	4.3%
4.0	82.1%	64.9%	78.4%	0.5%	33.0%	4.3%
h_2 = 1	4.9%	15.6%	6.4%	4.7%	52.8%	10.4%
2	1.2%	2.4%	1.4%	0.4%	14.2%	2.0%
3	1.0%	1.2%	0.8%	0.1%	5.7%	0.8%
4	0.2%	0.8%	0.6%	0.0%	1.9%	0.2%
5	0.0%	0.2%	0.0%	0.0%	0.9%	0.1%
6	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

2.3 Chamber Geometry

2.3.1 Pad Sizes

Whereas the forward chamber geometry is determined by the ray trigger requirements, the barrel chambers are used only for the z-vertex trigger. Its pad sizes can therefore be fully matched to the requirements of this trigger. The following formulas link pad sizes with the width of the z-vertex histogram bin, using simple geometric assumptions [4]:

$$Z = z_0 \, (R - r)/r \qquad (1)$$

$$z = z_0 \, (R - r)/R \qquad (2)$$

where Z = pad size of the outer barrel chamber

z = pad size of the inner barrel chamber

z_0 = histogram bin width

R = Radius of the outer barrel chamber

r = Radius of the inner barrel chamber

These equations can, however, only give an approximation to the real pad sizes, since actually four chambers are involved in each ray: both the inner and outer barrel chamber consist of a pair of chambers, for which the radii of the anode wire positions are about 10 mm apart. It is best to take the average radius for each pair for R and r respectively.

Further constraints are of a more practical nature: The inner barrel chamber should not have more than 60 pads. With an active length of 2190 mm, this results in 36.5 mm wide pads and, using (2), in z_0 = 53.5 mm. With an active length of 2150 mm for the outer barrel chamber, it would best contain 19 pads, 113.2 mm wide each. Further simulations refer to this geometry if not otherwise mentioned. Practical reasons may however again limit the number of pads to 18 and probably the active length to 2120 mm. 18 pads in a 2150 mm long chamber would result in z_0 = 55.7 mm (1). Thus, one would select an average histogram bin width of 54.6 mm.

2.3.2 Simulations of Various Detector Geometries

Whereas all simulations mentioned in chapter 2.2.4 used a detector geometry as listed in table 2.2 as 'old geometry', additional calculations were made using current data. 100% detector efficiency was assumed and the '3 out of 4' option was off. Table 2.5 gives an overview over the results, which prove to be independent of the exact geometry concerning the given significance of the results. However, indications are, that 19 pads for the outer chamber might be preferable as far as background rejection is concerned. Further data will be needed.

A further indication for the preference of 19 pads is given by feeding the trigger with random data: If the two chambers do not match in their resolution, random patterns,

which should fill a histogram (almost) only with background, would produce artificial peaks and therefore cause the trigger to accept the 'event'. 500 such 'events' have been generated, which flood the trigger with fired pads in one phi segment only. The resulting histograms show a much broader distribution of bin contents than those of physical events, Fig. 2.6 gives examples. (These broad histograms were the only case found, where trigger algorithm (1) from chapter 2.2.2 proved to be clearly preferable over (2)!) As expected, the acceptance for the random patterns increase for an outer barrel chamber with only 18 pads, compared to one with 19.

Table 2.5 Influence of chamber geometry

Detector Geometry	NCG1000	NCG100	CC100	Background (tot.)	Random Pattern (1)
Simul. Events σ (approx.)	407 1.3%	499 2.0%	499 1.3%	898 1.0%	500 2.1%
'Old Geom.' (2)	92.4%	72.9%	89.8%	5.1%	
'Current Geom.' (3)					
Long. 18	91.2%	72.5%	89.8%	5.3%	68.4%
Long. 19	91.6%	72.7%	90.0%	4.7%	63.4%
Long. 36	91.6%	72.5%	90.6%	5.5%	62.4%

(1) trigger parameter $h_1 = 4$
(2) refer to table 2.2
(3) refer to table 2.2
active length of outer barrel chamber:
z = -1100 .. 1050 mm
pad sizes for the outer barrel chamber:

	number of pads:	histogram bin width
Long. 18:	2 x 18 à 119.4 mm	54.6 mm
Long. 19:	2 x 19 à 113.2 mm	53.5 mm
Long. 36:	2 x 36 à 59.7 mm	54.6 mm

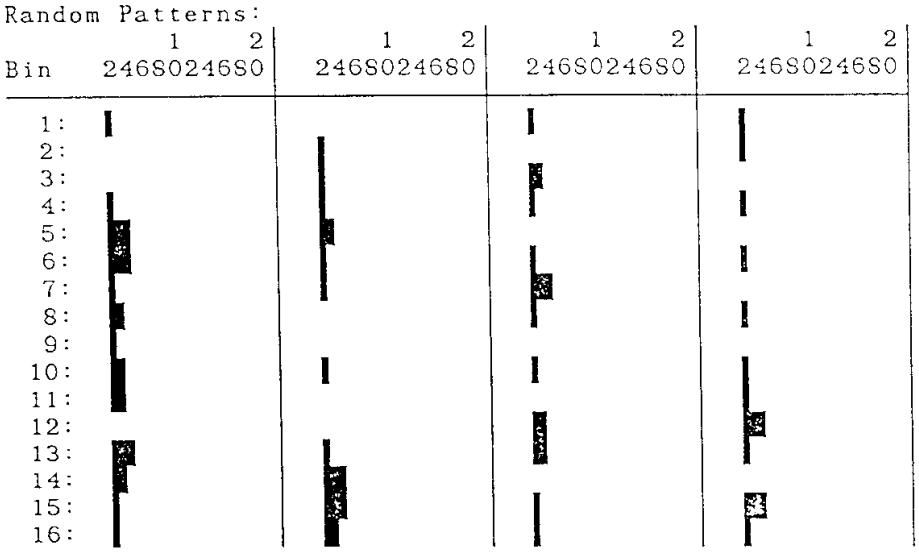


Fig. 2.6: z-vertex histograms of random patterns

2.3.3 Simulation of Various Ray Finder Algorithms

The new setup for the forward chamber has fewer pads (20 instead of 21). Consequently more rays than in the older version have a very bad resolution along the z-axis, i.e., $z_{max} - z_{min} >= 100$ mm (see Fig. 2.5). Thus, a ray found does not necessarily contribute to the correct histogram bin. If these rays are neglected in the reconstruction, the trigger efficiency decreases by about 5%, but the acceptance for background events from the interaction zone is almost halved. This is probably caused by the lower multiplicity of the background events compared to e-p collisions.

Table 2.6 gives an overview. Again simulations with more data will be needed. For these results, perfect chambers and a disabled '3 out of 4' option were assumed, but the situation is similar for chambers with just 97% efficiency. This effect does not influence the trigger design, since for each ray there will be a programmable 'Enable' bit

Table 2.6: Various ray finder algorithms

Ray Finder Algorithm	NCG1000	NCG100	CC100	Background beam tube	inter- action zone	tot.
Simul. Events	407	499	499	789	109	898
σ (approx.)	1.6%	2.0%	1.5%	0.3%	4.1%	0.6%
use all geometrically possible rays						
	91.6%	72.7%	90.0%	0.5%	34.9%	4.7%
use rays with z(max) - z(min) < 100 mm only						
	85.0%	64.7%	84.4%	0.5%	20.2%	2.9%

anyway, as described in chapter 3.4.1, so that a proper choice can be made in the software. However, disabled rays will be lost for the generation of big tower signals.

2.4 Big Tower Signals for Calorimeter Triggers

If one wants to support calorimeter triggers with z-vertex reconstruction data, it is essential to know, whether the z-vertex trigger found a peak in the histogram in the bin that included the true vertex or somewhere else. If the peak was found in a wrong bin, wrong rays would be traced to the calorimeter thus reading out wrong calorimeter big towers.

Table 2.7 shows, that at least the neighbouring bins to the one containing the peak should also be considered when determining which rays to trace to the big towers. If one divides the polar angle theta into 16 equally sized big towers and neglects azimuthal segmentation, signals such as shown in Fig. 2.7 are generated. Each bin of the ray finder indicates, that different calorimeter big towers should be analyzed, but one finds, that it does not matter too much

Table 2.7: Trigger bins

Trigger Bin (compared to real vertex) (1)	NCG1000	NCG100	CC100	Background beam tube	Background inter-action zone	Background tot.
Simul. Events	407	499	499	789	109	898
σ (approx.)	1.6%	2.0%	1.5%	0.3%	4.1%	0.6%
total rate	91.6%	72.7%	90.0%	0.5%	34.9%	4.7%
correct bin	69.5%	55.1%	67.9%	0.1%	23.8%	3.0%
corr. ± 1 bin	17.2%	15.0%	17.4%	-	11.0%	1.4%
elsewhere	4.9%	2.6%	4.7%	0.4%	-	0.3%
Peak = 1 bin						
correct bin	60.7%	48.1%	58.1%	0.1%	22.0%	2.8%
corr. ± 1 bin	17.2%	14.6%	16.8%	-	11.0%	1.4%
Peak = 2 bins						
correct bin	6.1%	5.2%	6.6%	-	-	-
corr. ± 1 bin	-	0.4%	0.4%	-	-	-
Peak > 2 bins						
correct bin	0.2%	-	0.8%	-	-	-
corr. ± 1 bin	-	-	-	-	-	-
> 1 Peak						
correct bin	2.5%	1.4%	2.4%	-	1.8%	0.2%
elsewhere	1.0%	1.4%	2.4%	-	-	-
Single bin						
correct bin	-	0.4%	-	-	-	-
corr. ± 1 bin	-	-	0.2%	-	-	-

(1) 'Peak = x bin(s)' : Peak in the histogram is x bin(s) wide.
'> 1 Peak' : Histogram contains more than one peak with the same hight.
'Single bin' : Histogram entries in just one bin
'correct bin' : Peak contains bin, which belongs to the true vertex
'corr. ± 1 bin' : True vertex is one bin to the left or right of the histogram peak
'elsewhere' : histogram peak further away from true vertex than one bin

which bin contains the true vertex. They all contain rays, true ones or just combinations of fired pads, which point into the same direction. This argument holds best for forward oriented events.

It is not yet clear, whether the outer barrel chamber can be realized with pads as big as 110 mm, or whether

their capacity would be too big, so that they have to be split in half and added electronically. If so, one might think, that it is preferable not only to OR the pads before feeding them to the trigger but use them separately. Simulations showed, that the overall trigger performance will not change (table 2.5), but about 3% more events will trigger in the correct bin. This improvement disappears again, if one looks at the neighbouring bins of the histogram peak as well. Since the ray finder logic would almost be doubled, this idea is far too expensive.

	NCG 100: Big Tower ——→	NCG 1000:	CC 100:
Bin	1 1234567890123456	1 1234567890123456	1 1234567890123456
1:	1---11----------	11----------	----------
2:	---1-11----------	-1----------	----------
3:	1---11----------	11----------	----------
4:	1---111---------	11----------	----------
5:	1--111111-------	11----------	----------
6:	!>1--1111111111---	!>11----------	----------
7:	-----111111-----	11----------	----------
8:	1-----111111----	-1----------	1----------
9:	---------1111---	-1----------	---1----------
10:	-------111------	-1----------	!>--111----------
11:	---------1111---	-1----------	---111----------
12:	----------111---	-1----------	----11----------
13:	----------	-1----------	----------
14:	----------	-1----------	----------
15:	----------11--	-1----------	----------
16:	----------	----------	----------

'>' indicates the true vertex location
'!' indicates the histogram peak location
'1' indicates, that there is at least one ray originating from bin y pointing to the big tower x.

Fig. 2.7: Generation of big tower signals

3. Realization of the Ray Finder

3.1 Problem Description

As shown in Fig. 2.1, the task of the ray finder is to deliver a histogram as fast as possible, so that the vertex finder can analyze it. The ray finder consists of 8 or 16 completely identical parts, one for each phi segment. Each phi segment produces its own histogram, separate adders will later sum them up.

The histogram consists of 16 bins along the z-axis, each about 50 mm wide. They are centered around z = 0. This allows the ray finder to be subdivided into 16 circuits, each required to deliver a number corresponding to the height of the associated histogram bin. Each such subcircuit fulfills the following tasks: about 166 MWPC pads[4] are to be combined to between 110 and 135 rays. Each ray is a combination of 4 pads, which lay on a straight line in the r-z-plane of the detector, which crosses the beam axis within the z-bin to be analyzed[5].

The ray finder has to produce an 8-bit number for each HERA clock cycle, indicating how many of these rays are logically true. A ray is considered to be true, if all 4 pads associated with it have fired, or if a '3 out of 4' condition, as described in detail in chapter 3.4.2, becomes true.

Furthermore, once the vertex finder has found a peak in the histogram, calorimeter triggers are eager to know into which of their big towers the rays are pointing. Therefore, rays should be grouped and stored in a pipeline for about 9

---

[4] The exact number of pads depends on the final design of the MWPCs.

[5] see chap. 2.2.3 for an exact definition of what a ray is.

HERA clock cycles[6], until the vertex finder has found a peak and they are needed by the calorimeter triggers.

Of course, by doing so, one saves all big tower signals, even of those z-bins which will not be needed later, since they do not correspond to a vertex candidate in the histogram and its associated rays have to be considered as only combinatorial with no physical meaning. However this was found to be cheaper than reconstructing rays again once the vertex finder has done its work, which would theoretically be possible, since all MWPC pads are pipelined in the receiver card.

3.2 Alternative Ray Finder Designs

Although the tasks of the ray finder appear to be relatively simple at first glance, the design becomes rather complex, once one realizes, how large the number of input signals is, which are all to be combined with most others.

Apart from the solution finally realized, two others were investigated. One approach was based on standardized digital ICs and PALs only, the other on an analog circuitry. Table 3.1 gives a comparison. Although alternatives A and B have never been discussed in full detail, it will become clear, that C is clearly preferable.

A ray becomes true, if all four associated MWPC pads have fired. 4-input NANDs are only available in packages containing just two of them. Alternatively, PALs of the type 16L8-D or 20L8-B [7] allow the integration of 3 respectively 4 circuits as shown in Fig. 3.1 in one package

---

[6] The exact timing will depend on the requirements of the calorimeter triggers and will be fixed later.

Table 3.1: Alternative ray finder designs

	A	B	C
Type	TTL / PALs	analog	Gate Array
Chip Count (1),(2)	33 PALs (2) 19 PROMs 21 Adder	33 PALs (2) 22 HCT Buff. 2 OpAmps 1 FADC	6 Gate Arr. 10 Mux 8 Adders
Total	73 Chips 1300 Pins	58 Chips 1016 Pins	24 Chips 672 Pins
Propagation Delay (max.)	89 ns (3)	65 ns (3)	108 ns (4)
max. Power Consumption	50.6 W	31.3 W	4.6 W (5)
Costs (6) Silicon Development Stock	269.-	303.- 20'000.- (7)	126.- (5) 33'500.- (8) 8'000.- (9)
Total (10)	68'900.-	97'600.-	73'800.- (5)

(1) PALs : 16L8-D, 20 DIP, 180 mA max., ca. 4.--
PROMs : 38S22, 16 DIP, 160 mA max., ca. 5.--
Adders : 74F283, 16 DIP, 55 mA max., ca. 2.--
HCT Buff. : 74HCT04, 14 DIP, 1 mA max., ca. -.50
OpAmps : not evaluated,
14 DIP, 250 mW max., ca. 20.--
FADC : not evaluated,
20 DIP, 1 W max., ca. 120.--
Gate Array: ASIC, 64 S-DIP, 60 mA max., 16.--
Mux : 74AC157, 16 DIP, 12 mA max., ca. 1.40
(2) 130 Rays assumed, 4 Rays per PAL
(3) +5 ns, if PAL 20L8-B instead of 20L8-B is used
(4) includes latching before gate array outputs, final delay time calculations after routing
(5) includes part of big tower generation
(6) only of the above mentioned ICs, no infrastructure or glue chips, Swiss Francs
(7) complex circuit and print design
(8) gate array development
(9) 500 gate arrays
(10) 16 Phi Segments

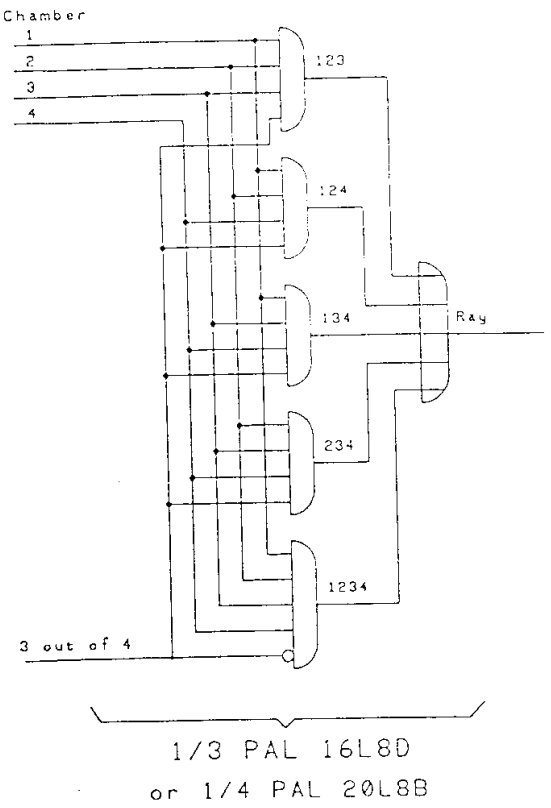


Fig. 3.1: '3 out of 4' with PALs

with only slightly more pins. As a side effect, a '3 out of 4' option can be included, which allows a ray to be true even if only 3 of its 4 defining pads have fired. This may help to compensate for a possible chamber inefficiency. The number of input pins per PAL restrict the maximum number of such circuits per package. However, since one pad is required in more than one ray, one could possibly further increase integration slightly. It would of course complicate stock keeping, since several differently programmed versions of the same basic type would have to be made available. A PAL delivers its answer after about 10 to 15 ns.

To find out, how many rays contribute to a z-bin, one might consider a RAM or ROM solution - and immediately discard it again in view of the maximum number of 130 rays.

Alternative A, as shown in Fig. 3.2, uses a cascade of FAST TTL adders. The first stage however are fast ROMs, which combine 7 rays to one 3-bit number in only 15 ns.

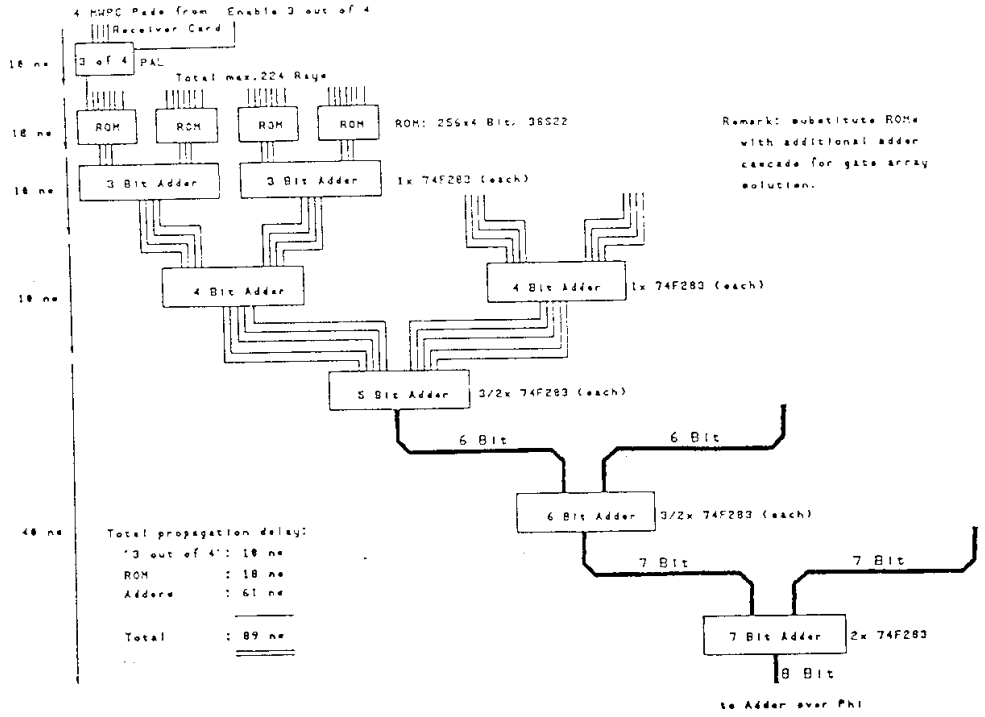


Fig. 3.2: Ray finder with standard TTL ICs

A different approach was suggested by Mr. Schoeps from SIN. It involved analog adders as shown in Fig. 3.3. Still, the rays would be build with PALs[7]. The TTL signals would be shaped by HCT buffers. Fast operational amplifiers add the signals and deliver an output voltage proportional to the number of true rays. This would be digitized again by a FADC, which might as well operate in a non-linear mode. The various drift, noise and other problems call for an experienced engineer for the circuit and print design rather than a diploma student.

---

[7] Analog solutions, involving another operational amplifiers and a discriminator would not improve neither the chip count nor costs.

