DEUTSCHES ELEKTRONEN-SYNCHROTRON DESY

DESY DV-72/1
October 1972

Use of an ARGUS 500 Computer to Support the Operation of a Linear Accelerator

by

K. Dahlmann, J. Ehrig, O. Hell

2 HAMBURG 52 · NOTKESTIEG 1

To be sure that your preprints are promptly included in the HIGH ENERGY PHYSICS INDEX, send them to the following address (if possible by air mail):

DESY
Bibliothek
2 Hamburg 52
Notkestieg 1
Germany

Use of an ARGUS 500 Computer to Support the Operation of a Linear Accelerator

by

K. Dahlmann, J. Ehrig, O. Hell

1. Introduction

In 1969 a new linear accelerator (LINAC) was installed at DESY. This machine injects an electron or a positron beam into the synchrotron. To facilitate the control of the LINAC, DESY bought the computer ARGUS 500.

The computer is needed for setting values, checking status and data and for measuring and manipulating data to be displayed on a screen or printed out on a teletype. The reader who is interested in an overview may restrict himself to Chapter 2 which gives a general view of the processes and the computer hardware, furthermore Chapters 4.1.1 and 5 which give an introduction into the operating system used and discuss the results. Chapter 3 describes the man-machine-interface which represents the link between the accelerator and the computer. Chapter 4 contains the discription of the operating system and the application programs.

2. Hardware

2.1 Processes Involved

The processes involved can be divided into the actual accelerator with its 12 sections which was built by VARIAN and the beam transport system which was installed by DESY. Figure 1 gives a schematic view.

The LINAC is an electron/positron accelerator respectively. It injects particles into the synchrotron and in the near future into the new storage ring.

With the beam transport system it is possible to lead the particles to five different places. This optical system is composed of bending magnets, quadrupole lenses, and steering coils. The transport system was planned so that its elements can be operated both manually and by the process computer.

VARIAN installed the accelerator irrespective of computer control considerations. Thus, DESY had to build the interface hardware to the LINAC.

In the future DESY will replace the most important power supplies by ones which can be computer-controlled.

2.2 The Computer

The control computer is an ARGUS 500 (Fig.2) delivered by FERRANTI Ltd. The equipment consists of:

Core storage:	32 K words
Word length:	24 Bit
Cycle time:	2 microseconds
Accumulators:	8 accumulators register sets with 8 accumulator registers each. Three accumulators of each set can be used as index registers.
Interrupt system:	8 interrupt lines (with one priority level).

Peripherals:

1 disc (fixed head system)
capacity: 640 000 24-bit words
access time (average): 17,3 milliseconds
1 graphic display with keyborad, joystick and 4 K buffer store.
1 paper tape reader, 300 characters/sec.
1 paper tape punch, 150 characters/sec.
2 teletypes.
240 digital outputs (relais)
80 digital outputs (semiconductor)
160 digital inputs (semiconductor)
384 digital inputs (relais)
256 analog inputs
16 analog outputs.

Special features of the computer are:

(I) direct addressing of input-output signals,

(II) direct store access for high speed information transfers in bulk to and from the computer core store,

(III) busy lines for peripheral controllers,

(IV) The I/O lock-out control circuit in the I/O equipment together with the accumulator relativiser facility enables program testing and development to be done while the computer is controlling a process, without risk of interference to the on-line control program.

(V) Several special registers are protected against change by user programs and can only be modified when the computer is in supervisor state.

3. The Man-Machine-Interface

Both accelerator and operator receive and send information. It is the task of the computer to process and to transmit this information. This requires an interface between the accelerator and computer, and on the other hand an interface between the computer and operator. The interface between computer and accelerator consists of hardware elements as analog modules and the driver programs for these elements in the operating system. The man-machine interface is a bit more complicated. The control purposes are well defined, but the imple-

mentation should come as near to the requirements of the operators as possible even if the programming becomes a tedious task. Computer-operator communication could be achieved by the use of a keyboard, or special purpose buttons [3]. Special purpose buttons are trivially easy and fast to use but have disadvantages if a new function (program) is to be implemented or if a program should require parameterised information.

Designing the control system, we felt, that the man-machine interface should meet the following requirements:

1. The operator should be able to supply parameters such as magnet numbers to the program to be executed.
2. It should be possible to execute a command at a specified time and possibly to repeat it after a specified time-interval.
3. It should be easy, to add new commands, since the need of new applications cannot always be foreseen.

