ISTITUTO NAZIONALE DI FISICA NUCLEARE

Centro Nazionale
Analisi Fotogrammi

INFN/TC-82/4
6 Aprile 1982

M.L. Luvisetto and E. Ugolini:

AN ON-LINE DATA ACQUISITION SYSTEM FROM EXTERNAL DEVICES THROUGH A PDP 11 CHAIN USING A PDP 11 CPU FRONT END.

Servizio Documentazione
dei Laboratori Nazionali di Frascati

Istituto Nazionale di Fisica Nucleare
Centro Nazionale Analisi Fotogrammi

INFN/TC-82/4
6 Aprile 1982

M.L. Luvisetto and E. Ugolini: AN ON-LINE DATA ACQUISITION SYSTEM FROM EXTERNAL DEVICES THROUGH A PDP 11 CHAIN USING A PDP 11 CPU FRONT END.

1. - INTRODUCTION.

Our centre is equipped with various film measuring devices among which two HPDs. Furthermore the HPDs can be connected to a special hardware processor (SHP) designed to perform an on-line data reduction involving the HPD digitizings and acting between the measuring device and the data acquisition computer. Referring to the HPDs in the following parts of this report we will speak of digitizings as HPD output, while the SHP output will be referred to as segments.

This chain was connected until the beginning of 1981 to an IBM 360/44. As this computer had grown too old and unreliable to support the measuring devices, we decide to substitute the old computer with one of the new ones in our centre.

The computer section of our centre is equipped with a PDP 11-70 as general purpose computer, therefore to improve the speed in servicing interrupts, to avoid system overhead and to create an intermediate synchronization memory area, we decided to enlarge our PDP configuration by adding a front end computer to the PDP 11-70. In our case the front end computer is a PDP 11/34.

The connection between front end and main frame and between front end and on-line devices is realized through DIGITAL link hardware. The interface between the DIGITAL link and our devices has been realized in our centre.

2. - HARDWARE LAYOUT.

The hardware layout is shown in Fig. 1.

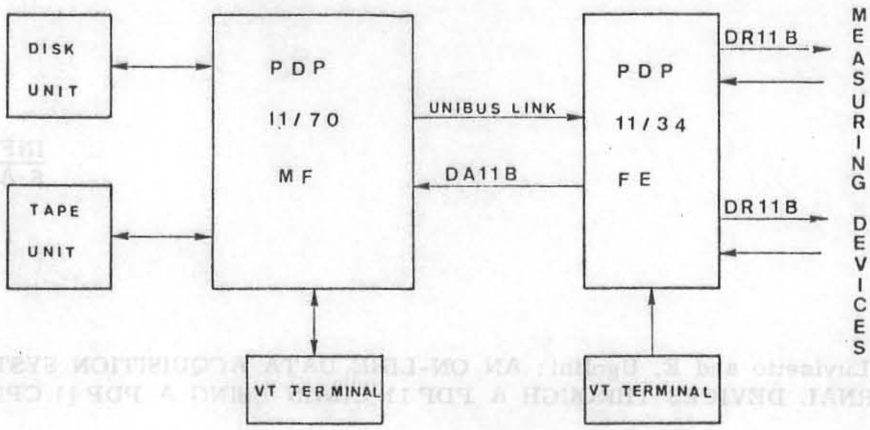


FIG. 1 - Computer Link.

The PDP 11/70 is equipped with the following peripherals :

- Disk Unit Model RP04,
- Tape Unit Model TU16,
- VT terminals model VT52, VT55, VT100,
- Unibus Link Model DA11B to connect Via Unibus both computer CPUs.

Furthermore the 11-70 has a total memory amount of 448 Kbytes and it works with RSX11M operating system.

The PDP 11-34 has no peripherals except a VT100 terminal. The memory amounts to 64 Kbytes plus floating point as special feature, the computer works without operating system. The bootstrap of the 11-34 is realized by means of a special ROM projected and realized at our centre.

The standard version of the built-in bootstrap ROM enables the user to initialize the computer from the operating system through the bootstrap switch on the console. In our case we have written a simple MACRO program that puts the computer in a WAIT state asking for data from the UNIBUS LINK, data that represent an executable image that begins running as soon as the end-of-information flag is set from the host computer. After testing this bootstrap program via software, the instructions have been writen on a ROM that has been plugged in substitution to the standard one.