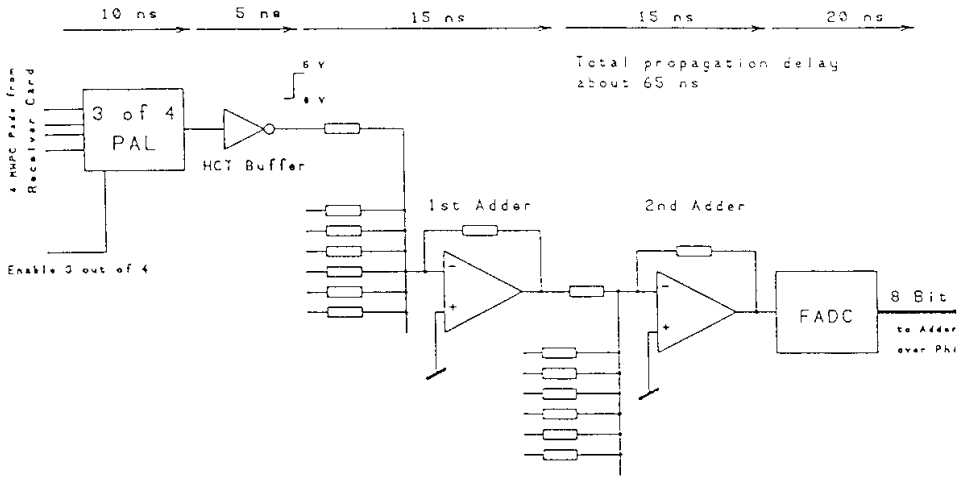


Fig. 3.3: Analog solution for the ray finder

3.3 The Gate Array Solution

None of the above alternatives were completely satisfying even though the problem of the generation of the big tower signals has not even been addressed. A higher degree of integration seemed desirable. Since readily available ICs could not solve the problem, gate arrays seemed favorable. Preproduced masters contain any number of gates from 300 to 20'000, each gate corresponding to a 2-input NAND or NOR. Two customized metallic layers implement almost any logic circuit desired. Gate arrays become economically interesting if one needs at least, depending on the master size, 500 to 2000 identical circuits. CAD tools allow the definition and simulation of the circuit. The results of these simulations are then guaranteed by the manufacturer, in our case NEC.

Currently, gate arrays are offered in three technologies: ECL - the Ferrari, amazingly fast and equally expensive, nothing for poor researchers. CMOS is the standard today, BiCMOS a sort of a compromise. The future will come

up with GaAs, for people with their stop watch calibrated in femto seconds.

CMOS gate arrays are well comparable with the fastest TTL families (AS, FAST). Their gate delay times may be slightly higher than their TTL equivalents, but due to the higher degree of integration, lots of I/O buffers can be saved - and time with them. Table 3.2 gives a comparison.

Table 3.2: CMOS gate arrays versus FAST TTL

	FAST TTL	CMOS Gate Array
Technology	Fairchild FAST	NEC CMOS 4/4A
Condition Temperature Vcc Load	Commercial 0° - 70° Celsius 4.75 - 5.25 V 50 pF	 -40° - +85° Celsius 4.5 - 5.5 V F/O = 3, 3mm wiring
2 Input NAND t (PLH) t (PHL)	74F00 2.4 ns ... 6.0 ns 1.5 ns ... 5.3 ns	F302 (1) 0.6 ns ... 2.5 ns 0.7 ns ... 3.1 ns
4 bit Adder C(0) to S(n) A(n) to S(m) (m >= n) C(0) to C(4) A(n) to C(4)	74F283 (2) 3.5 ns ... 10.5 ns 3.5 ns ... 10.5 ns 3.0 ns ... 8.5 ns 3.0 ns ... 8.5 ns	F523 (1),(3) 2.0 ns ... 11.3 ns 1.7 ns ... 16.5 ns 1.9 ns ... 9.7 ns 1.3 ns ... 11.3 ns

(1) internal gate delays only, no I/O Buffers concerned, to restrict temperature to 0° - +70° Celsius and Vcc to 4.75 - 5.25 V:
t (PD,min)' = t(PD,min) * 1.26
t (PD,max)' = t(PD,max) * 0.89
(2) Look ahead carry
(3) Ripple carry (?)
Sources: Fairchild FAST Data Book, 1985 [11]
NEC Block Library [12]

With ideas about as vague as shown in appendix A, we started discussing the problem with some of the about 40 gate array vendors in Switzerland. It proved to be a good idea to involve vendors already at this early stage of the design, since it helped to make optimal use of the specific advantages of gate arrays. Starting point was a circuit

similar to Fig. 3.2 - only the ROMs had to be substituted by additional adders.

Theoretically, the entire ray finder logic of one z-bin and phi segment could be integrated into one chip - 20'000 gates of the biggest available masters and 280 pins of the biggest pin grid array package could easily accommodate it[8]. Only, this would restrict our needs to 256 pieces (16 phi segments x 16 z-bins) and involve astonishing development costs. Fortunately, the adder cascade has the form of a binary tree, so that it can be broken up into 2^n identical parts, if only the root is realized with standard MSI ICs. As n increases, the gate arrays get less complex in terms of gate count and more identical ICs would be needed. On the other hand, the total pin count increases with n, since, first, the MSI root gets bigger and bigger, and, second, one pad has to be fed into more than one gate array, since it would be needed in rays built in different ICs.

Rough cost estimates led to an 8-fold breakup. This calls for a 2360 gate master, NEC μPD65024 was suitable. The final gate count of the circuit is 1968, which is equal to a 83.4% gate usage. This is a rather low usage rate, 95% would still be possible. However there is an other limiting factor, the 'pin pair count', the number of point-to-point connections. In our case, this number is 2012, which is very high and caused by the fact that most of the gate array logic is just combinatorial and includes only a few complex block library elements. We will need 2500 pieces (16 phi segments x 16 z-bins x 8 gate arrays = 2048, stock and other applications) at a cost of SFr. 16.-- each (in the cheap 64 pin Shrinked DIL package). Development costs were SFr. 33'500.--.

---

[8] Appendix B gives an overview over available CMOS gate array masters from NEC.

Critical were of course propagation delays within the gate array. NECs 1.5 μm technology was certainly one of the fastest at the time we evaluated the manufacturer (spring 1987). However, CMOS delays are heavily dependent on the driven load capacity, which is given by the required fan out and the wire length of the output signal on the chip. This dependency is not true for BiCMOS arrays, since their bipolar output transistors after each gate switch almost independent of their load. Since the critical paths in the adder cascade never drive more than one input, neither the necessary fan out nor the expected wire lengths are such, that the 30%-40% additional costs for BiCMOS would be justified. We estimated an overall gain of about 15 to 20 ns, compared to total delays of 108 ns (max.).

3.4 The Gate Array Circuitry

3.4.1 The Block Scheme

Fig. 3.4 shows the block scheme of the gate array, appendix C contains the detailed schemes. The block 'Input' builds the rays. One gate array accommodates 31 rays. This would require a package with at least 124 pins (31 x 4), but since some pads are used in up to four adjacent rays, less pins are needed. This allows the use of a 64 pin package, which leaves 45 pins for MWPC pad signals[9]. Table 3.3 shows the pin to ray allocation scheme applied. It results in some false combinations, i.e. rays that do not correspond to a straight line in the r-z-plane of the detector. Programmable latches in the chip allow each individual ray to be enabled or disabled. A total of up to 135 possible rays per z-bin require more than 4 ($=2^2$) gate arrays with 124 rays in total anyway. However, 8 ($=2^3$) gate

[9] other pins: 2 Gnd, 1 Vcc, 5 adder cascade outputs, 8 pipeline outputs, 2 clock inputs, 1 program enable input.

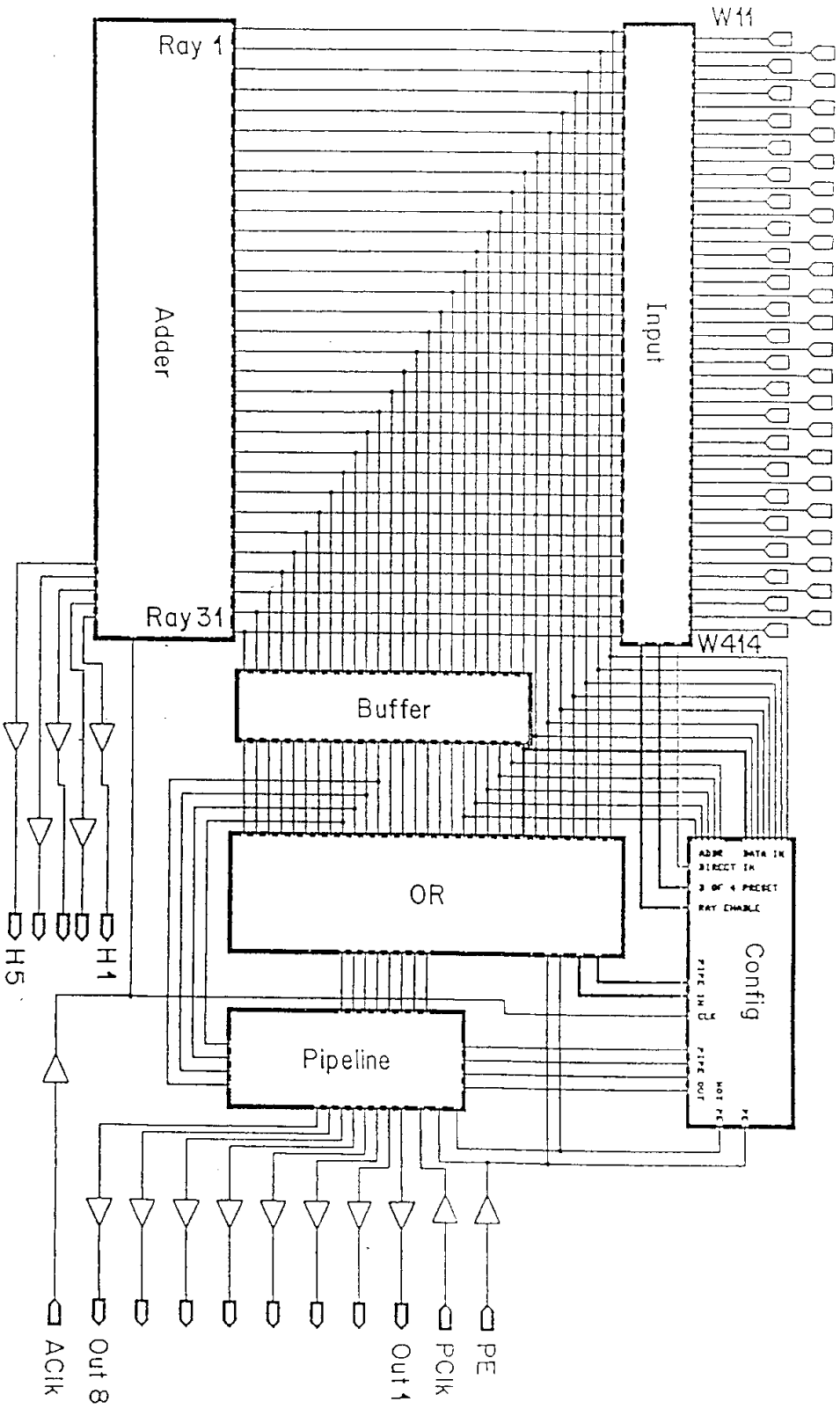


Fig. 3.4: Gate array block scheme

arrays accommodate up to 228 rays (8 x 31), enough to allow for these wrong and therefore disabled combinations.

The rays are fed into the block 'Adder', which contains the adder cascade and flip flops at the output for synchronization purposes, clocked by 'AClk'. The result is a 5 bit number, indicating how many rays were true.

Table 3.3: Ray and pin allocation for the gate array

Ray	Chamber 1	2	3	4	Direct Mode (1)	Pipe-lines (2)	Reassignable to (3) Function	Cond.
1	P11	P21	P31	P41	P31	1...5	Data 0	PE
2	"	P22	"	"	P41	"	" 1	"
3	"	"	P32	P42	P32	"	" 2	"
4	P12	"	"	"	P42	"	" 3	"
5	"	"	P33	P43	P33	"	" 4	"
6	"	P23	"	"	P43	"	" 5	"
7	"	"	P34	P44	P34	"	" 6	"
8	P13	"	"	"	P44	2...6	" 7	"
9	"	"	P35	P45	P35	"	Address* 0	"
10	"	P24	"	"	P45	"	" 1	"
11	"	"	P36	P46	P36	"	" 2	"
12	P14	"	"	"	P46	"	" 3	"
13	"	"	P37	P47	P37	"	" 4	"
14	"	P25	"	"	P47	"		
15	"	"	P38	P48	P38	"		
16	P15	"	"	"	P48	3...7		
17	"	"	P39	P49	P39	"		
18	"	P26	"	"	P49	"		
19	"	"	P310	P410	P310	"		
20	P16	"	"	"	P410	"	Pipe Enable	(4)
21	"	"	P311	P411	P311	"	Length 0	PE
22	"	P27	"	"	P411	"	" 1	"
23	"	"	P312	P412	P312	"	" 2	"
24	P17	"	"	"	P412	4...8	Pipeline 1	"
25	"	"	P313	"	P27	"	" 2	"
26	"	P28	"	"	P313	"	" 3	"
27	"	"	"	P413	P413	"	" 4	"
28	P18	"	"	"	P28	"	" 5	"
29	"	"	P314	"	P314	"	" 6	"
30	"	P29	"	"	P29	"	" 7	"
31	"	P29	"	P414	P414	"	" 8	"

(1) PE = High or 'Mode Control' bit set
(2) configurable through 155 bits of programmable OR
(3) '*' denotes 'active low' input
(4) PE = High or 'Pipe Enable' bit set

All the rays, some of them through a buffer, are fed into programmable ORs in the block 'OR'. Each ray is input to a maximum of 5 of the total of 8 ORs. Table 3.3 shows the association. The ORs help to define the big tower signals, which are then stored in a pipeline in the block 'Pipeline' during 1 to 8 clock cycles given by 'PClk'.

The block 'Config' contains all the necessary latches for programming the various functions of the gate array. They include: enabling of each ray, programming the ORs, the length of the pipeline, the '3 out of 4' option as described below, a 'Mode Control' bit and a 'Pipe Enable' bit.

3.4.2 The '3 out of 4' Option

MWPCs may, for various reasons, operate with non-optimal efficiency. It is therefore desirable, to have a software-selectable option, which accepts rays as true, even if only 3 out of the 4 defining MWPC pads have fired. Chambers may work inefficient, if they can not be operated with sufficiently high voltage. Since high voltage supplies are separate for each chamber, and probably even per phi segment, one would like to have the possibility not only to have a global '3 out of 4' option, as suggested in Fig. 3.1, but also to select, which chambers could be optional in a ray.

This calls for at least four bits, which define the current '3 out of 4' status of the ray finder - one bit per chamber. At least one IC contains rays formed from all four barrel chambers as well as rays formed from the two inner barrel and the forward chambers. This IC then would call for six bits - but which rays of its 31 belong to the barrel, which to the forward region? The clue is, to include a preset bit for each single MWPC pad. This gives a

total of 45 bits per gate array, as many as there are input pins.

45 bits might look as overkill. But imagine the following situation: All chambers operate normally, but a single preamplifier is broken. It is possible to preset the associated pad to logical true or false in the receiver card. Setting it to low will make all rays through it lost, setting it to high may lead, as the experiment progresses, to a situation, where more pads are preset to high than have effectively fired. Setting the pad to low and enabling the '3 out of 4' option circumvents both these problems.

3.4.3 The Mode Control Bit, Programming and General Purpose Applications for the Gate Array

When designing a gate array, it is worthwhile to consider future or more general applications. Our gate array basically solves two problems of more general interest: It is a 31 bit parallel adder, indicating, how many out of 31 input signals are logically true. Second, it contains an 8 bit wide pipeline with a length selectable between 1 and 8 clock cycles. More application specific is only the input section, where 4 input pins are combined into what is called a ray.

These thoughts gave rise to the idea of a 'Mode Control' bit, which can be set to direct mode when programming the chip. Once it is set, 31 of the 45 input pins are reassigned to directly represent a ray. (In this mode, at least the input pins P11 to P18 have to be set to low.)

Furthermore, it might be disturbing in other applications, that one has to provide external logic for programming the chip, even if this would not be required. Therefore, direct mode is also entered as long as PE (Program Enable) is high. Also, when PE is high, all rays are ena-

bled, 8 of them are directly dedicated to the inputs of the 8 pipelines, 3 others to control their length. One ray is reassigned to a 'Pipe Enable' function (active low), which allows to disable pipeline output on the next clock. These reassignments have no negative side-effects as long as the chip is operated as parallel adder, but allow the gate array to operate as a simple pipeline. That such operation may be economically interesting is shown in table 3.4.

Table 3.4: General purpose applications

	Discrete TTL (MSI)	CMOS Gate Array
Technology I/O Levels	HC or HCT TTL or CMOS	NEC CMOS 4/4A CMOS
Parallel Add.	few bits: ROM / RAM '283	1 Gate Array per 31 bits
Pipeline: 1 to 8 bit long, 8 bit wide		
Chips	8 x '164	1 Gate Array
Cost (1)	8 x 1.30 = 10.40	16.-- Fr
Pin count	112	64
f (max)	36 MHz	38 MHz (2)
selectable Length:	additional:	
Chips	8 x '151	no additional logic
Cost (1)	8 x 1.-- = 8.-- Fr	
Pin count	128	

(1) Swiss Francs, Prices per 100 for H-CMOS
(2) long setup time: 26 ns, but negative hold time, i.e., clock 2 ns 'after' new data possible due to long delays from input pins of gate array to actual pipeline.

PE enables also programming of the gate array: 5 rays define a address between 0 and 29 of an 8 bit latch, 8 others provide data to be loaded. Data are passed to the latch during 'AClk' is high. Appendix D shows the meaning of all the 236 programmable bits.

Fig. 3.5 and table 3.3 may help in understanding the various reassignments and modes of operation of the gate array.

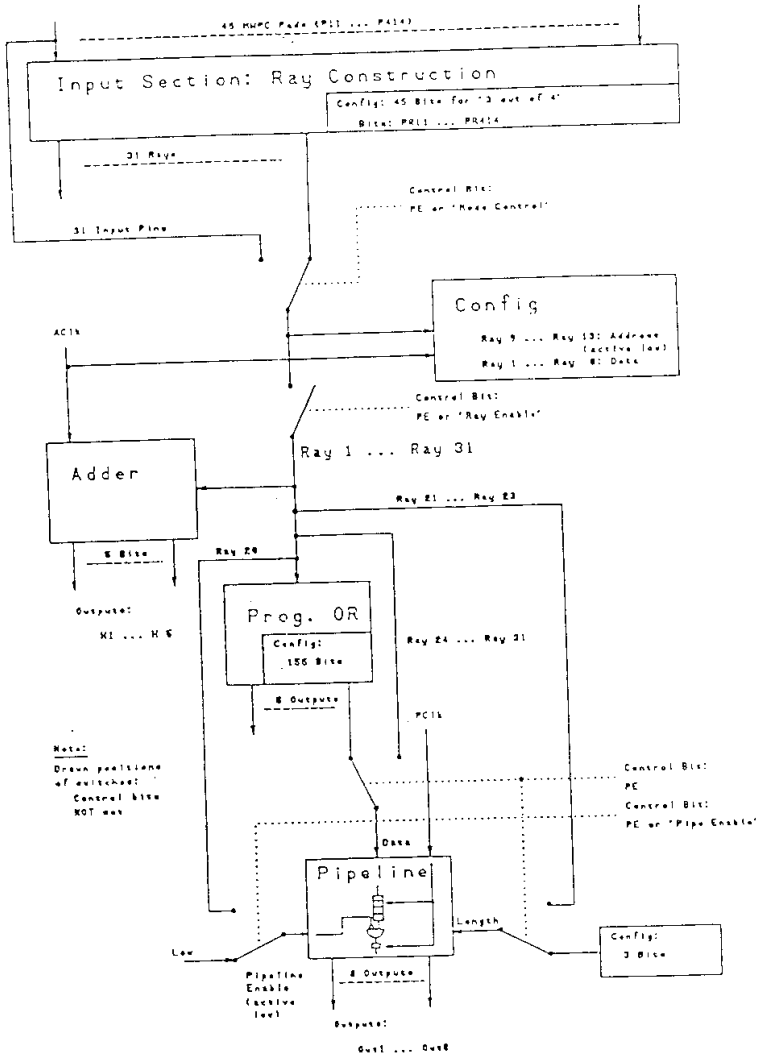


Fig. 3.5: Float diagram of the gate array

3.4.4 Use of the Gate Array for Generation of Big Tower Signals

As already explained in chapter 3.4.1, rays are grouped into big towers by means of the programmable OR and the pipeline. The calorimeter contains 16 big towers per phi segment. However, if one has 8 gate arrays per z-bin and phi segment, one has a maximum of 64 (8 x 8) pipeline

outputs available, some of them associated with the same big towers. We have to OR them again, and save them for another couple of clock cycles before reading them out, since the calorimeter triggers may need them later than just 8 HERA clock cycles after the rays were built.