It is very unlikely, that push buttons can meet these requirements. Therefore we designed and implemented a language, the Linac control Language (LILA) [2], as a tool of machine-operator communication. On the other hand, we felt, that in simple cases using the language should be nearly as handy as using push-buttons.

This leads to the following very simple syntax:

Label: Operator Operand Clause(s)

Of these elements only the "Operator" is necessary. It may be a verb or a program name. In the latter case any "operand" and/or "clauses" are ignored. If the "operator" is a verb, only the first three letters are significant, i.e. one can write PRI instead of PRINT. "Operand" and "clause" may be exchanged. The different parts of a command are seperated by at least one blank.

The label consists of up to four letters followed by a colon as delimiter.

The "operand" may be a list of names of elements of the LINAC or names of groups of such elements. The command

DIS S1, M1

simply means: display the currents of the magnet S1 and of the magnet M1. The command has a list of names of LINAC elements as "operands" whereas the operand

DIS S

specifies a group of elements: the currents of all S-magnets are to be displayed.

There are two kinds of clauses, the destination clause and the time clause, which allows repetition too. The time clause specifies time, time interval and number of executions. The destination clause specifies the destination of the result; e.g. to put the result on one of the two teletypes. It is allowed to combine both clauses, e.g.:

A: PRINT VV AT 20 H EVERY 10H
B: PRI EVERY 1H UNTIL 20H UPON TELB VV,M

The maximum length of a command is 125 characters, and it is terminated by a special character.

The full set of operators and clauses may be found in the appendix.

4. Software

4.1 The Operating System

4.1.1 Introduction

There are two sources of requests to the computer:

— the operator issuing a command and
— the accelerator generating an interrupt, requiring a certain task to be performed.

The different tasks are of varying importance. Hence an operating system with multiprogramming and priority scheduling was required. When we started in 1968, FERRANTI's Small Maschine Organizer (SMO) [1] was the only system that met these requirements and worked. Thus SMO became the heart of the executive part of the present system.

The SMO together with three additions form the basic operating system called LINAC Machine Organizer (LMO). These additions are

- a display support package

- a table of data and addresses providing communication and linkages between different parts of the system (transfer vector)
- a set of reentrant coded utility routines including generalized space control routines.

The LMO is supported by a command language interpreter program.

4.1.2 The Executive Part of the System

The nucleus of the LMO is a very compact program residing in the protected area of core. It supports all interrupt lines as well as program interrupts. It allows for seven independent tasks and organizes input from, and output to slow peripherals and the disc. It also recognizes operator commands and activates the interpreter.

The executive part of the system consists of an interrupt handler and a number of interrupt routines. It has two entrypoints, one for hardware interrupts and one for program interrupts. Some of the interrupt routines must run under interrupt lock out, such as serving the timer or accepting a character from a keyboard. Other routines are themselves interuptable, e.g. resource administration, process program management, and most display services. Each interrupt routine can schedule other interrupt routines, as necessary.

The process management routines serve to overcome the limitation of only seven independent tasks (a direct result of having seven different sets of accumulators, plus one for the system). Each process program is assigned to one of the seven priority levels. Within each level the scheduled tasks are handled serially.

4.1.3 Display Support Package

This package consists of a number of interrupt routines.
Our FERRANTI display with keyboard is to serve three purposes:

1. graphic displays generated by process programs ('pictures'),
2. display of alarm messages,
3. as a terminal (with fast output) for communication between the operator and the computer, i.e. like a teletype.

While the keyboard is devoted solely to the third purpose, the screen has to serve all three. Since it was wished that console output should be independent of other displays, while pictures and alarms may be mutually exclusive, the screen was logically divided into two sections:
An upper larger section for pictures and alarms, and a lower section of 9 lines of text for the teletype-like application.

The lower section also continuously displays the date and the time of day.

The display buffer contains separate regions for the pictures and the alarms. Thus an alarm can overlay the picture on the screen without touching the picture's buffer store (which, incidentally, is controlled dynamically by the bitstring routines mentioned earlier). After the operator has acknowledged the alarm, the picture then reappears. The process program that generates the picture is not aware of this interuption.

4.1.4 Reentrant Routines

A number of such routines have been implemented, following ideas expressed in [4]. In the present version of the system they all reside in core. Typical ones are routines to facilitate output: packing and unpacking character strings, output of character strings by device name (and independent of device characteristics).