The connection of the measuring devices to the PDP 11-34 is realized through a couple of DR11B. This DIGITAL feature is similar to the Unibus Link used between the two CPUs, in fact it acts in a certain way as a simplified DA11B. In Figs. 2a and 2b are shown the details of the on-line connections.

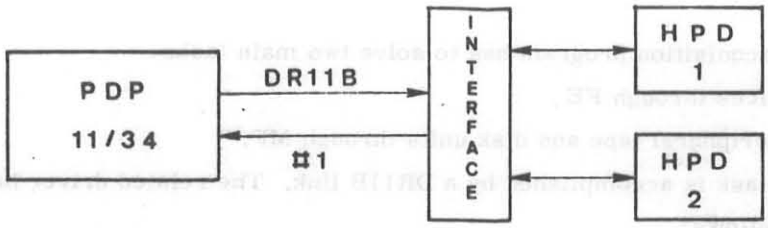


FIG. 2a - Computer HPD Link.

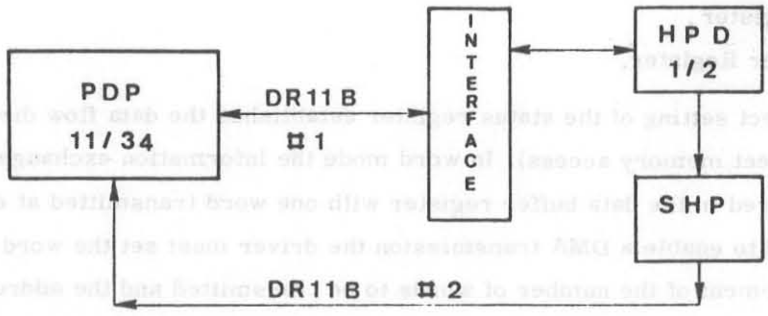


FIG. 2b - Computer - SHP Link.

Let us see in detail the meaning of the two schemes. First of all we must consider that in every situation there is a common data path used to send commands to the measuring devices. This data flow travels along the first DR11B and is used to select the device involved in the operation and to bootstrap it.

At this point a different data path takes place depending on the measuring technique required for each application.

If digitizing are required you have the situation shown in Fig. 2a in which the data from the devices travel on the same DR11B as the commands. If, on the other hand, segments are used as measuring results, the HPDs are bootstrapped through the first DR11B and the segments are sent from the SHP to the computer via the second DR11B.

On the computer side the HPD is seen as a peripheral device equipped with a 16 bit bus with bidirectional data flow either in word mode or in DMA. An interrupt is generated to signal the end of transmission, furthermore the program can access a status register both on the HPD and on the SHP.

The above hardware description is only introductory as the principal aim of this paper is the description of the software system realized at CNAF.

3. - SOFTWARE LAYOUT.

The data acquisition takes place through the link between PDP 11/70 and PDP 11/34 as shown in Fig. 1. To simplify flow-charts etc. from now on we will identify the PDP 11/70 as MF (main frame) and the PDP 11/34 as FE (front end).

Any data acquisition program has to solve two main tasks :

- pilot the devices through FE ,
- access the peripheral tape and disk units through MF.

The first task is accomplished by a DR11B link. The related driver has access to four registers as follows :

- Word Count Register ,
- Address Register ,
- Status Register ,
- Data Buffer Register.

The correct setting of the status register establishes the data flow direction with or without DMA (direct memory access). In word mode the information exchanged between FE and devices is stored in the data buffer register with one word transmitted at each operation. On the other hand to enable a DMA transmission the driver must set the word count register with the 2's complement of the number of words to be transmitted and the address register must be loaded with the starting location. As soon as a bit in the status register is set to 1, the DMA operation takes place and it stops only when an interrupt is generated. This can happen either when the word count register is reduced to zero or when the device generates an end-of-transmission interrupt. Furthermore the status register has a set of 3 read/ write bits plus 3 read only bits at user's disposal. By clearing, setting and reading these bits the user is enabled to exchange information between FE and devices.