This is exactly a task for which the gate array is built: We can operate two more chips in direct mode, with 30 of their inputs connected to the preprocessed big tower signals. It is then easily possible to program the internal OR such that 8 big tower signals result. Thus, two chips are needed for the 16 required big towers. It is no restriction that only 60 instead of the maximum of 64 pipelines of the first stage are used, as geometrical examinations show. One ray, ray 20, is reassigned to 'Pipe Enable', this time not by PE but by an internally programmable bit of the gate array. It will be connected to a bin select signal (see chapter 3.5.1).

3.5 The Ray Finder Printed Circuit Board

3.5.1 The Block Scheme

All elements of the gate array are now explained and the block scheme of the ray finder board as shown in Fig. 3.6 is the natural result of this discussion. One such card contains the logic for one z-bin and phi segment.

The MWPC pads, which have been digitized in the receiver card, are input to the ray finder card and first synchronized with the HERA clock in flip flops or latches. The pads are then input to a maximum of 8 gate arrays, each of them form up to 31 rays. All adder outputs of the gate arrays (H1 to H5) are then summed up again in a cascade of 74F283 adders. The resulting 8 bit number is sent to the following adders over all phi segments as a twisted pair

outputs available, some of them associated with the same big towers. We have to OR them again, and save them for another couple of clock cycles before reading them out, since the calorimeter triggers may need them later than just 8 HERA clock cycles after the rays were built.

This is exactly a task for which the gate array is built: We can operate two more chips in direct mode, with 30 of their inputs connected to the preprocessed big tower signals. It is then easily possible to program the internal OR such that 8 big tower signals result. Thus, two chips are needed for the 16 required big towers. It is no restriction that only 60 instead of the maximum of 64 pipelines of the first stage are used, as geometrical examinations show. One ray, ray 20, is reassigned to 'Pipe Enable', this time not by PE but by an internally programmable bit of the gate array. It will be connected to a bin select signal (see chapter 3.5.1).

3.5 The Ray Finder Printed Circuit Board

3.5.1 The Block Scheme

All elements of the gate array are now explained and the block scheme of the ray finder board as shown in Fig. 3.6 is the natural result of this discussion. One such card contains the logic for one z-bin and phi segment.

The MWPC pads, which have been digitized in the receiver card, are input to the ray finder card and first synchronized with the HERA clock in flip flops or latches. The pads are then input to a maximum of 8 gate arrays, each of them form up to 31 rays. All adder outputs of the gate arrays (H1 to H5) are then summed up again in a cascade of 74F283 adders. The resulting 8 bit number is sent to the following adders over all phi segments as a twisted pair

signal. Two additional gate arrays follow the pipeline outputs of the above mentioned chips to form the final big tower signals. The 'Bin Select' signal allows to select one z-bin according to the output of the vertex finder, so that one can disable all big tower signals for the other bins and then OR over all of them by means of a wired OR, which becomes possible due to the open collector output buffers.

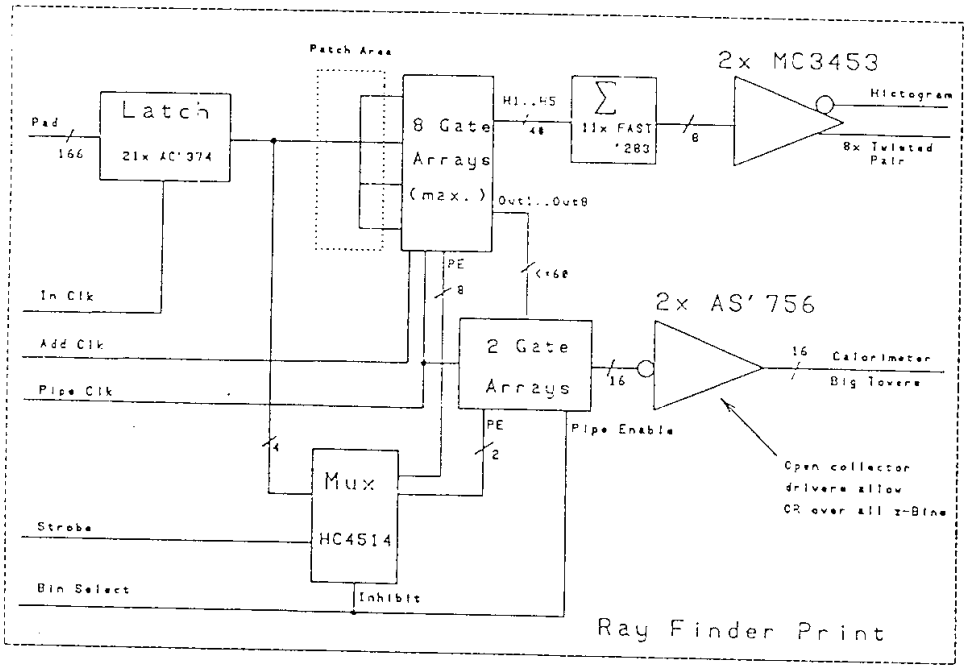


Fig. 3.6: Ray finder PCB block scheme

To minimize additional logic for programming the gate arrays, we use the test pattern capabilities of the receiver card to provide programming patterns. The receiver card can load up to two test patterns and send them rather than the actual MWPC signals to its outputs. This approach saves additional computer connections to all 256 ray finder PCBs. If one wants to read data from the ray finder back to the computer for monitoring purposes, this should be done from the following logic circuitry such as the adder cards, where all phi segments per bin are summed up.

A programming cycle starts by selecting a z-bin by the 'Bin Select' signal, which is connected to the 'Inhibit' input of a HC4514 4-to-16 line decoder. Four MWPC pads are dedicated to send the chip address (0 to 9, corresponding to the 10 gate arrays on one board) to the decoder, where they will be latched by means of the 'Strobe' signal. After this procedure, the PE input of the selected chip is set to high, all others to low. In a next step, programming patterns (latch addresses and data) are sent and read in with 'AClk'. These two steps are repeated until all chips are sufficiently programmed. Finally, the decoder has to be set to a illegal chip address, say 15, before normal operation can resume. To program the two chips operating as the big tower signal pipeline, one has to set the preceding gate arrays into direct mode, so that they are transparent. Of course, it is possible to program all phi segments in parallel.

It is currently a point of discussion, whether a finer granularity, especially in the forward direction of the detector, than big towers could be used in conjunction with a possible drift chamber trigger as proposed by H.-J. Behrend from DESY [8]. This would just require one additional gate array, used together with the two that build part of the big tower pipeline.

3.5.2 Realization of the Ray Finder PCBs

Let us deal with a small problem first: During programming, the gate arrays operate in direct mode and 13 (5 + 8) input pins are reassigned to be address and data pins. Of course, they should not be connected to one another externally - which may happen, since they may in some cases belong to the same MWPC pads when constructing rays during normal operation. Therefore, some additional 2-to-1 multiplexers have to be provided externally, which are also switched by the PE signal.

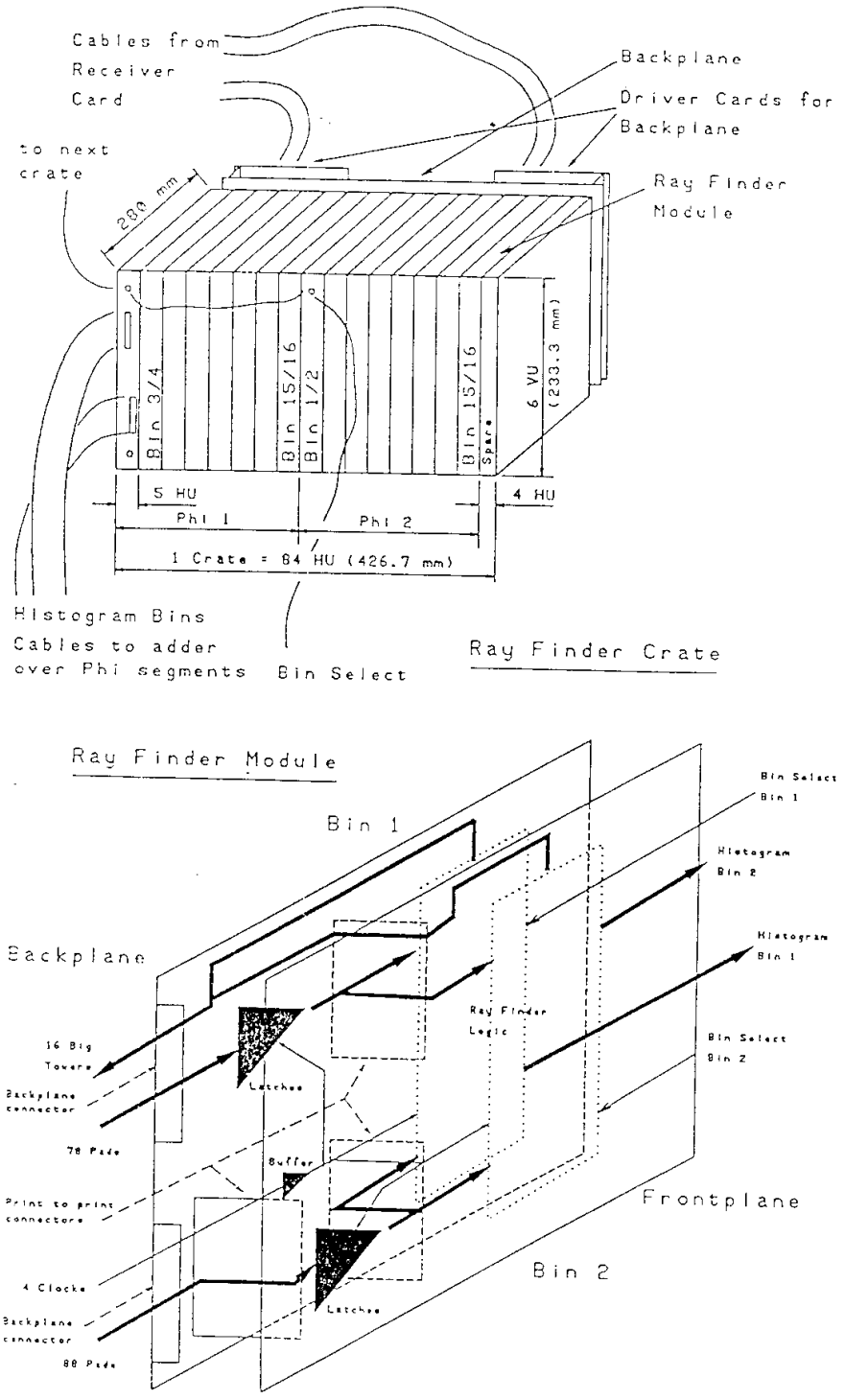


Fig. 3.7: Mechanical setup of the ray finder

The patch area mentioned in Fig. 3.6 is the only area where the various ray finder boards differ: It is unique for each z-bin, but too complex to be made by means of wire wrapping or similar techniques. About 166 input signals have to be distributed to a maximum of 360 gate array pins (8 x 45). This results in the unpleasant fact, that one needs different ray finder PCBs for each z-bin. A four layer board has been developed in such a way, that the patch area is entirely located in the two inner layers, whereas the outer layers as well as the drill map for through wholes are kept identical for all z-bins. This approach saves time in developing the various boards and makes their production cheaper as well.

Latching the MWPC signals at the inputs has a twofold purpose: It not only helps synchronizing the signals but works also as a buffer. Each pad may be used at up to four gate array pins per z-bin. With 16 z-bins and a maximum input capacitance of 10 pF, this would result in an untolerable heavy load of 640 pF (16 x 4 x 10 pF) for a backplane driver. A reasonable load seems to be around 80 pF, which implies one driver / latch for every second z-bin or ray finder PCB only. Therefore, a ray finder module consists of a sandwich of two ray finder PCBs for two z-bins, each containing latches for half the input signals.

Fig. 3.7 shows the mechanical dimensions of the ray finder which uses the 19" technique and Euro mechanics. One module of two boards has a thickness of 24 mm, so that it fits into five horizontal units (25.4 mm). Its height corresponds to six vertical units (233.3 mm) or a double height Eurocard. The card is 280 mm deep. Each crate accepts 16 modules or two phi segments, leaving four horizontal units (84 - 16 x 5) free for a possible additional support module of any kind.

It is natural, that the backplane contains all signals common to all z-bins of one phi segment. These are curren-

tly 166 MWPC pads, 4 clocks, the 16 big tower output signals and power (Vcc, Vee, Gnd). There is not much demand for Vee (-5 V), only the twisted pair drivers require it. This leaves three pins of the two 96-pin Euro connectors free for Gnd and two for Vcc. Table 3.5 shows the maximum power requirements. 2.3 A per board result in 4.6 A per module or 2.3 A per connector pin, which is slightly above the limit for pins in Euro connectors (2.0 A at 25°C). This problem, however, does not appear to be too serious, since none of the 16 z-bins require more than six gate arrays[10] (instead of the possible 8) to build all their rays, so that power consumption decreases to 2.0 A per board. Furthermore, the forward chamber has been redesigned recently and contains now one pad less, freeing an additional connector pin. The front plane contains the 'Bin Select' signal as well as the output of the adder cascade.

Table 3.5: Power requirements of the ray finder

ICs	Maximum configuration	Typical configuration
Gate Array	10 x 60 mA = 600 mA	8 x 60 mA = 480 mA
F'283 (Adder)	11 x 55 mA = 605 mA	9 x 55 mA = 495 mA
HC'157 (Mux)	16 x 12 mA = 192 mA	11 x 12 mA = 132 mA
ACT'374 (FF.)	11 x 58 mA = 638 mA	11 x 58 mA = 638 mA
AS'756 (Buffer)	2 x 80 mA = 160 mA	2 x 80 mA = 160 mA
MC3453 (Driver)	2 x 50 mA = 100 mA	2 x 50 mA = 100 mA
Total	2295 mA	2005 mA

A prototype ray finder PCB has been developed and soldered, but not yet fully tested. It was intended to represent bin 9 of the histogram (z = 0 mm to z = 50 mm), the MWPC pad to pin allocation is given in appendix E. It uses 147 of the 166 MWPC pads and combines them to 133 rays. The assumed chamber geometry is the 'old geometry' of table

[10] may increase to seven gate arrays, depending on the exact geometry of the outer barrel chambers (see chapter 2.3.2)

2.2. Therefore, the card will change again for the final release. All big tower signals were assumed to occupy an angle of 11.25° in theta (180°/16) since more accurate geometrical data are currently lacking. Certainly, this will not be the final situation. A future redesign should furthermore intend to substitute the 74ACT374 input flip flops by 74ACT573 latches or 74ACT574 flip flops, which are functionally identical, but simplify PCB design due to their different pin assignments.

3.5.3 Timing Considerations

Once all MWPC pads are available at the ray finder PCB, 'InClk' loads their data into the flip flops or latches. Flip flops are less sensitive to the clock pulse width but latches offer the opportunity for transparent operation. The board design is not influenced because flip flops and latches are available with identical pinning.

12 ns after loading (max., 74ACT573) most data are available at the input pins of the gate arrays, but some have to pass the additional multiplexers mentioned in chapter 3.5.2 and will arrive not more than 8 ns later. Another 25 ns (typ.) later[11], 'AClk' can move them into the output registers of the ICs. Again 11 ns (typ.) later, signals appear at the adder output pins of the gate array. If the gate array operates under restricted conditions ($5.00 <= Vcc <= 5.50$ V, $25°C <= T_A <= 45°C$), and one allows for 10 ns skew[12] between the various 'AClk' signals on all the ray finder PCBs, the gate arrays deliver valid results not later than 74 ns after the data arrival at their input

---

[11] refer to chapter 4.2 for propagation delays concerning the gate arrays.

[12] skew: time difference between different signals, which should theoretically come simultaneously. Skew is often caused by different path lenghts for different signals, for example due to different wiring, etc.

Table 3.6: Formulas concerning the ray finder timing

Latest availability of data at the gate array adder cascade outputs:

$$t_{max}^{Gate\ Array} = (t_{Setup}^{max} + t_{AClk \rightarrow Output}^{max}) * F_1 + \Delta t_{AClk}$$
$$= 64\ ns$$

Latest availability of data at the ray finder print histogram outputs:

$$t_{max}^{Ray\ Finder} = t_{pd(max)}^{Latches} + t_{pd(max)}^{Mux} + t_{max}^{Gate\ Array} + t_{max}^{Adder\ Cascade} + t_{pd(max)}^{Drivers} = 139\ ns$$

Allowed input skew on ray finder print:

a.) if input latches operate transparent:

$$t_{skew\ (a.)}^{Ray\ Finder} = \Delta t_{HERA} - t_{MaxMin}^{Latches} - t_{Max}^{Mux} - t_{MaxMin}^{Gate\ Array} - \Delta t_{AClk} = 35\ ns$$

b.) if input latches are clocked

$$t_{skew\ (b.)}^{Ray\ Finder} = \Delta t_{HERA} - t_{setup}^{Latches} - t_{Puls\ width}^{Latches} - \Delta t_{InClk}$$
$$= 79\ ns$$

where

t_{Setup}^{max}	= 25 ns	(see chapter 4.2)
t_{Hold}^{Min}	= 10 ns	
$t_{AClk \rightarrow Output}^{max}$	= 11 ns	
F_1	= 1.48	
F_2	= 0.64	
$t_{pd(max)}^{Latches}$	= 12 ns	(Source: Fairchild FACT Data-book, 74ACT573 [9])
$t_{pd(min)}^{Latches}$	= 1 ns	
$t_{Setup}^{Latches}$	= 3 ns	
$t_{Puls\ Width}^{Latches}$	= 4 ns	
$t_{pd(min)}^{Mux}$	= 8 ns	(Source: Fairchild FACT Data book, 74AC157 [9])
$t_{pd(max)}^{Drivers}$	= 15 ns	(Source: Motorola Interface Data-book, MC3453 [10])
$t_{max}^{Adder\ Cascade}$	= 40 ns	(Source: Fairchild FAST Data-book, 74F283 [11])
$t_{MaxMin}^{Gate\ Array}$	$= t_{Setup}^{Max} * F_1 - t_{Hold}^{Min} * F_2$ = 31 ns	
$t_{MaxMin}^{Latches}$	$= t_{pd(max)}^{Latches} - t_{pd(min)}^{Latches}$ = 11 ns	
Δt_{HERA}	= 96 ns	(HERA Clock Cycle)
Δt_{AClk}	= 10 ns	(estimated skew of AClk)
Δt_{InClk}	= 10 ns	(estimated skew of InClk)

pins. With a HERA clock cycle of 96 ns, 36 ns of skew of the input data can be accommodated, as the formulas in table 3.6 show. If all preceding electronics, up to and including the ray finder backplane can guarantee less than this limit, the input latches could operate transparent. Otherwise, they have to be clocked and one can accommodate up to about 79 ns of input skew, allowing additional 10 ns of skew for the various 'InClk' signals.

The MSI adder cascade needs 40 ns (max.), the twisted pair drivers 15 ns. The ray finder card finishes its work 139 ns after it received input data. Assumed propagation delays for all ICs except the gate array hold for the full commercial range of operating conditions. No formulas are available to scale them down to a restricted environment, which would positively affect maximum delays. Furthermore, since all ICs will be bought together, one can expect them to be produced in the same batch such that skew problems due to process tolerances can be expected to decrease. The transit times mentioned above arise therefore from calculations based on the the most pessimistic assumptions.

3.6 Test Equipment

A major problem concerns the test of the ray finder card. In the final setup, test patterns can be sent by the receiver card and the resulting histogram can then be compared with the theoretical one. But this procedure can not be applied during development and debugging of the circuit. Therefore a test setup had to be developed.

A scheme is shown in Fig. 3.8. A computer (IBM PC) controls via an IEEE-488 interface a circular shifter as well as a logic analyzer (Tektronix DAS 9100). The circular shifter contains 192 (12 x 16) 8 bit long shift registers, which can be loaded by the computer. Then, they are all

synchronously shifted, which produces a 192 bit wide output pattern, switching at the final operating frequency of the ray finder, namely 10 MHz, and repeats every 8th clock cycle.

166 bits out of these 192 are input to the ray finder card - or 45 of them to a single IC socket, so that one individual gate array can be tested. The output of either the ray finder card (8 bits of the adder cascade and 8 pipeline outputs) or the single gate array (13 pins) is read by the logic analyzer and sent back to the computer. It can then be compared with a simulated output.

