Format of the RAM-file

J. Kuhn

December 14, 1996

This document describes the format of the file which is fed into the RAM.

1 Introduction

The RAM's address space is divided into 3 coordinates, called x, y and z. Each of these coordinates runs from channel #0 to channel #127 and is assigned a certain part of one multi-element detector in the experiment.

In the case for the E852 1997 run we plan to use 2 RAMs, which will have the following assignments:

RAM	*x-channel*	*y-channel*	*z-channel*
1	TPX2	TPX3	TPX1
2	TPX2	TPX3	$\overline{C9}$/H9 and H9

Table 1: Assignment of the detectors to the RAM's address space

In what follows the RAMs with TPX2, TPX3, TPX1 and TPX2, TPX3, $\overline{C9}$/H9 will be refered to as the TPX1-RAM and the H9-RAM, respectively. If a certain combination (x,y,z) in the RAM's address space is in agreement with an "allowed" combination for a track in the detector, that is if an allowed track of the particle in the detector can be connected to the detector-elements which are associated with these RAM-address, the flag with the address (x,y,z) is set from 0 to 1.

NOTE: Only the flags that are set to one are written to the RAM-file!

2 Detector-Assignment to the RAM-coordinates

The following assignment of the detecors is currently valid:

TPX1	wire #241 to 368	(ganged by 2)
TPX2	wire #217 to 472	(ganged by 2)
TPX3	wire #33 to 672	(ganged by 5)
$\overline{C}9$/H9 and H9	The 32 channels are going directly into the RAM	one-to-one translation

These channles are assigned to the RAM-addresses in the following way:

TPX2	*TPX3*	*TPX1*	$\overline{C}9/H9$	*H9*
0 to 127	0 to 127	0 to 63 (K^+)	-	-
		64 to 127 (K^-)	-	-
0 to 127	0 to 127	-	0 to 31 (K^+)	32 to 63 (K^+)
		-	64 to 95 (K^-)	96 to 127 (K^-)

3 x- and y-coordinates

The x- and y-coordinates that go into the RAM are translated one-to-one into hexadecimal format and written to file. There is no conversion of any kind.

4 z-coordinates

As the addresses in z are often clustered, that is several flags are set with the same x- and y-address, and with several consecutive z addresses, the z-addresses in the RAM-file are expressed in a format, that gathers together several z-addresses.

The z-address consist of 2 integer numbers:

The z-base address and
the z-datapattern

The translation is as follows:

The z-address, that is to be assigned is first divided by 4. The truncated result (fractional part is discarded) gives the z-base address. The z-datapattern is calculated from the remainder of this division:

remainder	*datapattern (binary)*	*datapattern (decimal)*
0	0001	1
1	0010	2
2	0100	4
3	1000	8

If there are several flags to be set with the same z-baseaddress, the logical or of the binary datapattern is taken to determine the datapattern that is written to the RAM-file. For example if 2 flags are set with the same baseaddress and one of them has remainder 0 and the other one has remainder 3 the datapattern is 1001 (binary) or 9 (decimal).

The result of this calculation is written in hexadecimal format to the RAM-file.

5 Example

Suppose the following addresses in the RAM's address space are to be set to one:

x-address	*y-address*	*z-address*
12	25	117
12	25	118
12	25	119
12	25	120
12	25	122
13	25	117
13	25	122

The x- and y-addresses do not change at all and are written directly to the file. For the z-address, calculate the z-base address first:

117 : 4 = 29 (+ remainder 1)
118 : 4 = 29 (+ remainder 2)
119 : 4 = 29 (+ remainder 3)
120 : 4 = 30 (+ remainder 0)
122 : 4 = 30 (+ remainder 2)

Therefore one has the following addresses right now:

x-address	*y-address*	*z-base address*
12	25	29
12	25	30
13	25	29
13	25	30

To calculate the datapattern we have to translate the remainder: For the address (12,25,29) we have the following remainders: 1 , 2 and 3. This gives the following z-datapattern 1110 (binary) or 14 (decimal). In the same way the datapattern for the (12,25,30) address (remainders are 0 and 2) is calculated to be 0101 (binary) or 5 (decimal). For the (13,25,z-base) addresses there is only one remainder each: 1 for the case z-base=29 and 2 for the case z-base=30. This gives the z-datapattern to be 2 and 4, respectively.

Therfore the complete addresses in the RAM-file read now:

x-address	*y-address*	*z-base address*	*z-datapattern*
c	19	1d	e
c	19	1e	5
d	19	1d	2
d	19	1e	4

Note that the values are printed in hexadecimal format.