Operating in word mode the wanted external device is selected, checked for operational status and bootstrapped on the wanted event on film. At this stage the device is ready to start measuring and the program operates the link in DMA with a flip buffer technique at each interrupt service. When an interrupt takes place, an user dependent interrupt routine checks the situation to distinguish between word count zero and end-of-information interrupts forcing the required action.

With regard to the second task, the normal mode of operation of any data acquisition program is based on a synchronized operation of FE and MF. In practice we have two separate programs running on both computers and a protocol to exchange information. The data acquisition takes place on an event basis with the event information stored on to disk. To simplify the operator's intervention all information exchange between program and operator is visualized on the FE machine, therefore any data acquisition program can be standardized as described in the general flow-chart shown in Figs. 3a and 3b.

To achieve the synchronization between the two CPUs, the machine in wait state asks for a status area from the other one. As soon as this status area has been filled the waiting machine wakes and takes action as required by the status flags in the status area while the other machine changes its status into wait. The status area contains the whole information to synchronize MF and FE and is used to transmit error flags, status words, device constants, frame advance, error codes, etc.

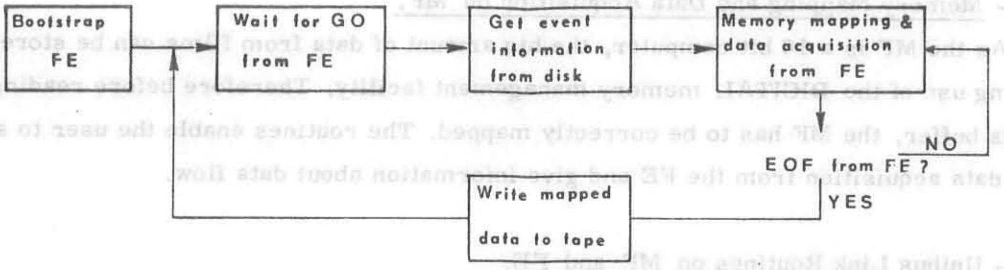


FIG. 3a - Flow diagram on MF.

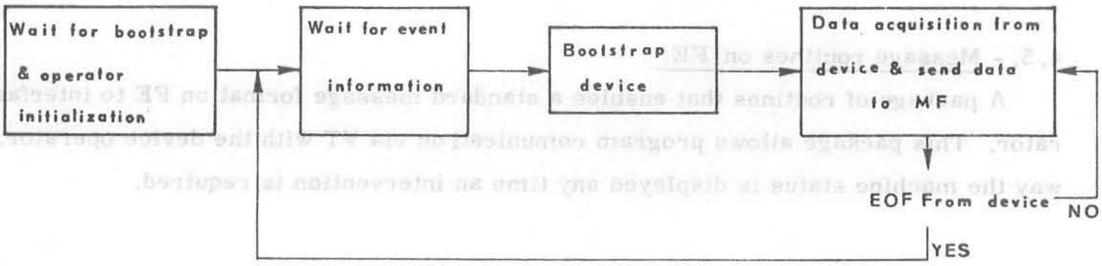


FIG. 3b - Flow diagram on FE.

The main idea is to have a certain number of packages that solve all general purpose problems so that any application program can easily be built up by making use of the general purpose library plus a very small number of problem dependent routines.

Furthermore, as the FE works without operating system the general purpose library includes drivers for VT and Unibus Link plus interrupt and trap handlers.

4.- GENERAL PURPOSE LIBRARY.

4.1. - Data Acquisition on FE.

Routines to read data from HPD or SHP flipping buffers. Each buffer is 2048 16 bit words long. The routines detect the device interrupt and any malfunction during the transmission from device to FE. The routines provide also to send the just read buffer to the MF.

4.2. - Device Bootstrap from FE.

A package of routines to set the device on the current event, to initialize the measuring coordinates, to check the execution of the above operations and to inform the operator of any malfunction, to read the value of the device status word. If segments are required, the package enables the user to set the SHP parameters, to set the SHP in a measuring state and to activate the FSD as input device toward the SHP.

4.3. - Memory Mapping and Data Acquisition on MF.