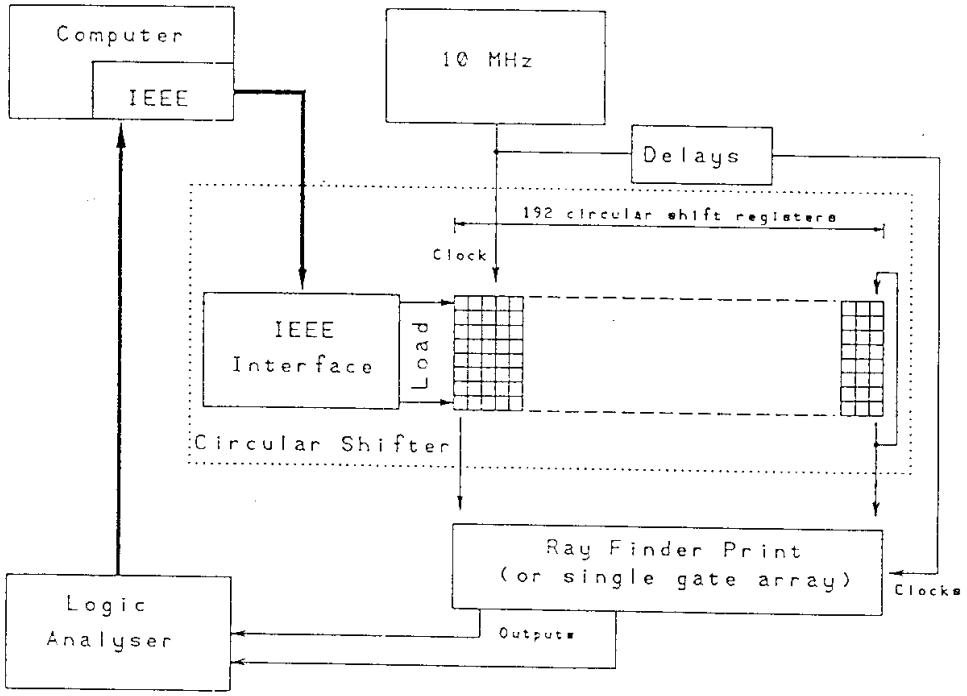


Fig. 3.8: Test circuitry for the ray finder PCB

The circular shifter can either be equipped with 74LS166 or 74HCT166 shift registers, in order to adapt to whatever logic levels (TTL or CMOS) will finally be selected for the ray finder backplane. It has been extensively tested with both types of ICs and used to measure relevant propagation

delays of the gate array. Currently, it operates with a reliability better than 99.97%, which is good enough to test one individual gate array or ray finder PCB. However the output signals look still too noisy to be trusted for extensive testing of a large series of ray finder cards. The problem is expected to be solved by a redesign of a backplane for the shifter, introducing more ground lines between signals.

4. Gate Array Development: Timing Considerations

4.1 Design Phase

Whereas the scheme of the gate array has been discussed in the previous chapter, another major point during the design of our chip was to achieve the best possible dynamic characteristics. The goal is to let the ray finder terminate its work as fast as possible. This means firstly to minimize maximum propagation delays, secondly synchronization should be done only as often as absolutely necessary. Each clock demands for some time reserve to compensate for possible skew in the clock distribution as well as drift. Consequently, one should try to match the longest and shortest data paths as close as possible.

One of the first decisions was to opt for 1.5 μm technology[13] in selecting the gate array manufacturer. This technique is about 25% faster than the older 2.0 μm processes. Furthermore, NEC documents the behaviour of their chip very well, i.e. they specify in detail, how much performance degrades with each of the three critical factors, process tolerance, temperature and supply voltage. One would wish, that all IC manufacturers followed this example, even for standard chips - but probably it would make life too easy!

Restricting the free air temperature to 35°C ± 10°C, and supply voltage (at the ICs) to 5.25 ± 0.25V results in degradation factors as follows:

$$t_{max} = t_{typ} * F_1$$

$$t_{min} = t_{typ} * F_2$$

with $F_1 = ((T_{max} - 25°C)/(85°C - 25°C) * 0.30 + 1)$
$* ((5.00V - V_{min})/(5.00V - 4.50V) * 0.15 + 1)$
$* 1.35 = 1.48$

---

[13] 1.5μm technology: size of the smallest structures on the silicon.

and $F_2 = (1 - (T_{min} - 25°C)/(-40°C - 25°C) * 0.23)$
$* (1 - (V_{max} - 5.00V)/(5.50 - 5.00V) * 0.13)$
$* 0.74 = 0.64$

(Formulas supplied by NEC.) To meet both the restrictions is facilitated by the small power consumption of the CMOS ICs (see Table 3.1 and 3.5).

Careful propagation delay calculations were made even before the circuit design was finalized. It has been influenced by assumptions on the wire length of critical signal paths on the chip. A standard value would be 3 mm, but it was considered to be sufficient to use 3 mm only for nets between larger blocks, whereas connections within the same block of the circuit were estimated to be not longer than 1 mm. The propagation delay of each block library element is given by the formula

$$t_{pd} = t_{pd0} + ((f-1) + 0.84 * l) * dt$$

where t_{pd0} : propagation delay for f = 1 and l = 0mm
f : fan out
l : wiring length in mm
dt : delay coefficient (usually about 0.18 ns)
(Source: NEC Block Library [12])

The results of these calculations are summarized in appendix F, which contains the project description as handed out to NEC, with maximum propagation delays which were, based on our calculations, guaranteed by the manufacturer even before placing and routing of the circuit.

The following steps were taken to increase performance as much as possible:

- It is possible to specify up to 16 critical data paths on the chip, which will be routed first and therefore be as fast as possible due to reduced wire lengths. The paths have been defined in such a way, that one can expect that they bind the block 'Adder' topologically together.

- The pinning was made such that pins which are used in the same rays are adjacent.
- Those rays, which enter the adder cascade at a carry bit position somewhere down the tree are slowed down by increasing their fan out as much as possible. That is why some rays do not pass the block 'Buffer' on their way to the block 'OR' (see Fig. 3.4).
- NEC Düsseldorf suggested to define the circuit in such a way on the DASY CAD workstation, that the netlist will extract in an optimal form, i.e. components with close relations to each other should be close together in the netlist.

However, it is difficult to judge how much each of these individual steps influenced the final result, since no comparison is available.

Final delay time simulations after placing and routing proved, that the specified requirements were met. To check this, special simulation patterns were provided and the outputs of the adder cascade just before being input to the output flip flops were observed. The chip was programmed to be in direct mode, and then all rays were simultaneously made true or false. The input signals of the flip flops mentioned were not influenced for at least 12 ns and stable again after 23.4 ns (typ.). The minimum propagation delay decreased in another pattern, stimulating just ray 2 in non-direct mode, to 10.2 ns.

The maximum propagation delay would further increase by 1.9 ns to 25.3 ns if the chip was not set to direct mode due to longer delays in the input section. This was not simulated but calculated on the basis of effective wire lengths. Calculations made by hand can never give an accurate result, but a limit: They are based on the longest delays of the respective block library elements. These delays may be different for various data paths and high-to-low versus low-to-high transitions. On the other hand, it is not possible in complex situations, as the block 'Adder'

is, to find a pattern which guarantees to stimulate the absolutely slowest path. There are just too many parameters involved: a single 1-bit adder F521 is defined by 12 different delay times. And the block 'Adder' contains 22 of them, each connected to nets with different total wire lengths.

However, careful selections for both the simulation patterns and the way, manual calculations were made, gave comparable results: The slowest path calculated was considered to be ray 16. It showed a propagation delay up to the input of the final flip flops of 28.5 ns, which is 12% above the longest simulated time of 25.3 ns but still clearly below the guaranteed maximum propagation delay as calculated in appendix F, which was 33.2 ns. Measurements, as discussed in chapter 4.2, show the simulated delays to be quite close to the reality: 22 ns for the above situation. A similar degree of reliability of manual calculations was found on other occasions.

Maximum delays of 25.3 ns and minima of 10.2 ns well meet the input skew requirements of appendix F: instead of the required 50 ns, one could accommodate up to 65 ns of skew at the input pins of the gate array (or 60 ns, if one believes in the maximum delay to be 28.5 ns).

Data paths towards the 'OR' and the pipeline do not have to be considered as critical, since maximum delays are of no concern: One could just make the pipeline one clock cycle longer or shorter. The 'OR' can also accept more input skew than the adder cascade, since there are no paths, which are obviously shorter than others, as there are in the cascade, where, for example, ray 1 is directly fed into the carry input of the last adder down the tree. Calculations made on the same basis than for 'Adder' in appendix F showed an acceptable input skew of about 63.5 ns compared to 53.4 ns. The difference of 10.1 ns is equivalent to about 48 mm wire length on the chip. We did not

intend to integrate inductivities by running circles around all the gates!

4.2 Testing the Engineering Samples

After the final delay time simulations had been accepted, we received 20 engineering samples of the gate array, which will later be used in the prototype PCB of the ray finder. Before doing so, we wanted to test all critical parameters of one chip extensively. In a first step, we just checked the pipeline in direct mode (PE = high), later we programmed the chip with the test equipment as shown in Fig. 3.8.

In the following discussion, the term 'Setup Time' refers to the minimal time between when data are made available to the input pins of the chip and the rising edge of the respective clock ('AClk' or 'PClk'). 'Hold Time' is the minimal time data has to be kept stable after clocking. It may become negative, which means, that a clock may even occur a short time after data disappeared at the input pins. The explanation is, that data need longer to proceed to the clocked flip flops than the clock signals do. If not otherwise specified, operating conditions were kept 'typical' (T_A = 25°C, Vcc = 5.00V).

4.2.1 Power Consumption

The easiest mode for testing the gate array is when PE is high, since no programming is required. The pipeline was checked this way. Since we did not want to burn the expensive device, supply current was checked first. Power consumption is given by the data sheet as 15μW per switching gate and MHz. The pipeline consists of roughly 500 gates, the adder cascade, which can not be turned off, even if no

'ACIk' is provided, of additional 190 gates. Let's now clock the pipeline with 10 MHz at 'PCIk', and change input data at the same rate. This would result in a supply current of (500+190)*10*15μW = 20.6 mA. We measured 23.4 mA, or 6.1 mA if just one gate array was fed with data. It is difficult to estimate the exact number of switching gates but the results are in accordance with the theory.

4.2.2 AC Characteristics of the Pipeline

The pipelines require a setup time of (13 ± 1) ns and a hold time of (-7 ± 1) ns when operated with a length of 1 clock cycle. Manually, one calculates, after placing and routing, a setup time of 17 ns, simulations were not available. The propagation delay between clock and data output was (14 ± 1) ns compared to 13 ns calculated. If the pipeline is operated at a length longer than 1 clock cycle, setup and hold times decrease to (7 ± 1) ns and (-2 ±1) ns respectively. This is due to the fact, that the F569 multiplexers, which determine the length of the pipelines (see appendix C) are no longer located before the first flip flops and are thus no longer influencing the timing.

A setup time of 13 ns (typ.) may increase to 26 ns (max.), which, without even takeing advantage of the negative hold time, implies a maximum operating frequency of 38 MHz. Considering the hold time, this may well increase to 83 MHz, or more if the length can be guaranteed to be longer than 1 clock cycle. However, I would suggest further tests with the actual frequency, if one intends to use the pipeline functions above 50 MHz. Dependencies of propagation delays from the supply voltage are given in table 4.1.

In a next step, access to the pipeline has been made through the programmable 'OR', which is of course expected to enlarge the setup time. In fact, it increased to 26 ns

for a length of 1 clock cycle and 19.5 ns for all other cases. The hold time was measured as -16.5 ns and -9 ns respectively. In any case, about 3 ns of setup time are gained in direct mode. All these times imply maximum operating frequencies of at least 19 MHz, enough for all ray finder and most other imaginable HERA applications.

4.2.3 AC Characteristics of the Adder Cascade

For all following measurements testing the adder section, we used test equipment as shown in Fig. 3.8, the shifter being equipped with HCT shift registers. They offer a transition time of both the rising and falling edge of 3.5 ns and a maximum relative skew of ± 2 ns. This should be kept in mind when comparing the following results with simulations. It is difficult to say, how much the skew influenced the measurements, since one can not easily judge, whether late signals stimulate fast or slow paths and vice versa. However, relative measurements, such as supply voltage and temperature dependency, which use the same patterns for a series of tests can be assumed to have an accuracy of about 0.5 ns relative to each other. Simulated times only include propagation delays from the input pins to the flip flop inputs, whereas measured times are actual setup times. They can be matched, since 'AClk' needs about as long from its input pin to the flip flop, as the flip flops own setup time is.

Ray 1 has a setup time in the adder cascade of just 13.0 ns, if it is the only ray to become true. Depending on the status of the preset registers of the '3 out of 4' option, and whether such a completion is necessary or not, it may vary by ± 0.5 ns. Simulations promised 13.6 ns (typ.), which is in good accordance indeed! Hold time was -8 ns. Ray 16 is, of course, much slower due to its far longer path and needs a setup time of 22 ns, the same as Ray 24.

A more complex pattern, let's call it 'All Rays', switches all rays on and off simultaneously. The Chip operates in direct mode, so that setup times may increase by about 2 ns if this is not the case. Measured setup time was 24 ns, the hold time -9 ns. Simulated were 23.4 ns and -12 ns respectively.

11 ns after 'AClk', rising edges of the output signals arrive at the pins, falling edges need 0.000'000'001 s longer. Theory says 9.65 ns and 10.6 ns respectively.

4.2.4 Temperature and Supply Voltage Dependencies

The 'All Rays' pattern was used to measure temperature and supply voltage dependency. Tables 4.1 and 4.2 show the results, which again are in accordance with the data sheet

Table 4.1: Supply voltage dependency

Vcc [V]	Measured [ns]	Theory [ns]
(1)		
4.50	+2.5	+4.0
4.75	+1.5	+2.0
5.00	27.0	(2)
5.25	-1.0	-1.8
5.50	-2.5	-3.5
(3)		
3.50	+7.0	(4)
4.50	+1.0	+2.5
4.75	+0.5	+1.3
5.00	17.0	(2)
5.25	-0.5	-1.1
5.50	-1.0	-2.2

Table 4.2: Temperature dependency

T(A) [°C]	Measured [ns]	Theory [ns]
(1)		
-30	-4.0	-5.4
-22	-3.5	-4.6
-9	-3.0	-3.3
-1	-2.5	-2.5
16	-1.0	-0.9
25	27.5	(2)
50	+2.0	+3.4
60	+2.5	+4.8
70	+3.0	+6.2
80	+3.5	+7.6

(1) Pattern: 'All Rays'
(2) Reference, 'typical' delay
(3) Pipeline with PE = high
(4) Formula invalid

and the formulas given in chapter 4.1. Whereas it is easy to control the supply voltage, temperature control is more difficult. Heating was done simply with a hair dryer, cooling with cooling spray. Temperature measurements have therefore typical uncertainties of ± 5°C.

4.2.5 Conclusion

Since all measurements agreed with expectations, the engineering samples can be accepted. I suggest to use simulated delay time values for further calculations as for example in chapter 3.5.3, since measured values are in good agreement with them, but have statistical and systematic errors. One of the latter comes from the fact, that one does not know, how the samples are made with respect to process tolerances. This may influence propagation delays by -26% to +35%. However, we have not found any reason or indication for not using the gate array in the ray finder.

Appendix A: First Ideas for a Gate Array

The following document has originally been handed out to gate array manufacturers as a description of our intentions. It should serve as an illustration how vage ideas still were at this time. Nevertheless, it was essential to contact manufacturers already then.

Beschreibung der Gate Array Anwendung im Ray Finder des z-Vertex Triggers im H1 Detektor von HERA (Hamburg).

(...)

Prinzipschaltung

Figur A.1 zeigt eine Prinzipschaltung des Rayfinders, wie er oben beschrieben wurde. In jedem Bin werden (beinahe) alle 180 Pads, die überhaupt in Rays vorkommen benötigt. Jeweils 4 werden zusammengefasst. Diese gestrichelt gezeichnete Region hat als Patcharea auf dem Print ausgeführt zu werden, sodass für alle Bins identische Prints verwendet werden können.

Jeweils 5 ANDs und ein OR bilden einen Ray mit der 3 of 4 Option. Diese 150 Rays werden über eine Kascade von Addierern zu einer 8 Bit Zahl reduziert. Gleichzeitig werden die Supertower Signale generiert und einer mit 10 MHz getakteten Pipeline zugeführt.

Realisierung

Vom Pincount her dürfte es natürlich schwer fallen, den mit der punktierten Linie umgebenen Bereich in einen einzigen Gate Array zu verpacken. Nutzt man jedoch die Tatsache, dass aus geometrischen Ueberlegungen klar wird, dass zwar alle Pads benötigt werden, pro Bin aber nur in relativ wenig Rays, und dass von 150 benötigten Rays bis zur nächsten mit Addierkaskaden behandelbaren Grenze von möglichen Rays ($256 = 2^8$) ein Spielraum für Overhead besteht, so scheint es möglich, die ganze Schaltung in 4 identischen Gate Arrays unterzubringen.

Genauste Ueberlegungen werden noch notwendig sein, um zu entscheiden, ob allenfalls ein Teil der Patch Area in den IC hineinverlegt werden muss, was es nötig machen würde, dem Array eingangsseitig verschiedene Modes programmieren zu können.

Eine Aufteilung in mehr als 4 identische Gate Arrays (von der Addierkaskade her wäre die nächste Stufe 8 Arrays) ist wenig wünschbar, da dann ein allzu grosser Teil der Kaskade mit MSI IC's realisiert werden müsste.

Da ein Freezing des genauen Timings der gesamten Triggerelektronik (nicht nur der hier beschriebenen) noch in weiter Ferne liegt, sollte die Länge der Supertower Pipeline (in Vielfachen des Clocks) zu Power Up bestimmbar sein.

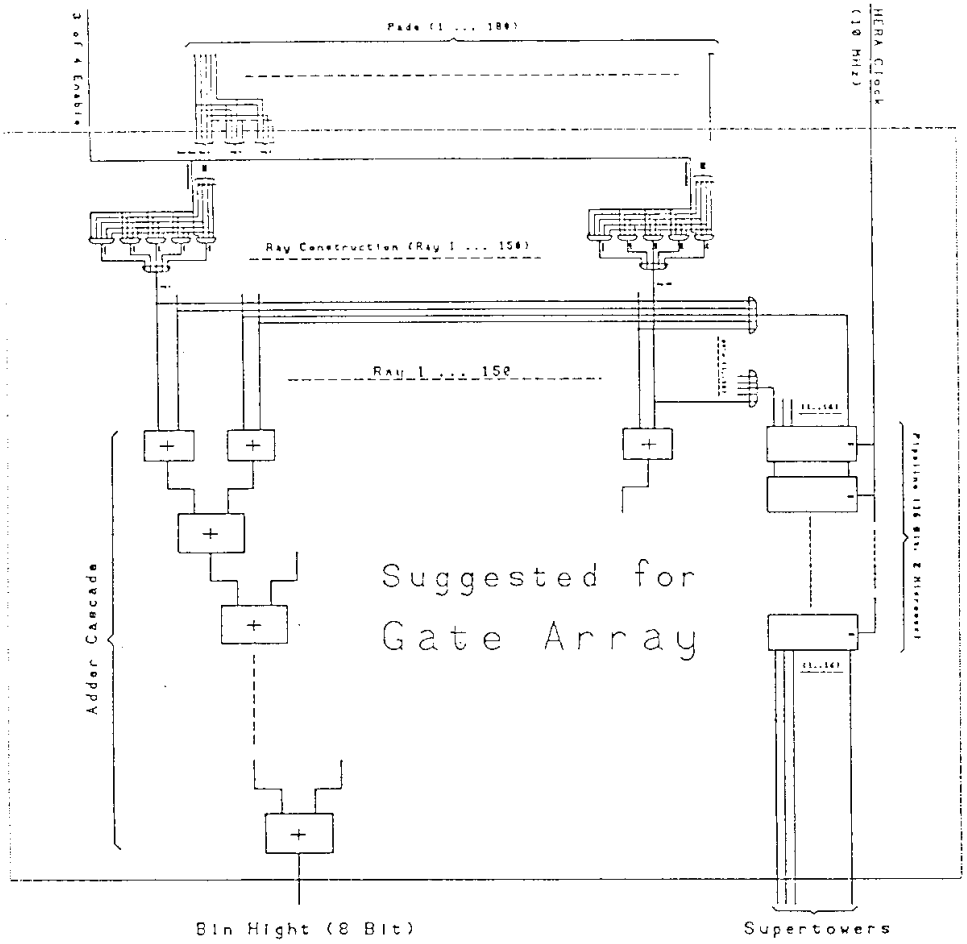
Neben dem hier beschriebenen Pflichtenheft wäre es im Hinblick auf allfällige zukünftige Anwendungen wünschbar, wenn der Gate Array auch in einem Mode betrieben werden könnte, der nur die Addierkaskade ausnützen würde, da das Problem 'wieviel von total x Leitungen sind an' ein wiederkehrendes in der Triggerelektronik von Detektoren ist.

Stückzahlen, Technologie

Gelingt es, die oben gestellten Forderungen mit jeweils 4 Arrays pro Bin zu erfüllen, so sind bei 16 Bins und 16 Phi-Segmenten total 1024 Arrays notwendig. Hinzu kommen Reserve (Maintenance), Prototyp und die Option, für allfällige zukünftige Anwendungen die Schaltung weiter verwenden zu können.

Die Technologie wird durch die Timingkriterien bestimmt, am Eingang stehen Twisted Pair Leitungen an, die mittels entsprechender Driver in alle gängigen Logikpegel umgesetzt werden können. Natürlich ist minimaler Stromkonsum als Nebenbedingung erwünscht, sollte aber nicht als limitierend betrachtet werden.

Fig. A.1: Prinzipschaltung



Appendix B: NEC CMOS Gate Array Product Spectrum
(Status: Spring 1987)

FAMILY NAME		CMOS-4A				
PRODUCT NO. μPD		65005	65006	65012	65013	65024
GATE COUNT		320	504	1088	1584	2242
I/O BUFFER	INPUT	54	62	82	100	108
	OUTPUT	54	62	82	100	108
	TOTAL	54	62	82	100	108
DELAY TIME	INTERNAL GATE	1.4 ns (F/O = 3, L = 3 mm)				
	OUTPUT BUFFER	4.0 ns (C_L = 15 pF)				
	INPUT BUFFER	2.0 ns				
TOGGLE FREQUENCY (TYP)		200 MHz				
OUTPUT BUFFER FREQUENCY		70 MHz (C_L = 15 pF) GUARANTEED				
OUTPUT BUFFER DRIVE CAPABILITY**		$I_{OLmin} = I_{OHmin}$ = 12 mA GUARANTEED				
POWER DISSIPATION (1 MHz)		15 μW/GATE 2.0 mW/OUTPUT BUFFER (C_L = 50 pF)				
AMBIENT TEMPERATURE		−40°TO 85°C (CMOS AND TTL LEVEL)				
POWER SUPPLY VOLTAGE		5V ±5% (TTL LEVEL) 5V ±10% (CMOS LEVEL)				
I/O INTERFACE		TTL CMOS COMPATIBLE				
PROCESS TECHNOLOGY		1.5 μm SI-GATE CMOS, 2 LAYER AL METALLIZATION				
NUMBER OF MACROS		230				
PACKAGE TYPE	DIL PLASTIC	16-48	16-64	20-64	24-64	28-64*
	FLAT PLASTIC	24-64	24-64	24-80	28-100	44-120*
	PGA PLASTIC	72	72	72	72-132	72-132*
	PLCC	18-52*	28-52*	28-68*	28-68*	44-84*

*NOTE: PACKAGE UNDER DEVELOPMENT **NOTE: MAX. 4 BUFFERS CAN BE CONNECTED IN PARALLEL WITHOUT LOSING EXTERNAL PINS

FAMILY NAME		CMOS-4								
PRODUCT NO. μPD		65022	65031	65042	65050	65070	65081	65101	65150	65200
GATE COUNT		2128	3575	4727	5742	7164	8510	10496	14943	19551
I/O BUFFER	INPUT	84	108	124	142	152	180	198	234	266
	OUTPUT	84	108	124	142	152	180	198	234	266
	TOTAL	84	108	124	142	152	180	198	234	266
DELAY TIME	INTERNAL GATE	1.4 ns (F/O = 3, L = mm)								
	OUTPUT BUFFER	4.5 ns (C_L = 15 pF)								
	INPUT BUFFER	2.0 ns								
TOGGLE FREQUENCY (TYP)		200 MHz								
OUTPUT BUFFER FREQUENCY		70 MHz (C_L = 15 pF) GUARANTEED								
OUTPUT BUFFER DRIVE CAPABILITY**		$I_{OL} = I_{OH}$ = 4 mA GUARANTEED								
POWER DISSIPATION (1 MHz)		15 μW/GATE 1.5 mW/OUTPUT BUFFER (C_L = 50 pF)								
AMBIENT TEMPERATURE		−40°TO 85°C (CMOS AND TTL LEVEL)								
POWER SUPPLY VOLTAGE		5V ±5% (TTL LEVEL) 5V ±10% (CMOS LEVEL)								
I/O INTERFACE		TTL CMOS COMPATIBLE								
PROCESS TECHNOLOGY		1.5 μm SI-GATE CMOS, 2 LAYER AL METALLIZATION								
NUMBER OF MACROS		230								
PACKAGE TYPE	DIL PLASTIC	28-64	40-64	40-64	40-64*	64	–	–	–	–
	FLAT PLASTIC	44-100	52-100	64-120	64-136*	64-136	52-136	120-160	120-160	120-160
	PGA PLASTIC	72	72-132	72-132	72-176*	72-176	72-208	72-208	72-208	72-280*
	PGA CERAMIC	72	72-132	72-132	72-176*	72-176	72-208	72-208*	72-280	72-280
	PLCC	68	68, 84	68, 84	68*, 84*	68, 84	68*, 84*	68*, 84*	–	–

*NOTE: PACKAGE UNDER DEVELOPMENT **NOTE: MAX. 3 BUFFERS CAN BE CONNECTED IN PARALLEL WITHOUT LOSING EXTERNAL PINS

Appendix C: Gate Array Schemes

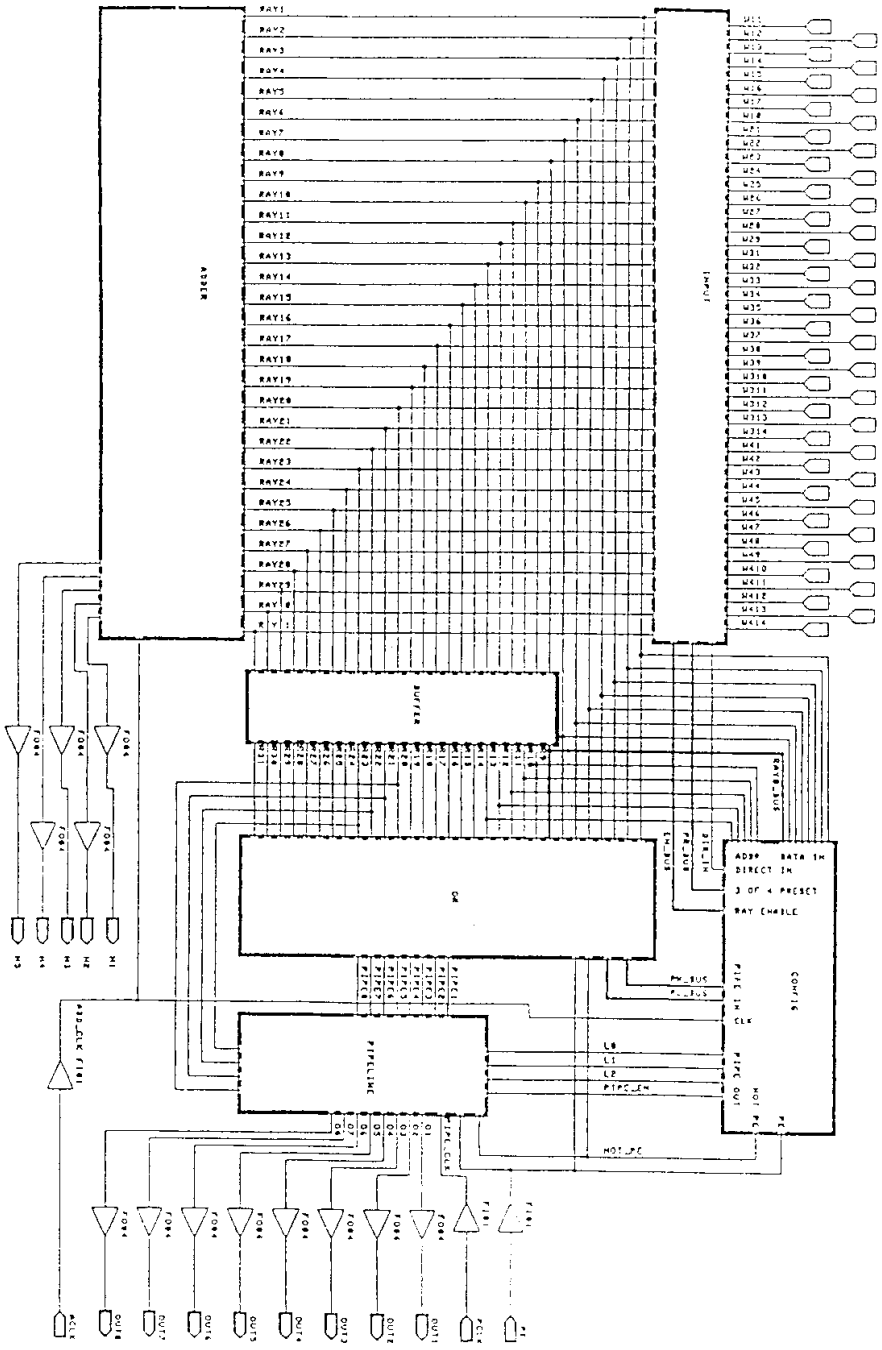


Fig. C.1 Block scheme

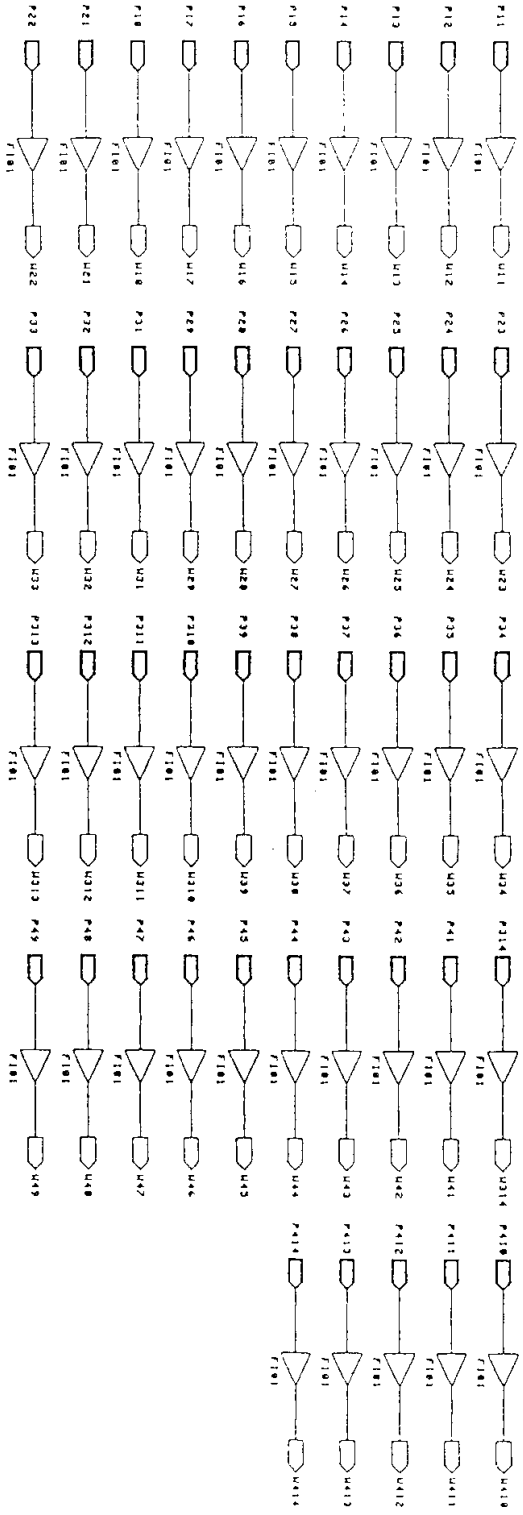


Fig. C.2: Input buffers

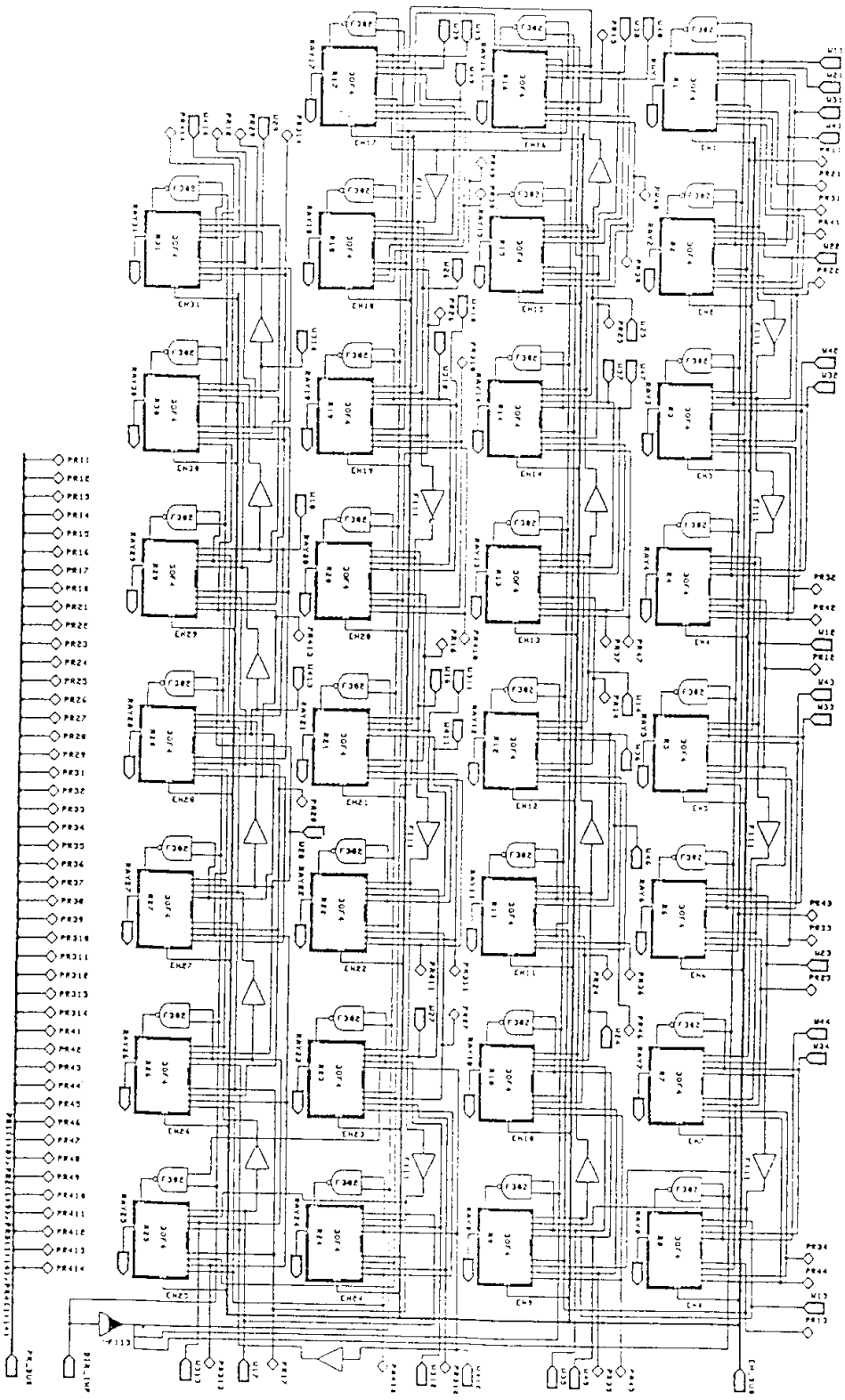


Fig. C.3: Block 'Input'

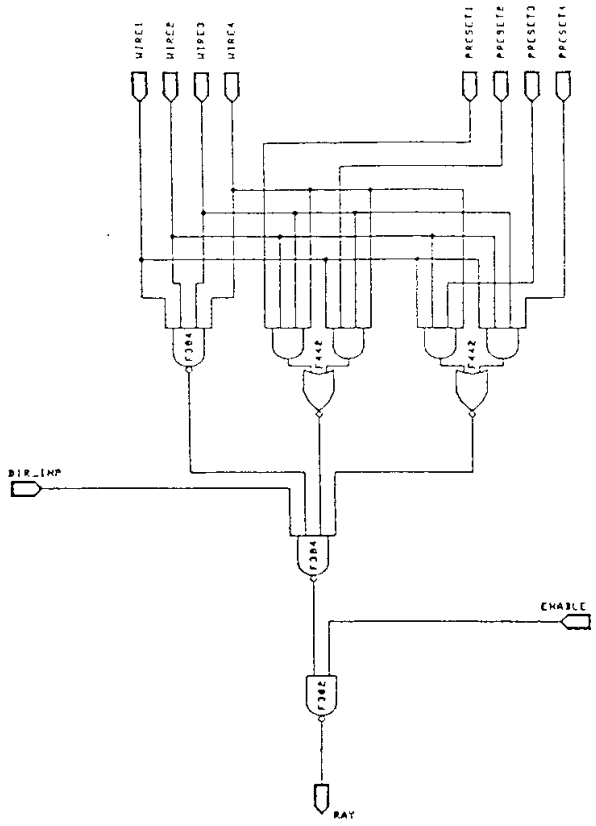


Fig. C.4: Block '3 of 4'

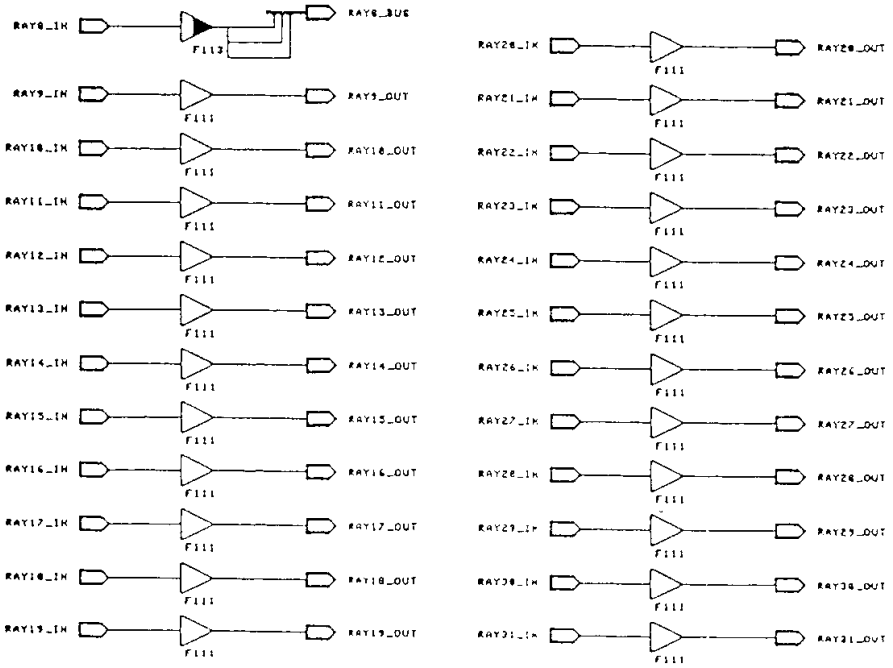


Fig. C.5: Block 'Buffer'

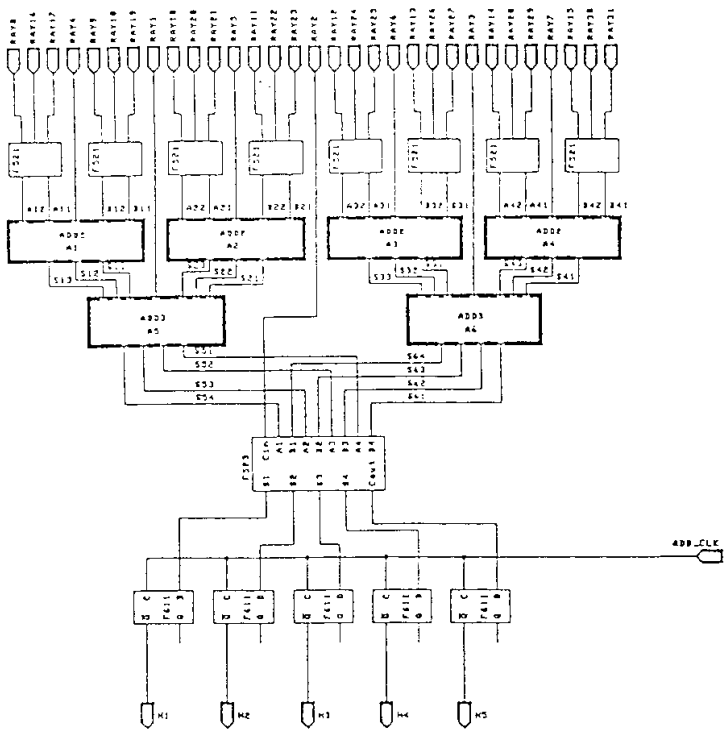


Fig. C.6: Block 'Adder'

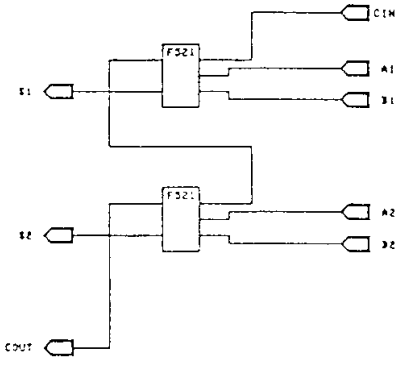


Fig. C.7: Block 'Add2'

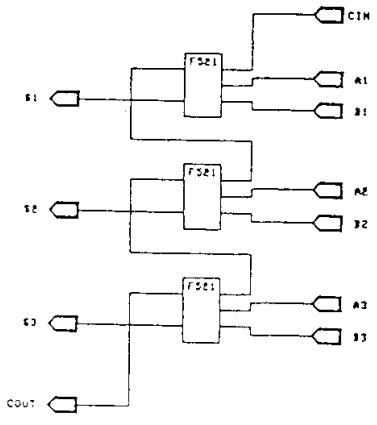


Fig. C.8: Block 'Add3'