As the MF is a 16 bit computer, the big amount of data from films can be stored only making use of the DIGITAL memory management facility. Therefore before reading a 2048 words buffer, the MF has to be correctly mapped. The routines enable the user to synchronize data acquisition from the FE and give information about data flow.

4.4. - Unibus Link Routines on MF and FE.

A package of routines that enables the user to regard each computer as a peripheral connected to the other one, on which I/0 operations as read and write can be accomplished.

4.5. - Message routines on FE.

A package of routines that enables a standard message format on FE to interface the operator. This package allows program comunication via VT with the device operator. In this way the machine status is displayed any time an intervention is required.

5. - APPLICATION PROGRAMS.

As we have pointed out describing the software layout, the main criteria are to keep programs as standard as possible and to have any variable item stored on to files that do not affect the program if changes are required.

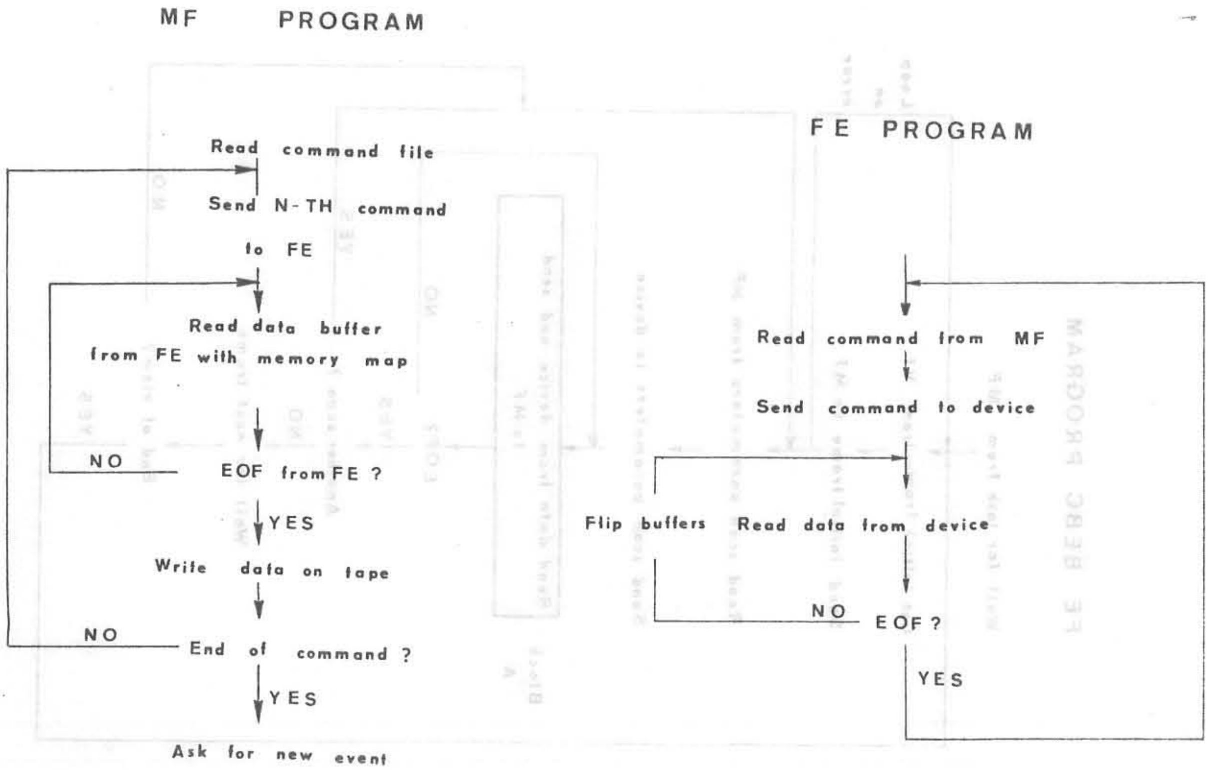
Regarding application programs we can consider two main chains :

- general data acquisition without processing,
- specific data acquisition.

To the first case belongs a very general program that is able to write on tape digitizings or segments or both in any sequence depending on a command file residing on MF disk. In this way using a single program chain for MF and FE any measuring sequence can be realized by writing an appropriate command file. The program chain is shown in Fig. 4 (Fig. 4a for MF program and Fig. 4b for FE program).