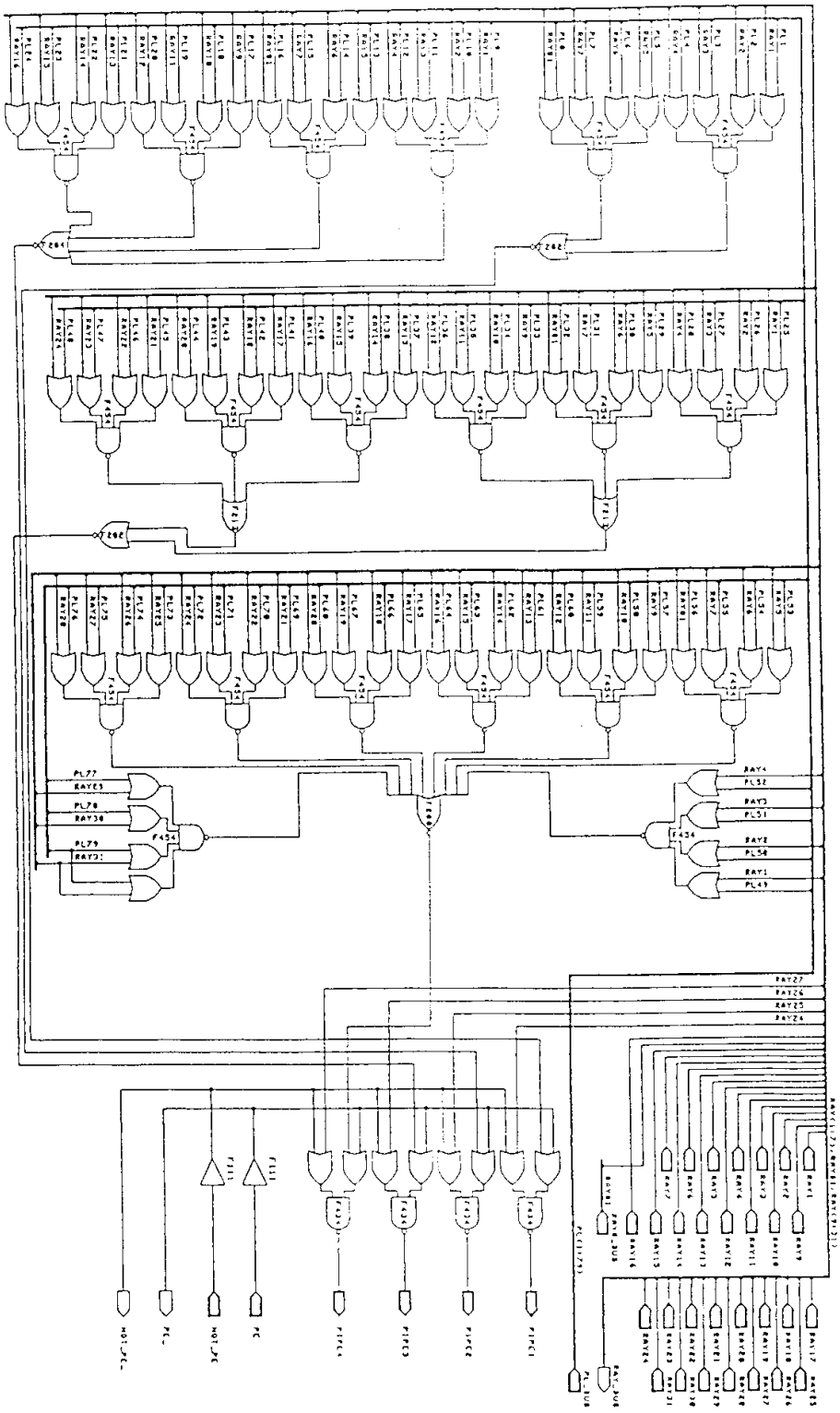


Fig. C.9: Block 'OR', Part 1

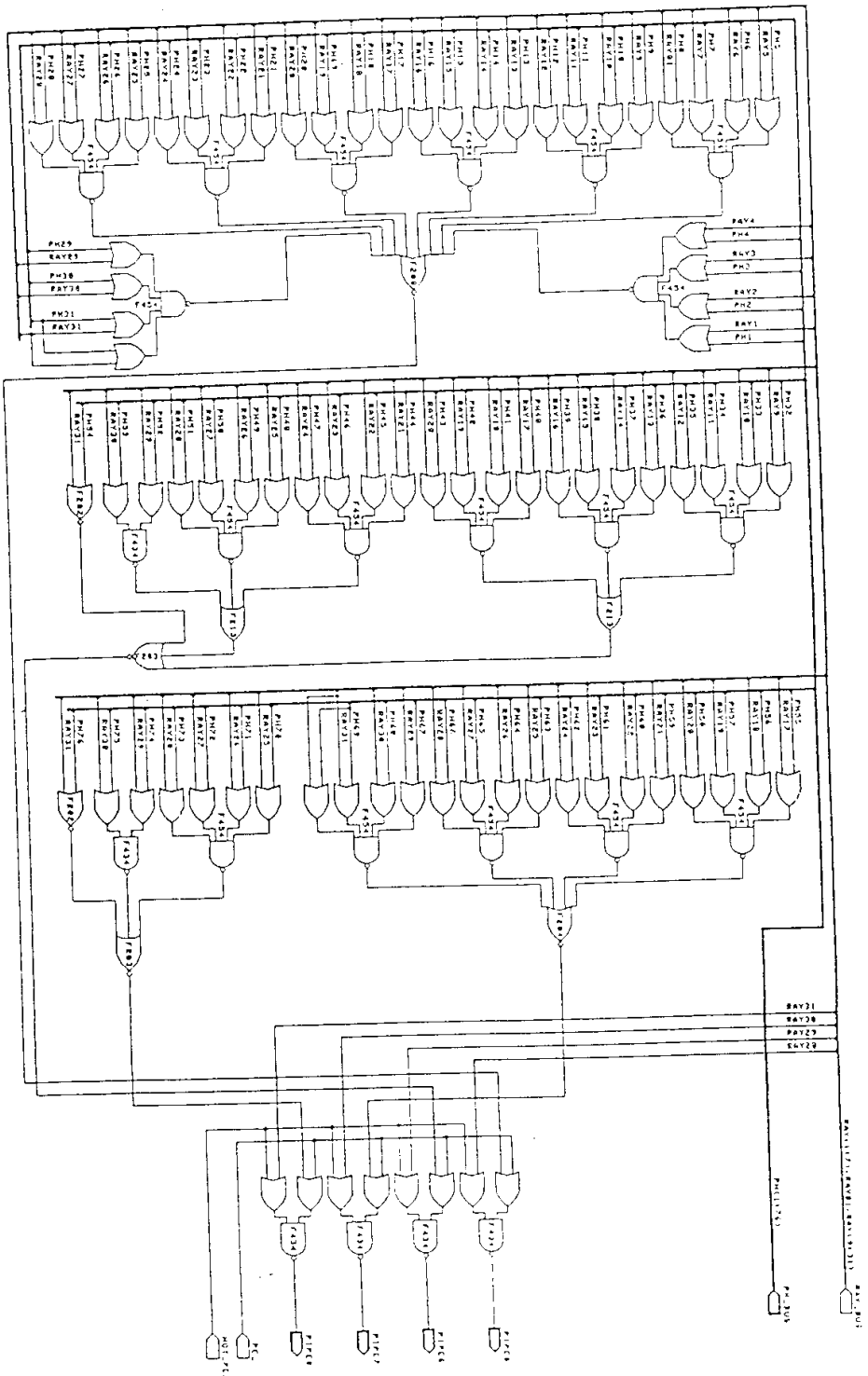


Fig. C.10: Block 'OR', part 2

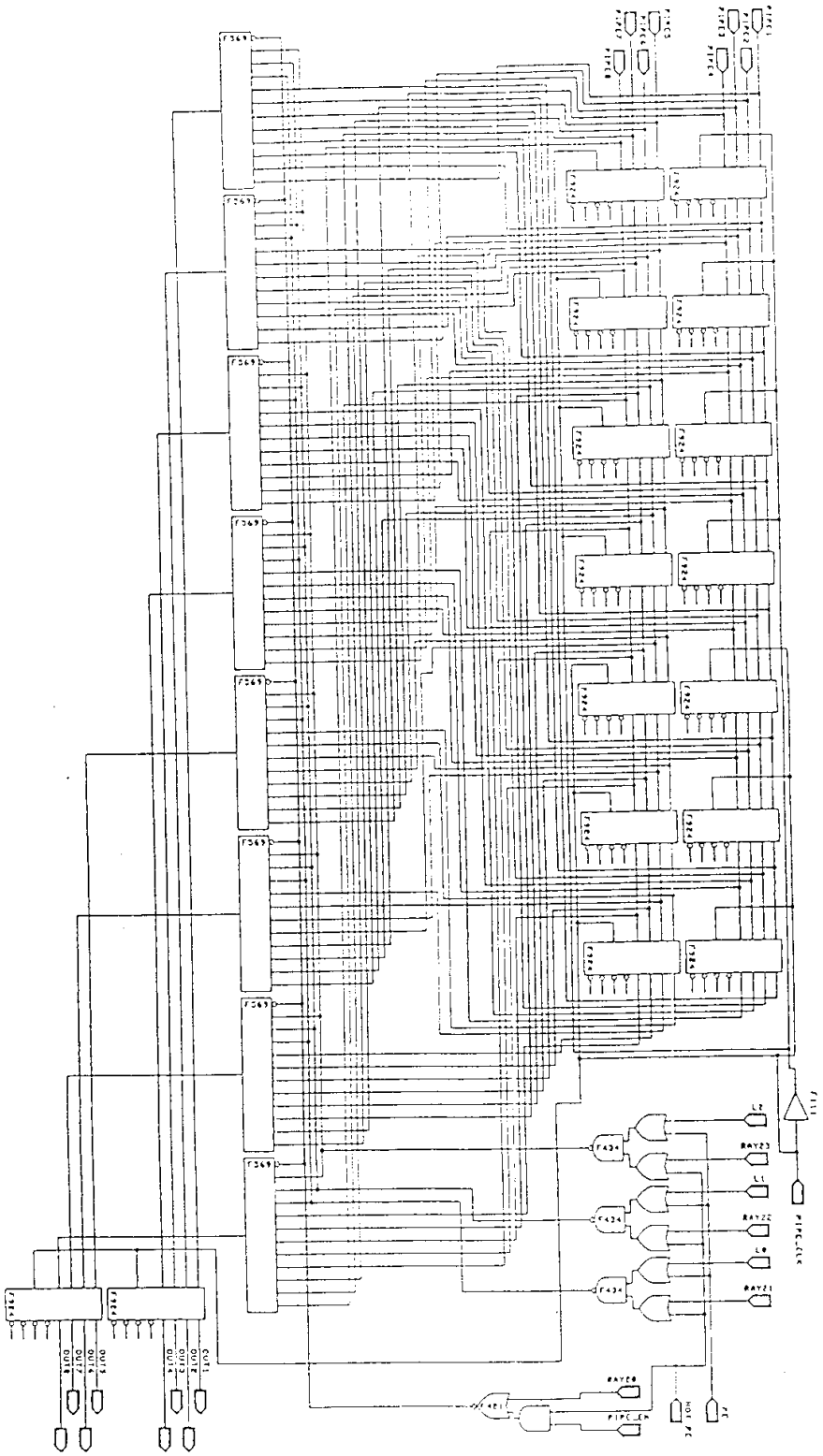


Fig. C.11: Block 'Pipeline'

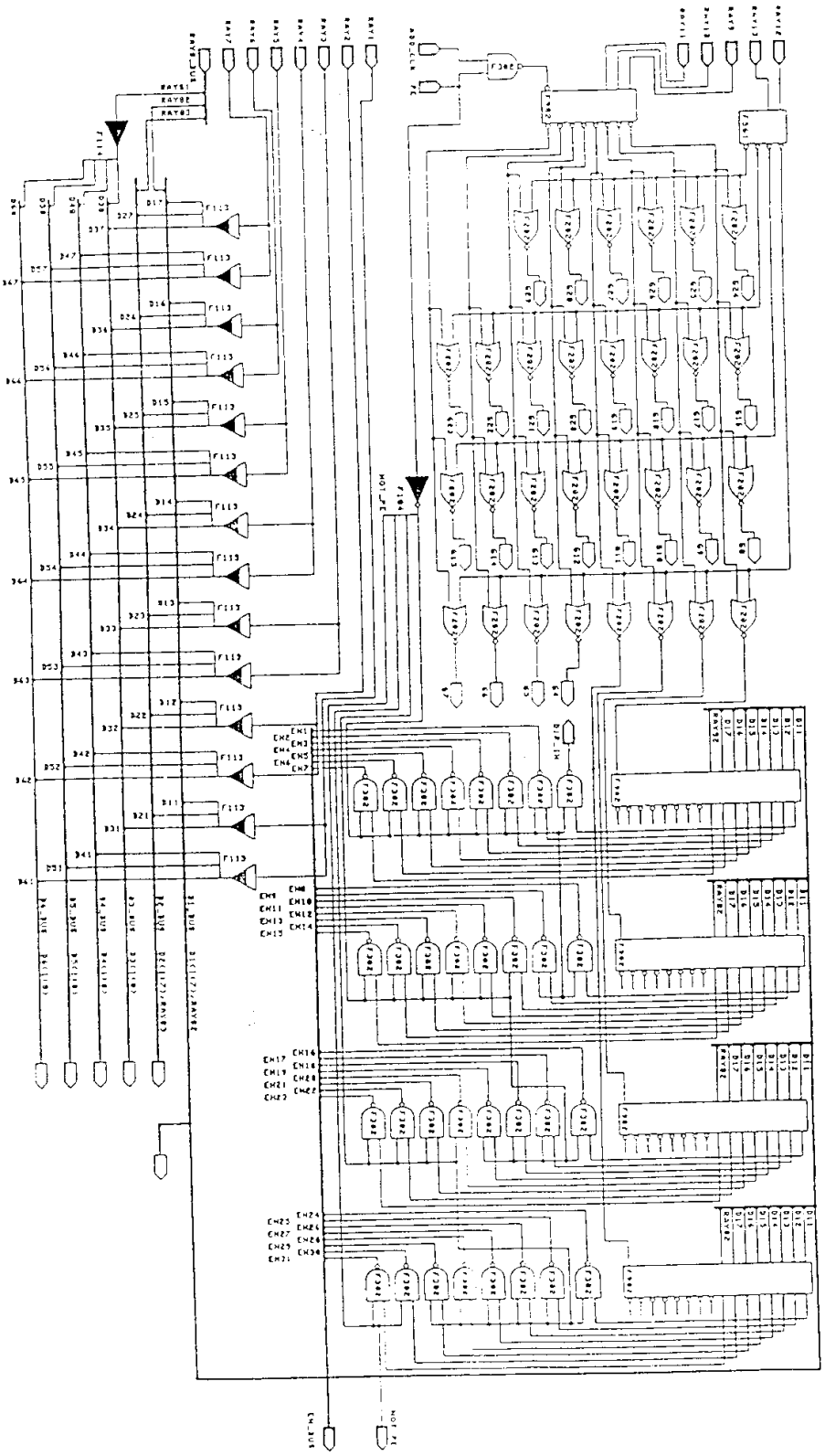


Fig. C.12: Block 'Config', part 1

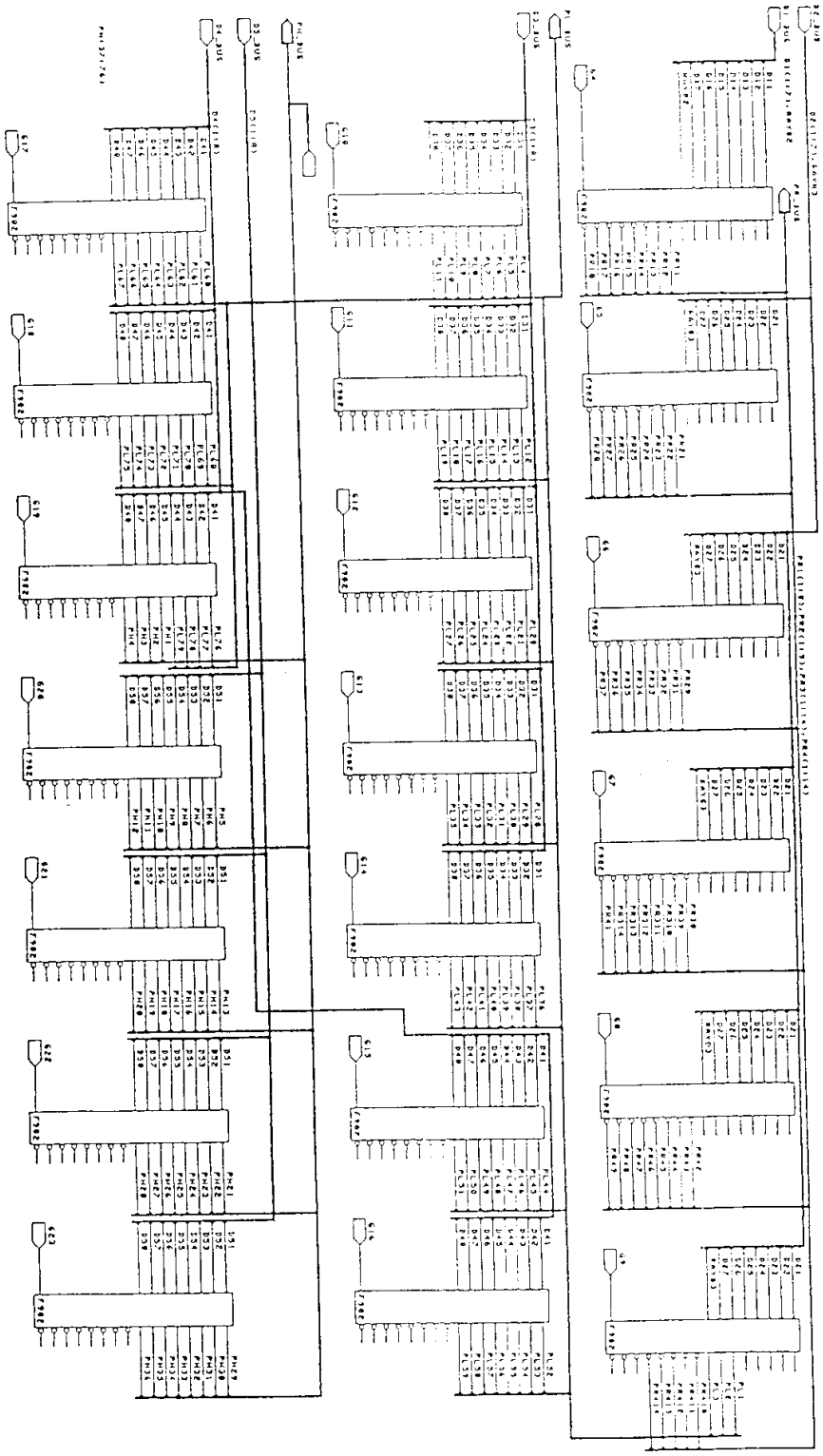


Fig. C.13: Block 'Config', part 2

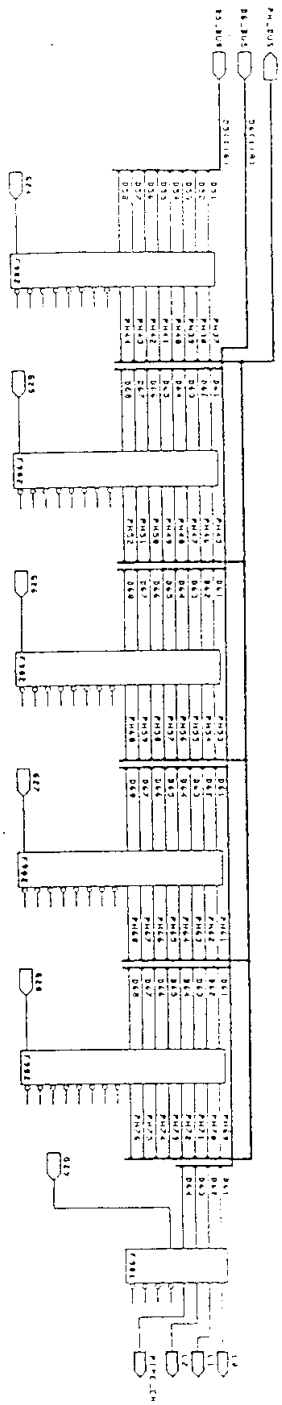


Fig. C.14: Block 'Config', part 3

Appendix D: Programmable Bits of the Gate Array

Address = 0
- Bit 0 = Mode Control (1)
- Bit 1 = Enable Ray 1
- Bit 2 = Enable Ray 2
- Bit 3 = Enable Ray 3
- Bit 4 = Enable Ray 4
- Bit 5 = Enable Ray 5
- Bit 6 = Enable Ray 6
- Bit 7 = Enable Ray 7

(1) High = Direct mode

Address = 1
- Bit 0 = Enable Ray 8
- Bit 1 = Enable Ray 9
- Bit 2 = Enable Ray 10
- Bit 3 = Enable Ray 11
- Bit 4 = Enable Ray 12
- Bit 5 = Enable Ray 13
- Bit 6 = Enable Ray 14
- Bit 7 = Enable Ray 15

Address = 2
- Bit 0 = Enable Ray 16
- Bit 1 = Enable Ray 17
- Bit 2 = Enable Ray 18
- Bit 3 = Enable Ray 19
- Bit 4 = Enable Ray 20
- Bit 5 = Enable Ray 21
- Bit 6 = Enable Ray 22
- Bit 7 = Enable Ray 23

Address = 3
- Bit 0 = Enable Ray 24
- Bit 1 = Enable Ray 25
- Bit 2 = Enable Ray 26
- Bit 3 = Enable Ray 27
- Bit 4 = Enable Ray 28
- Bit 5 = Enable Ray 29
- Bit 6 = Enable Ray 30
- Bit 7 = Enable Ray 31

Address = 4
- Bit 0 = Preset P11
- Bit 1 = Preset P12
- Bit 2 = Preset P13
- Bit 3 = Preset P14
- Bit 4 = Preset P15
- Bit 5 = Preset P16
- Bit 6 = Preset P17
- Bit 7 = Preset P18

Address = 5
- Bit 0 = Preset P21
- Bit 1 = Preset P22
- Bit 2 = Preset P23
- Bit 3 = Preset P24
- Bit 4 = Preset P25
- Bit 5 = Preset P26
- Bit 6 = Preset P27
- Bit 7 = Preset P28

Address = 6
- Bit 0 = Preset P29
- Bit 1 = Preset P31
- Bit 2 = Preset P32
- Bit 3 = Preset P33
- Bit 4 = Preset P34
- Bit 5 = Preset P35
- Bit 6 = Preset P36
- Bit 7 = Preset P37

Address = 7
- Bit 0 = Preset P38
- Bit 1 = Preset P39
- Bit 2 = Preset P310
- Bit 3 = Preset P311
- Bit 4 = Preset P312
- Bit 5 = Preset P313
- Bit 6 = Preset P314
- Bit 7 = Preset P41

Address = 8
- Bit 0 = Preset P42
- Bit 1 = Preset P43
- Bit 2 = Preset P44
- Bit 3 = Preset P45
- Bit 4 = Preset P46
- Bit 5 = Preset P47
- Bit 6 = Preset P48
- Bit 7 = Preset P49

Address = 9
- Bit 0 = Preset P410
- Bit 1 = Preset P411
- Bit 2 = Preset P412
- Bit 3 = Preset P413
- Bit 4 = Preset P414
- Bit 5 = Ray 1 to Pipe 1
- Bit 6 = Ray 2 to Pipe 1
- Bit 7 = Ray 3 to Pipe 1

Address = 10

Bit 0 = Ray 4 to Pipe 1
Bit 1 = Ray 5 to Pipe 1
Bit 2 = Ray 6 to Pipe 1
Bit 3 = Ray 7 to Pipe 1
Bit 4 = Ray 8 to Pipe 1
Bit 5 = Ray 1 to Pipe 2
Bit 6 = Ray 2 to Pipe 2
Bit 7 = Ray 3 to Pipe 2

Address = 11

Bit 0 = Ray 4 to Pipe 2
Bit 1 = Ray 5 to Pipe 2
Bit 2 = Ray 6 to Pipe 2
Bit 3 = Ray 7 to Pipe 2
Bit 4 = Ray 8 to Pipe 2
Bit 5 = Ray 9 to Pipe 2
Bit 6 = Ray 10 to Pipe 2
Bit 7 = Ray 11 to Pipe 2

Address = 12

Bit 0 = Ray 12 to Pipe 2
Bit 1 = Ray 13 to Pipe 2
Bit 2 = Ray 14 to Pipe 2
Bit 3 = Ray 15 to Pipe 2
Bit 4 = Ray 16 to Pipe 2
Bit 5 = Ray 1 to Pipe 3
Bit 6 = Ray 2 to Pipe 3
Bit 7 = Ray 3 to Pipe 3

Address = 13

Bit 0 = Ray 4 to Pipe 3
Bit 1 = Ray 5 to Pipe 3
Bit 2 = Ray 6 to Pipe 3
Bit 3 = Ray 7 to Pipe 3
Bit 4 = Ray 8 to Pipe 3
Bit 5 = Ray 9 to Pipe 3
Bit 6 = Ray 10 to Pipe 3
Bit 7 = Ray 11 to Pipe 3

Address = 14

Bit 0 = Ray 12 to Pipe 3
Bit 1 = Ray 13 to Pipe 3
Bit 2 = Ray 14 to Pipe 3
Bit 3 = Ray 15 to Pipe 3
Bit 4 = Ray 16 to Pipe 3
Bit 5 = Ray 17 to Pipe 3
Bit 6 = Ray 18 to Pipe 3
Bit 7 = Ray 19 to Pipe 3

Address = 15

Bit 0 = Ray 20 to Pipe 3
Bit 1 = Ray 21 to Pipe 3
Bit 2 = Ray 22 to Pipe 3
Bit 3 = Ray 23 to Pipe 3
Bit 4 = Ray 24 to Pipe 3
Bit 5 = Ray 1 to Pipe 4
Bit 6 = Ray 2 to Pipe 4
Bit 7 = Ray 3 to Pipe 4

Address = 16

Bit 0 = Ray 4 to Pipe 4
Bit 1 = Ray 5 to Pipe 4
Bit 2 = Ray 6 to Pipe 4
Bit 3 = Ray 7 to Pipe 4
Bit 4 = Ray 8 to Pipe 4
Bit 5 = Ray 9 to Pipe 4
Bit 6 = Ray 10 to Pipe 4
Bit 7 = Ray 11 to Pipe 4

Address = 17

Bit 0 = Ray 12 to Pipe 4
Bit 1 = Ray 13 to Pipe 4
Bit 2 = Ray 14 to Pipe 4
Bit 3 = Ray 15 to Pipe 4
Bit 4 = Ray 16 to Pipe 4
Bit 5 = Ray 17 to Pipe 4
Bit 6 = Ray 18 to Pipe 4
Bit 7 = Ray 19 to Pipe 4

Address = 18

Bit 0 = Ray 20 to Pipe 4
Bit 1 = Ray 21 to Pipe 4
Bit 2 = Ray 22 to Pipe 4
Bit 3 = Ray 23 to Pipe 4
Bit 4 = Ray 24 to Pipe 4
Bit 5 = Ray 25 to Pipe 4
Bit 6 = Ray 26 to Pipe 4
Bit 7 = Ray 27 to Pipe 4

Address = 19

Bit 0 = Ray 28 to Pipe 4
Bit 1 = Ray 29 to Pipe 4
Bit 2 = Ray 30 to Pipe 4
Bit 3 = Ray 31 to Pipe 4
Bit 4 = Ray 1 to Pipe 5
Bit 5 = Ray 2 to Pipe 5
Bit 6 = Ray 3 to Pipe 5
Bit 7 = Ray 4 to Pipe 5

Address = 20

Bit 0 = Ray 5 to Pipe 5
Bit 1 = Ray 6 to Pipe 5
Bit 2 = Ray 7 to Pipe 5
Bit 3 = Ray 8 to Pipe 5
Bit 4 = Ray 9 to Pipe 5
Bit 5 = Ray 10 to Pipe 5
Bit 6 = Ray 11 to Pipe 5
Bit 7 = Ray 12 to Pipe 5

Address = 21

Bit 0 = Ray 13 to Pipe 5
Bit 1 = Ray 14 to Pipe 5
Bit 2 = Ray 15 to Pipe 5
Bit 3 = Ray 16 to Pipe 5
Bit 4 = Ray 17 to Pipe 5
Bit 5 = Ray 18 to Pipe 5
Bit 6 = Ray 19 to Pipe 5
Bit 7 = Ray 20 to Pipe 5

Address = 22

Bit 0 = Ray 21 to Pipe 5
Bit 1 = Ray 22 to Pipe 5
Bit 2 = Ray 23 to Pipe 5
Bit 3 = Ray 24 to Pipe 5
Bit 4 = Ray 25 to Pipe 5
Bit 5 = Ray 26 to Pipe 5
Bit 6 = Ray 27 to Pipe 5
Bit 7 = Ray 28 to Pipe 5

Address = 23

Bit 0 = Ray 29 to Pipe 5
Bit 1 = Ray 30 to Pipe 5
Bit 2 = Ray 31 to Pipe 5
Bit 3 = Ray 9 to Pipe 6
Bit 4 = Ray 10 to Pipe 6
Bit 5 = Ray 11 to Pipe 6
Bit 6 = Ray 12 to Pipe 6
Bit 7 = Ray 13 to Pipe 6

Address = 24

Bit 0 = Ray 14 to Pipe 6
Bit 1 = Ray 15 to Pipe 6
Bit 2 = Ray 16 to Pipe 6
Bit 3 = Ray 17 to Pipe 6
Bit 4 = Ray 18 to Pipe 6
Bit 5 = Ray 19 to Pipe 6
Bit 6 = Ray 20 to Pipe 6
Bit 7 = Ray 21 to Pipe 6

Address = 25

Bit 0 = Ray 22 to Pipe 6
Bit 1 = Ray 23 to Pipe 6
Bit 2 = Ray 24 to Pipe 6
Bit 3 = Ray 25 to Pipe 6
Bit 4 = Ray 26 to Pipe 6
Bit 5 = Ray 27 to Pipe 6
Bit 6 = Ray 28 to Pipe 6
Bit 7 = Ray 29 to Pipe 6

Address = 26

Bit 0 = Ray 30 to Pipe 6
Bit 1 = Ray 31 to Pipe 6
Bit 2 = Ray 17 to Pipe 7
Bit 3 = Ray 18 to Pipe 7
Bit 4 = Ray 19 to Pipe 7
Bit 5 = Ray 20 to Pipe 7
Bit 6 = Ray 21 to Pipe 7
Bit 7 = Ray 22 to Pipe 7

Address = 27

Bit 0 = Ray 23 to Pipe 7
Bit 1 = Ray 24 to Pipe 7
Bit 2 = Ray 25 to Pipe 7
Bit 3 = Ray 26 to Pipe 7
Bit 4 = Ray 27 to Pipe 7
Bit 5 = Ray 28 to Pipe 7
Bit 6 = Ray 29 to Pipe 7
Bit 7 = Ray 30 to Pipe 7

Address = 28

Bit 0 = Ray 31 to Pipe 7
Bit 1 = Ray 25 to Pipe 8
Bit 2 = Ray 26 to Pipe 8
Bit 3 = Ray 27 to Pipe 8
Bit 4 = Ray 28 to Pipe 8
Bit 5 = Ray 29 to Pipe 8
Bit 6 = Ray 30 to Pipe 8
Bit 7 = Ray 31 to Pipe 8

Address = 29

Bit 0 = Length (Bit 0)
Bit 1 = Length (Bit 1)
Bit 2 = Length (Bit 2)
Bit 3 = Pipe Enable* (2)
Bit 4 : NA
Bit 5 : NA
Bit 6 : NA
Bit 7 : NA

(2) Low = Reassign ray 20 as 'Pipeline Enable'

Address = 30: NA

Address = 31: NA

(NA = Not applicable)

Appendix E: Ray Allocation for the Ray Finder Prototype

Remark: - Pin reassignments during PE = High are only given in cases where they differ from the PE = Low state.
- 'GA' = Gate Array

Gate Array Chip 1

Ray	Ch.1	Ch.2	Ch.3	Ch.4	Big Tower Min/Max
1	1023	2022	3001	4001	14/ 14
2	1023	2023	3001	4001	14/ 14
3	1023	2023	3002	4001	14/ 14
4	1024	2023	3002	4001	14/ 14
5	1024	2023	3002	4002	14/ 14
6	1024	2024	3002	4002	14/ 14
7	1024	2024	3003	4002	14/ 14
8	1025	2024	3003	4002	14/ 14
9	1025	2024	3003	4003	13/ 14
10	1025	2025	3003	4003	14/ 14
11	1025	2025	3004	4003	13/ 14
12	1026	2025	3004	4003	13/ 14
13	1026	2025	3004	4004	13/ 14
14	1026	2026	3004	4004	13/ 14
15	1026	2026	3005	4004	13/ 13
16	1027	2026	3005	4004	13/ 13
17	1027	2026	3005	4005	13/ 13
18	1027	2027	3005	4005	13/ 13
19	1027	2027	3006	4005	13/ 13
20	1028	2027	3006	4005	13/ 13
21	1028	2027	3006	4006	12/ 13
22	1028	2028	3006	4006	12/ 13
23	1028	2028	3007	4006	12/ 12
24	1029	2028	3007	4006	12/ 12
25	1029	2028	3007	4007	11/ 12
26	1029	2029	3007	4007	11/ 12
27	1029	2029	3008	4007	11/ 11
28	1030	2029	3008	4007	11/ 11
29	1030	2029	3008	4008	10/ 11
30	1030	2030	3008	4008	10/ 12
31	1030	2030	3009	4008	10/ 11

Order of channel exchange : 1 2 4 3
Number of used rays : 31
Number of free rays : 0

Big Towers:
Min. : 10
Max. : 14

Pin Allocation Table

				PE: low	high			low	high
P11	= 1023	P21	= 2022	P31	= 4001		P41	= 3001	
P12	= 1024	P22	= 2023	P32	= 4001	4005	P42	= 3002	
P13	= 1025	P23	= 2024	P33	= 4002		P43	= 3002	3005
P14	= 1026	P24	= 2025	P34	= 4002	4006	P44	= 3003	
P15	= 1027	P25	= 2026	P35	= 4003		P45	= 3003	3006
P16	= 1028	P26	= 2027	P36	= 4003	4007	P46	= 3004	
P17	= 1029	P27	= 2028	P37	= 4004		P47	= 3004	
P18	= 1030	P28	= 2029	P38	= 4004		P48	= 3005	
		P29	= 2030	P39	= 4005		P49	= 3005	
				P310	= 4005		P410	= 3006	
				P311	= 4006		P411	= 3006	
				P312	= 4006		P412	= 3007	
				P313	= 4007		P413	= 3008	
				P314	= 4008		P414	= 3009	

Out 1 to P31 of GA 9
Out 2 to P41 of GA 9
Out 3 to P32 of GA 9
Out 4 to P42 of GA 9

Out 5 to P33 of GA 9
Out 6 to P35 of GA 9
Out 7 to P39 of GA 9
Out 8 to P27 of GA 9

Gate Array Chip 2

Ray	Ch.1	Ch.2	Ch.3	Ch.4	Big Tower Min/Max
1	1031	2030	3009	4009	9/ 11
2	<wrong>				
3	1031	2030	3010	4009	9/ 9
4	1031	2031	3010	4009	9/ 10
5	1031	2031	3009	4009	9/ 11
6	<wrong>				
7	1031	2031	3010	4010	8/ 10
8	1031	2032	3010	4010	8/ 8
9	1031	2032	3010	4011	8/ 8
10	1032	2032	3010	4011	8/ 8
11	<wrong>				
12	1032	2031	3010	4010	8/ 10
13	<unused>	(2)			
14	<unused>	(2)			
15	<wrong>				
16	1032	2032	3011	4010	8/ 8
17	1032	2032	3011	4011	7/ 8
18	<wrong>				
19	<wrong>				
20	1032	2033	3011	4011	7/ 8
21	1032	2033	3011	4012	7/ 7
22	1033	2033	3011	4012	7/ 7
23	1033	2033	3012	4012	6/ 7
24	1033	2034	3012	4012	6/ 7
25	1033	2034	3012	4013	6/ 6
26	1034	2034	3012	4013	6/ 6
27	1034	2034	3013	4013	5/ 6
28	1034	2035	3013	4013	5/ 6
29	1034	2035	3013	4014	5/ 5
30	1035	2035	3013	4014	5/ 6
31	1035	2035	3014	4014	5/ 5

Order of channel exchange : 2 1 4 3

Number of used rays : 23

Number of free rays : 8

Big Towers:
Min. : 5
Max. : 11

Pin Allocation Table

				PE: low	high		low	high
P11	= 2030	P21	= 1031	P31	= 4009		P41 = 3009	
P12	= 2031	P22	= 1031	P32	= 4009	4013	P42 = 3010	
P13	= 2032	P23	= 1031	P33	= 4009	4014	P43 = 3009	3011
P14	= 2031	P24	= 1032	P34	= 4010		P44 = 3010	3012
P15	= 2032	P25	= 1032	P35	= 4011		P45 = 3010	3013
P16	= 2033	P26	= 1032	P36	= 4010	4015	P46 = 3010	3014
P17	= 2034	P27	= 1033	P37	= 4012		P47 = Gnd	
P18	= 2035	P28	= 1034	P38	= 4010		P48 = 3011	
		P29	= 1035	P39	= 4011		P49 = 3011	
				P310	= 4011		P410 = 3011	
				P311	= 4012		P411 = 3011	
				P312	= 4012		P412 = 3012	
				P313	= 4013		P413 = 3013	
				P314	= 4014		P414 = 3014	

Out 1 to P45 of GA 9 and P31 of GA 10
Out 2 to P49 of GA 9 and P32 of GA 10
Out 3 to P313 of GA 9 and P43 of GA 10
Out 4 to P413 of GA 9 and P44 of GA 10
Out 5 to P34 of GA 9 and P41 of GA 10
Out 6 to P36 of GA 9 and P42 of GA 10
Out 7 to P310 of GA 9 and P34 of GA 10
Out 8 to P28 of GA 9 and P45 of GA 10

Gate Array Chip 3

Ray	Ch.1	Ch.2	Ch.3	Ch.4	Big Tower Min/Max
1	<unused>	(1)			
2	1041	2041	5020	6020	3/ 3
3	<wrong>				
4	1041	2041	5020	6019	3/ 3
5	1041	2041	5019	6019	3/ 3
6	<wrong>				
7	<wrong>				
8	1040	2041	5020	6020	3/ 3
9	<wrong>				
10	1040	2041	5020	6019	3/ 3
11	1040	2041	3019	4019	3/ 3
12	1040	2040	3019	4019	3/ 3
13	1040	2040	3018	4019	3/ 3
14	1039	2040	3018	4019	3/ 3
15	1039	2040	3018	4018	3/ 3
16	1039	2039	3018	4018	3/ 3
17	1039	2039	3017	4018	3/ 3
18	1038	2039	3017	4018	3/ 3
19	1038	2039	3017	4017	3/ 4
20	1038	2038	3017	4017	3/ 4
21	1038	2038	3016	4017	4/ 4
22	1037	2038	3016	4017	4/ 4
23	1037	2038	3016	4016	4/ 4
24	1037	2037	3016	4016	4/ 4
25	1037	2037	3015	4016	4/ 4
26	1036	2037	3015	4016	4/ 4
27	1036	2037	3015	4015	4/ 5
28	1036	2036	3015	4015	4/ 5
29	1036	2036	3014	4015	5/ 5
30	1035	2036	3014	4015	5/ 5
31	1035	2036	3014	4014	5/ 5

Order of channel exchange : 2 1 3 4

Number of used rays : 26

Number of free rays : 5

Big Towers:
Min. : 3
Max. : 5

Pin Allocation Table

				PE: low	high		low	high
P11	= 2041	P21	= Gnd	P31	= 5020		P41 = 6020	
P12	= 2041	P22	= 1041	P32	= 5020	3017	P42 = 6019	
P13	= 2041	P23	= 1040	P33	= 5019		P43 = 6019	4018
P14	= 2040	P24	= 1040	P34	= 5020	3016	P44 = 6020	4017
P15	= 2039	P25	= 1039	P35	= 5020	3015	P45 = 6019	4016
P16	= 2038	P26	= 1038	P36	= 3019		P46 = 4019	
P17	= 2037	P27	= 1037	P37	= 3018		P47 = 4019	
P18	= 2036	P28	= 1036	P38	= 3018		P48 = 4018	
		P29	= 1035	P39	= 3017		P49 = 4018	
				P310	= 3017		P410 = 4017	
				P311	= 3016		P411 = 4017	
				P312	= 3016		P412 = 4016	
				P313	= 3015		P413 = 4015	
				P314	= 3014		P414 = 4014	

Out 1 to P37 of GA 9
Out 2 to P43 of GA 9 and P313 of GA 10
Out 3 to P44 of GA 9 and P411 of GA 10
Out 4 to P46 of GA 9 and P39 of GA 10
Out 5 to P47 of GA 10
Out 6 to P36 of GA 10
Out 7 to P35 of GA 10
Out 8 to P33 of GA 10

Gate Array Chip 4

Ray	Ch.1	Ch.2	Ch.3	Ch.4	Big Tower Min/Max
1	<wrong>				
2	<wrong>				
3	1041	2042	501S	601S	3/ 3
4	<wrong>				
5	1041	2042	5019	601S	3/ 3
6	<wrong>				
7	1041	2042	5019	6019	3/ 3
8	<wrong>				
9	1042	2042	501S	6017	2/ 3
10	<wrong>				
11	1042	2042	501S	6018	3/ 3
12	<wrong>				
13	1042	2042	5019	6018	3/ 3
14	<wrong>				
15	1042	2043	5017	6017	2/ 2
16	<wrong>				
17	1042	2043	501S	6017	2/ 3
18	<wrong>				
19	1042	2043	501S	601S	3/ 3
20	<wrong>				
21	1043	2043	5017	6016	2/ 2
22	<wrong>				
23	1043	2043	5017	6017	2/ 2
24	<wrong>				
25	<wrong>				
26	1043	2044	5017	6016	2/ 2
27	1043	2044	5016	6016	2/ 2
28	1044	2044	5016	6016	2/ 2
29	1044	2044	5016	6015	2/ 2
30	1044	2045	5016	6015	2/ 2
31	1044	2045	5015	6015	2/ 2