To manage the second case we have written different programs for various cases. At present we are analyzing events from BEBC and from streamer chambers. The general layout is much the same for any chamber and the main difference regards binary number decoding routines and scan sequence. We can regard BEBC as the more general case as four different film scans are required to collect data from one event in one view. The flow charts of MF and FE programs are shown in Figs. 5a and 5b.

In block A of either program can be included any data analysis routine, in our case there is a binary number decoding routine that checks during the appropriate scan that the digitized frame is the correct one and if not, tries recovery if possible or asks for operator intervention if every automatic recovery try failed.



a b

FIG. 4

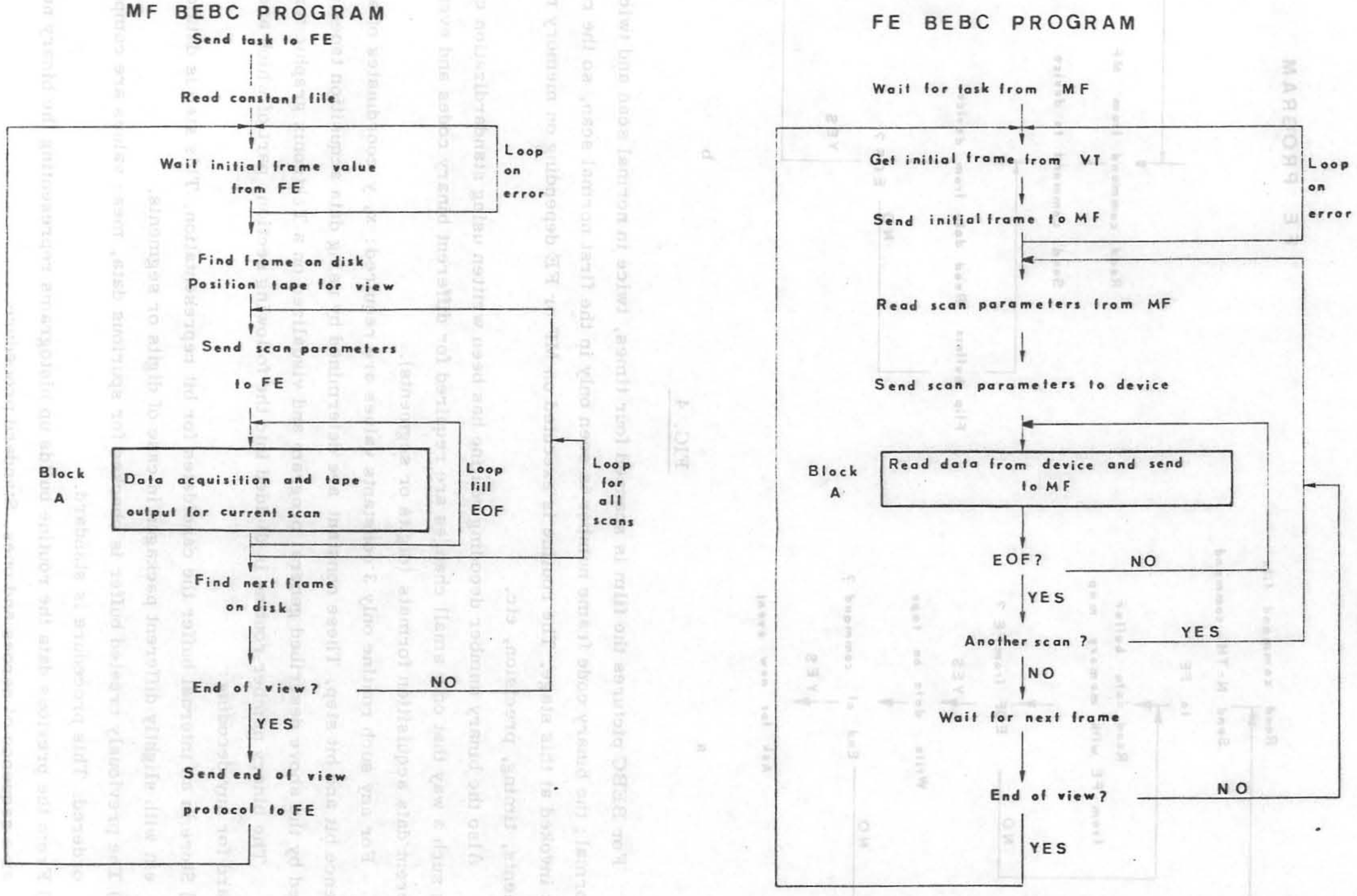
For BEBC pictures the film is scanned four times, twice in normal scan and twice in abnormal; the binary code frame number is seen only in the first normal scan, so the routine is invoked at this stage. The routine is executed on MF or FE depending on memory requirements, timing, precision, etc.