Order of channel exchange : 1 2 4 3

Number of used rays : 17

Number of free rays : 14

Big Towers:
Min. : 2
Max. : 3

Pin Allocation Table

				PE: low	high		low	high
P11	= 1041	P21	= 2041	P31 = 6020		P41	= 5020	
P12	= 1041	P22	= 2042	P32 = 6018		P42	= 5018	
P13	= 1042	P23	= 2042	P33 = 6018	6016	P43	= 5019	
P14	= 1042	P24	= 2042	P34 = 6019		P44	= 5019	5017
P15	= 1042	P25	= 2043	P35 = 6017		P45	= 5018	5016
P16	= 1043	P26	= 2043	P36 = 6018	6015	P46	= 5018	5015
P17	= 1043	P27	= 2043	P37 = 6018	6014	P47	= 5019	
P18	= 1044	P28	= 2044	P38 = 6017		P48	= 5017	
		P29	= 2045	P39 = 6017		P49	= 5018	
				P310 = 6018		P410	= 5018	
				P311 = 6016		P411	= 5017	
				P312 = 6017		P412	= 5017	
				P313 = 6016		P413	= 5016	
				P314 = 6015		P414	= 5015	

Out 1 to -
Out 2 to P36 of GA 10
Out 3 to P47 of GA 10
Out 4 to P48 of GA 10
Out 5 to P312 of GA 10
Out 6 to P414 of GA 10
Out 7 to -
Out 8 to -

Gate Array Chip 5

Ray	Ch.1	Ch.2	Ch.3	Ch.4	Big Tower Min/Max
1	<unused>	(1)			
2	1050	2052	5010	6009	2/ 2
3	<wrong>				
4	1050	2051	5010	6010	2/ 2
5	<wrong>				
6	1049	2051	5011	6010	2/ 2
7	1049	2051	5010	6010	2/ 2
8	<wrong>				
9	<wrong>				
10	1049	2050	5011	6010	2/ 2
11	<wrong>				
12	1049	2049	5011	6011	2/ 2
13	1049	2049	5011	6010	2/ 2
14	<wrong>				
15	1048	2049	5012	6011	2/ 2
16	1048	2048	5012	6011	2/ 2
17	1048	2048	5012	6012	2/ 2
18	1047	2048	5012	6012	2/ 2
19	1047	2048	5013	6012	2/ 2
20	1047	2047	5013	6012	2/ 2
21	1047	2047	5013	6013	2/ 2
22	1046	2047	5013	6013	2/ 2
23	1046	2047	5014	6013	2/ 2
24	1046	2046	5014	6013	2/ 2
25	1046	2046	5014	6014	2/ 2
26	1045	2046	5014	6014	2/ 2
27	1045	2046	5015	6014	2/ 2
28	1045	2045	5015	6014	2/ 2
29	1045	2045	5015	6015	2/ 2
30	<wrong>				
31	<wrong>				

Order of channel exchange : 2 1 4 3

Number of used rays : 22

Number of free rays : 9

Big Towers:
Min. : 2
Max. : 2

Pin Allocation Table

				PE: low	high		low	high
P11	= 2052	P21	= Gnd	P31	= 6009		P41 = 5010	
P12	= 2051	P22	= 1050	P32	= 6010		P42 = 5010	5012
P13	= 2050	P23	= 1049	P33	= 6010	6012	P43 = 5011	
P14	= 2049	P24	= 1049	P34	= 6010	6013	P44 = 5010	5013
P15	= 2048	P25	= 1048	P35	= 6010	6014	P45 = 5011	5014
P16	= 2047	P26	= 1047	P36	= 6011		P46 = 5011	5015
P17	= 2046	P27	= 1046	P37	= 6010	6015	P47 = 5011	
P18	= 2045	P28	= 1045	P38	= 6011		P48 = 5012	
		P29	= 1044	P39	= 6012		P49 = 5012	
				P310	= 6012		P410 = 5013	
				P311	= 6013		P411 = 5013	
				P312	= 6013		P412 = 5014	
				P313	= 6014		P413 = 5015	
				P314	= 6015		P414 = 5016	

Out 1 -
Out 2 to P38 of GA 10
Out 3 to P310 of GA 10
Out 4 to P412 of GA 10
Out 5 to P28 of GA 10
Out 6 to -
Out 7 to -
Out 8 to -

Gate Array Chip 6

Ray	Ch.1	Ch.2	Ch.3	Ch.4	Big Tower Min/Max	Ray	Ch.1	Ch.2	Ch.3	Ch.4	Big Tower Min/Max
1	1051	2052	5009	6009	2/ 2	17	1056	205S	5006	6006	1/ 1
2	<wrong>					18	<wrong>				
3	1051	2053	5009	6009	2/ 2	19	1057	205S	5006	6005	1/ 1
4	<wrong>					20	<wrong>				
5	<wrong>					21	1057	2059	5006	6005	1/ 1
6	1052	2053	5009	600S	2/ 2	22	<wrong>				
7	<wrong>					23	<wrong>				
8	1053	2054	5008	600S	2/ 2	24	105S	2059	5005	6005	1/ 1
9	<wrong>					25	105S	2060	5005	6005	1/ 1
10	1053	2055	5008	6007	1/ 2	26	<wrong>				
11	<wrong>					27	1059	2060	5005	6004	1/ 1
12	1054	2056	5007	6007	1/ 1	28	<unused>	(1)			
13	<wrong>					29	<unused>	(2)			
14	1055	2057	5007	6006	1/ 1	30	<unused>	(3)			
15	<wrong>					31	<unused>	(4)			
16	1056	2057	5006	6006	1/ 1						

Order of channel exchange : 4 3 1 2

Number of used rays : 14

Number of free rays : 17

Big Towers:
Min. : 1
Max. : 2

Pin Allocation Table

				PE: low	high		low	high
P11	= 5009	P21	= 6009	P31	= 2052	P41	= 1051	
P12	= 5009	P22	= 6009	P32	= 2053	P42	= 1051	1055
P13	= 5008	P23	= 600S	P33	= 2053	205S P43	= 1052	
P14	= 5007	P24	= 6007	P34	= 2054	P44	= 1053	
P15	= 5006	P25	= 6006	P35	= 2055	P45	= 1053	1056
P16	= 5006	P26	= 6005	P36	= 2056	P46	= 1054	
P17	= 5005	P27	= 6005	P37	= 2057	P47	= 1055	
P18	= Gnd	P28	= 6004	P38	= 2057	P48	= 1056	
		P29	= Gnd	P39	= 205S	P49	= 1056	
				P310	= 205S	P410	= 1057	
				P311	= 2059	P411	= 1057	
				P312	= 2059	P412	= 105S	
				P313	= 2060	P413	= 1059	
				P314	= Gnd	P414	= Gnd	

Out 1 to -
Out 2 to P48 of GA 10
Out 3 to P311 of GA 10
Out 4 to P27 of GA 10
Out 5 to P314 of GA 10
Out 6 to -
Out 7 to -
Out 8 to -

Gate Array Chips 7 and 8: not needed

Big Tower Allocation

Gate Array Chip 9

Big tower generation, second part
Out 1 ... Out S are Big 1 ... Big S

Input Signals:

P21	= Gnd	P31	= Out 1/GA 1	P41	= Out 2/GA 1
P22	= Gnd	P32	= Out 3/GA 1	P42	= Out 4/GA 1
P23	= Gnd	P33	= Out 5/GA 1	P43	= Out 2/GA 3
P24	= Gnd	P34	= Out 5/GA 2	P44	= Out 3/GA 3
P25	= Gnd	P35	= Out 6/GA 1	P45	= Out 1/GA 2
P26	= Gnd	P36	= Out 6/GA 1	P46	= Out 4/GA 3
P27	= Out 8/GA 1	P37	= Out 1/GA 3	P47	= Gnd
P28	= Out 8/GA 2	P38	= Gnd	P48	= Gnd
P29	= Gnd	P39	= Out 7/GA 1	P49	= Out 2/GA 2
		P310	= Out 7/GA 2	P410	= Bin Select
		P311	= Gnd	P411	= Gnd
		P312	= Gnd	P412	= Gnd
		P313	= Out 3/GA 2	P413	= Out 4/GA 2
		P314	= Gnd	P414	= Gnd

P11 ... P18 = Gnd

Gate Array Chip 10

Big tower generation, second part
Out 1 ... Out 8 are Big 9 ... Big 16

Input Signals:

P21	= Gnd	P31	= Out 1/GA 2	P41	= Out 5/GA 2
P22	= Gnd	P32	= Out 2/GA 2	P42	= Out 6/GA 2
P23	= Gnd	P33	= Out 8/GA 3	P43	= Out 3/GA 2
P24	= Gnd	P34	= Out 7/GA 2	P44	= Out 4/GA 2
P25	= Gnd	P35	= Out 7/GA 3	P45	= Out 8/GA 2
P26	= Gnd	P36	= Out 6/GA 3	P46	= Out 2/GA 4
P27	= Out 4/GA 6	P37	= Out 5/GA 3	P47	= Out 3/GA 4
P28	= Out 5/GA 5	P38	= Out 2/GA 5	P48	= Out 2/GA 6
P29	= Gnd	P39	= Out 4/GA 3	P49	= Out 4/GA 4
		P310	= Out 3/GA 5	P410	= Bin Select
		P311	= Out 3/GA 6	P411	= Out 3/GA 3
		P312	= Out 5/GA 4	P412	= Out 4/GA 5
		P313	= Out 2/GA 3	P413	= Out 6/GA 4
		P314	= Out 5/GA 6	P414	= Gnd

P11 ... P18 = Gnd

Detailed schemes of the ray finder PCB are available but occupy much larger paper formats than would be appropriate to include here.

Appendix F: Project Description for the Gate Array

The following document has been handed out to NEC together with the buying contract and commits them to meet te specified timing requirements.

Project Description of the Gate Array Project 'Ray Finder'

1. Overview

The Gate Array will be used in the trigger electronics of the new HERA electron-proton accelerator in Hamburg, Germany. A particle originating from a collision of an electron and a proton is seen as a ray in the detector as it fires four pads of multi-wire proportional chambers (MWPCs). The main purpose of the Gate Array under development is to analyze the signals from the MWPCs and to find out, how many rays were present and into which direction they pointed.

Collisions may occur every 96 ns, which implies an operating frequency for the entire electronics of 10 MHz.

2. Block Diagram

Pads from four groups of MWPCs are fed into the gate array at pins called P11 .. P18, P21 .. P29, P31 .. P314 and P41 .. P414 respectively. The INPUT block combines them to so called rays (Ray 1 to Ray 31). For each ray, this is done in the subblock '3 of 4', which requests either all four pads (i.g. P11 and P21 and P31 and P41 for Ray 1) to have fired or just 3 of them, depending on the programming of the corresponding latches in the block CONFIG.

Each ray has also an 'Enable' bit in the programming area CONFIG associated with it, so that it can be globally rejected. This is due to the pin count restriction, which required to use the same pin for different rays (i.g. P11 in Ray 1 to Ray 3), which may result in undesired combinations.

The '3 of 4' subblocks can be bypassed if the 'Dir_In' bit is set. This is to give the chip a more general interface for possible future applications.

The resulting rays are then fed into a cascade of adders in the block ADDER, which returns a 5 bit number equivalent to the number of rays detected.

Also, the rays are fed into 8 programmable ORs in the block OR, which associates the rays with 8 possible directions they are pointing at. This information is then stored by a pipeline in the block PIPELINE for up to 8 clock cycles of 96 ns each.

The programming of the OR as well as the length of the pipeline is also done in the block CONFIG, which is intended to operate under 'slow' conditions, since programming occurs only at power up time.

The chip is set into a program enable mode by the PE pin, which reassigns the pins P35, P36, P37, P45 and P46 as the address of the latch in the block CONFIG and the pins P31 .. P34 and P41 .. P44 as the data bits. Data is read in with the AClk, which, under normal operation conditions, is used as the clock of the latches at the end of the cascade of adders in the block ADDER.

Also, if PE is high, the pins P27 .. P29, P311 .. P314 and P410 .. P414 are reassigned, so that the pipeline can be used directly as a 8 bit wide general purpose shift register with selectable length.

3. Critical paths

3.1 Problem Description

Whereas the block CONFIG is not time critical at all, and the block OR and PIPELINE only insofar, as they should be clockable with 10 MHz, all paths from the inputs Pxx to the outputs H1 .. H5 have to be considered as important.

Surrounding electronics have 1500 ns available to feed the gate arrays inputs Pxx and to analyze the 5 bit number at H1 .. H5, but the entire circuit has to work deadtimeless, which requires pipelined, synchronous operation. This means latching, whenever minimum and maximum propagation delays under operating conditions differ by more than 96 ns.

The 1500 ns decision time has to be regarded as short, so that one has to economize time consummation, which can be achieved in two different ways:

- each subcircuit has to minimize its propagation delay.
- latching has to be minimized, which requires a trimming of different path lengths. Especially, one wants omit a latch just in front of the gate arrays Pxx inputs.

3.2 Propagation Delay Calculations

For the gate array, propagation delays have been calculated. The calculations were governed by the following rules:

- actual fan out
- wire length 3 mm
- inside the blocks ADD2 and ADD3 (subblocks of ADDER) and 3OF4 (subblock of INPUT), a wire length of 1 mm was used.

For the slowest path, typical propagation delays were calculated as follows:

Input Buffer FI01, F/O = 10 (e.g. P27)	2.90 ns
Buffer F111 in INPUT, F/O = 8 (e.g. P27 to '3 of 4' of ray 24)	2.50 ns
3 OF 4 Block, F/O = 3 (slow path through F442)	5.91 ns
ADDER up to D-Inputs of F611	21.90 ns
Total : T(max,theor.) =	33.21 ns

The AClk Signal needs 2.50 ns typically to the C-Inputs of the F611 flip flops. This increases to 2.74 ns under maximum operating conditions and minimum delay degrading due to process tolerances.

From this, one can see, that under maximum conditions (85 centigrades, 4.5V, worst case process), the clock should not come earlier than (2 * 33.21 + 2.10 - 2.74) = 65.78 ns after the data is available at the pins Pxx. (2.10 ns is the setup time for F611). In the test patterns, the clock comes 67 ns after the data, so that they should pass final delay time simulations with no problems.

However, if the actual time T(max,act.) is much shorter than T(max,theor.), this would not only be important to know, but also facilitate the circuit design around the gate array.

The fastest path is given by ray 2, which is directly fed into the Cin of the F523 4-bit adder in the block ADDER. The calculations are as follows (typical times):

Input Buffer FI01, F/O = 9 (e.g P31)	2.80 ns
3 OF 4 Block, F/O = 3 (through F304 at input, L->H change)	4.28 ns
ADDER up to D-Inputs of F611 (includes difference to actual F/O of '3 OF 4', which is 9)	3.41 ns
Total : T(min,theor.) =	10.49 ns

The difference between T(max) and T(min) defines the maximum allowable skew of the input signals. Under maximum operating conditions and minimal process-related delay degrading, T(min,theor.) increases to 11.50 ns, so that the maximum skew has to be below (96 - 2 * 33.21 + 11.50) = 41.09 ns.

However, several gate arrays will work in parallel, so that one can not guarantee the same temperature and supply voltage for all of them. But operating conditions can be restricted to (35 +- 10) centigrades and (5.25 +- 0.25)V, which implies, that T(max,theor.) is not above 49.32 ns and

T(min,theor.) not below 6.71 ns. From this, the allowed skew would be less than (96 - 49.32 + 6.71) = 53.39 ns = T(skew,operat.,theor.) This value can be regarded as acceptable, since expected input skew will be 30 ns, leaving ample time for security.

The behavior of the circuit as far as these calculations are concerned, can not be tested by other means than by observing the D-Inputs of the F611 flip flops. Test patterns are provided, that set all those inputs to high and then activate the fast path as described above. A change at the input pins is not allowed to have any influence to the D-Inputs of the flip flops earlier than T(min,theor.) after having been applied.

Of course, the path lengths could be much better trimmed by inserting delay lines into those signals which are directly fed into carry inputs of adders other than those at the top level. This would especially affect the signals Ray 1 .. 7 in ADDER. This has not been done so far for two reasons:

- The pin pair count is already in the critical area.
- Delay lines should only be so long that they do not increase T(max,act.), since this is the more critical parameter. T(max,act.) is expected to be below T(max,theor.), which has been calculated at rather pessimistic wire length assumptions as mentioned at the beginning of this chapter, since outputs of adders drive only one pin pair.

(Of course, a short T(max,act.) would also increase the allowable skew at the gate array inputs.)

However, if more realistic wire lengths allow better calculations of delay line lengths, one may feel free to introduce them.

Critical path definitions in the project information sheet (exhibit B of the contract) are only meant to support the above design goals. They are selected as to speed up the slowest paths in the ADDER block.

3.3 Guaranteed Propagation Delays

From the above calculations, one can see, that it is realistic to require:

- T(max,act.) < 35 ns under typical operating and process conditions (25 °C, 5.0 V)
- T(skew,operat.,act.) > 50 ns, where

  T(skew,operat.,act.) = (96ns - F(1)*T(max,act.) + F(2)*T(min,act.)) and

  F(1) = 1.48 = degrading factor for T = 45 °C, U = 5.0 V, worst case process

  F(2) = 0.64 = degrading factor for T = 25 °C, U = 5.5 V, best case process

  T(max,act.) and T(min,act.) under typical operating and process conditions (25 °C, 5.0 V).

Of course, better results would be greatly appreciated.

Dictionary of Abbreviations

AC	Advanced CMOS logic
ACT	Advanced CMOS logic with TTL levels
AS	Advanced Schottky TTL logic
ASIC	Application Specific IC
BiCMOS	Bipolar CMOS
CAD	Computer Aided Design
CC 100	Charged Current Events with Q^2 >= 100 GeV^2
CMOS	Complementary MOS
DESY	Deutsches Elektronen Synchrotron (Hamburg)
DIL	Dual In Line package
DIP	Dual In line Package
ECL	Emitter Coupled Logic
FACT	Fairchild Advanced CMOS Technology
FADC	Flash Analog Digital Converter
FAST	Fairchild Advanced Schottky Technology
Gnd	Ground (0 V)
HC	High speed CMOS logic
HCT	High speed CMOS logic with TTL levels
HERA	Hadronen Elektronen Ring Anlage
HU	Horizontal Unit (Euro mechanics, 2/10")
HV	High Voltage
IC	Integrated Circuit
MOS	Metal Oxid Semiconductor
MSI	Medium Scale Integration
MWPC	Multi Wire Proportional Chamber
NCG 100	Neutral Current events with Q^2 >= 100 GeV^2
NCG 1000	Neutral Current events with Q^2 >= 1000 GeV^2
PAL	Programmable Array Logic
PCB	Printed Circuit Board
PROM	Programmable ROM
RAM	Random Access Memory
ROM	Read Only Memory
S-DIP	Shrinked DIP
SIN	Schweizerisches Institut für Nuklearforschung
TTL	Transistor Transistor Logic
Vcc	+5 V DC
Vee	-5 V DC
VU	Vertical Unit (Euro mechanics)

Bibliography

[1] Letter of Intent, H1 Collaboration, June 28, 1985
[2] Charm, Bottom and Top Production in High Energy Electron Proton Collisions, by R. A. Eichler and Z. Kunszt, Institute for Intermediate Energy Physics, ETH Zurich, February 1988
[3] Technical Proposal for the H1 Detector, H1 Collaboration, March 25, 1986
[4] The First Level MWPC Trigger for the H1 Detector Internal Report H1-4/87-61, April 6, 1987
[5] Status Report of the First Level Proportional Chamber Trigger, Internal Report, September 15, 1987
[6] Design of the MWPC Receiver Card, Present Status, H1 MWPC Note No. 3 by H. Krehbiel, February 10, 1988
[7] LSI Data Book, Monolithic Memories, 7th edition 1986
[8] Hardware Triggers Using the Driftchambers of H1, Internal Report H1-11/87-74, by H.-J. Behrend, November 20, 1987
[9] Fairchild Advanced CMOS Technology (FACT) Logic Data Book, Fairchild, 1987
[10] Motorola Linear/Interface Integrated Circuits, Motorola, 1984/85
[11] Fairchild Advanced Schottky TTL (FAST) Data Book, Fairchild, 1985
[12] NEC Block Library for CMOS-4/CMOS-4A Gate Array 1.5 Micron Technology

Programs for Ray Finder Simulations as described in chapter 2 are available in Turbo Pascal, Version 3.0 (Borland) and run on any IBM PC. They expect a file of cross points of particle tracks with the MWPCs as input.

Complete schemes of the Ray Finder PCB prototype are available in many CAD file formats. (Plots require at least A1 paper size to be readable.)