Also the binary number decoding routine has been written using standardization criteria in such a way that only small changes are required for different binary codes and even for different data acquisition formats (digits or segments).

For any such routine only 3 constants values are required: x, y coordinates of a reference bit and bit step. These constant are determined by using data acqusition tapes produced by the above described general program and visualized on a Tektronix graphic terminal.

The binary number routine is divided into the following section, part of which are standard for any decoding:

a) Store in an internal buffer the candidates for bit representation. This step is data dependent with slightly different packages in case of digits or segments.
b) The previously created buffer is checked for spurious data, mean values are computed and ordered. This procedure is standard.
c) From the previous data the routine builds up histograms representing the binary number as a sequence of zeroes and ones. Standard procedure.


MF BEBC PROGRAM
Send task to FE
Read constant file
Wait initial frame value from FE
Loop on error
Find frame on disk
Position tape for view
Send scan parameters to FE
Block A
Data acquisition and tape output for current scan
Loop till EOF
Loop for all scans
Find next frame on disk
End of view ?
NO
YES
Send end of view protocol to FE
FE BEBC PROGRAM
Wait for task from MF
Get initial frame from VT
Loop on error
Send initial frame to MF
Read scan parameters from MF
Send scan parameters to device
Block A
Read data from device and send to MF
EOF?
NO
YES
Another scan ?
YES
NO
Wait for next frame
End of view?
NO
YES

a b

FIG. 5

d) From the histogram configuration the decimal number is computed according to the film dependent binary code used.

e) According to the redundancy check used in the binary representation, the decoded number is checked against the required frame number and action is taken depending on the result.

The efficiency of the binary number routines ranges from 90 to 100 %. The failure reasons depend strictly on film quality and therefore are different from one experiment to another. The most common rejects are due to critical scratches or spots obscuring the binary bits on film or to bad quality brenner marks that cause a wrong film positioning in the measuring device.

MF programs are enterely written in FORTRAN. The length depends widely on the program itself and ranges from nearly 10 k 16 bit words for the general acquisition one to 25--30 K words for the more complex ones (i. e. BEBC).

FE programs are enterely written in MACRO except for the binary bit decoding routine (when included in the FE task) with typical lengths of 13 K words.

FE programs reserve an area of four 2048 16 bit word buffers for data acquisition structured as circular buffers.

MF programs map the memory over 2048 word windows, the amount of mapped memory depend strictly on the application program and ranges from 65 windows for the general program to 21 windows for BEBC. This size is tuned sperimentally according to the amount of data generated by each specific measurement.

Even if the amount of storage needed for each job limits the number of simultaneously running tasks, the MF machine can work in a multitask environment without affecting the data acquisition program and an average timing is nearly 9" per scan, therefore BEBC requires 36-37" per event in one view.

The previous on-line computer caused some problems with BEBC films when an event had too many tracks as the machine was not fast enough to store the whole information measured by the device. With the new set-up, taking into account that the MF machine is nearly 4 times faster than the old one, it has a bigger amount of memory and a dedicated front end, data loss problem has been solved and the overall speed has been improved by a factor of 1. 35. Tape handling is also more efficient as the new FM machine supports 1600 bpi tape units, as a consequence the amount of events stored per tape has nearly doubled.

The new chain has been throughly tested by measuring a sample of nearly 5000 BEBC events. Checks have been done along the whole chain with comparison of results at geometry level that lead to satisfactory agreement.