Since reentrant routines may call one another, certain conventions for stacking register contents have to be obeyed. We decided on a stack provided by the calling process program, the address of the stack being passed in a standard register. The stacking routines are themselves reentrant.

The reentrant routines may be assembled and loaded independently of the process programs. The linkage of the process programs to these routines is done via the transfer vector which starts on a fixed location in core. At assembly time symbolic names are assigned to the elements of this vector.

4.1.5 Dynamic Space Control

All kinds of space allocation as allocation of core store for parameter lists or buffer space for display pictures is done via interuptable bitstrings routines [5]. The request for allocation causes a search, of the relevant bitstring, for a specified number of adjacent zeroes, which are then set to ones. The relative address within the bitstring are returned. De-allocation requests are met, by equivalent clearings of adjacent bits, at a specified relative address within a bitstring. The significance of the bits in these bitstrings is defined within the calling routines, e.g. blocks of core, or backing store.

4.1.6 The LINAC Language Interpreter

The LINAC Language Interpreter, called LILI, runs as a task with highest priority and is to interpret the commands entered by the operator. The analysis of such a command does not in every case result in the activation of a process program. Several request, e.g. display of active commands, are fulfilled by LILI itself. But in general one or more process programs will have to be activated. In this case the interpreter will determine the programs to be activated and their parameter lists, using the data in the command and a decision table. It will also do some validity tests and check the syntax of the command. The storage for the parameter lists is controlled dynamically, using the bitstring routines described earlier. If a time clause exists the command will be stored (the necessary store also being controlled dynamically).

After complete interpretation of a command LMO is called to activate the process programs requested previously.

4.2 Process Control Software

4.2.1 Magnet Current Setting Program

After starting up the LINAC the operator manually directs the beam into the measurement room by the bending magnets M1 and M11 (see Fig. 1). When the beam has the proper energy and intensity, the operator requests this program which reads the shunt voltage of the magnet M1 from which the energy is calculated.

The shunt voltage is measured by a digital voltmeter and the computer reads the outputs of this device via fast digital inputs. The currents of the different quadrupoles are proportional to this energy by different factors. The program calculates the currents of all the elements of the switched-on beam path - in this case beam path measurement room - and sets up the appropriate power supplies.

If the beam is to be changed from one path to another (e.g. to the synchrotron) the operator calls the Energy program described in 4.2.2. Then the LINAC operator turns on the desired beam path and reactivates the Setting program. Now this program uses the energy value produced by the Energy program to calculate the currents of the bending magnets and quadrupole lenses of the new beam path, proportional to the energy measured, and sets up the power supplies.

Similarly the LINAC operator can switch on and off each beam path.

4.2.2 Measuring of the Beam Energy

This program reads the shunt voltage of the bending magnets M1 or M2 depending on the beam path switched on, calculates the energy of the particles and displays the value to the operator. Besides that the energy value is stored in the table of data and addresses where it can be read by the Setting program.

4.2.3 LINAC In-Run Machine Report

This program prints, on request, an in-run machine report on the teletype. The report contains the parameters of the accelerator and the beam transport system.

The program has a dialog part where the operator must respond to certain questions. As there are only slow teletypes for print-outs, the operator can direct the report to the display, in order to survey quickly certain parameters.

In this case the program shows enough lines of the report to fill the whole screen and then waits for operator action. Now the program can accept commands from the keyboard and it is possible to send the next line or next "page" of the report to the CRT. One page is defined as 33 lines which is a full screen.

This procedure may be terminated by the operator before the full report is given. Furthermore the contents of the screen may be onbtained in hard-copy form, on the teletype.

4.2.4 Beam Position Monitor Program

At the present time the beam position monitors are not yet completely available. Therefore this program still cannot be used to its full extent.

The different beam positions and the beam intensity will be displayed in diagrams on the CRT. The diagrams also contain the elements of the appropriate beam path as symbols. This program interacts with the LINAC operator. It is possible to change the scaling of each diagram.

If the beam leaves the center of the tube then the operator can tune the current of each steering coil or magnet with the joystick. The joystick is serving two purposes:

Firstly, in order to address an element, the operator moves a cursor over the screen to the element required and presses the appropriate key. After addressing the element, the current appears on the CRT.

Secondly, to change this current, the operator presses a special key and then, the rate of change of the current is proportional to the inclination of the joystick.

4.2.5 Checking of the Vacuum

A routine in the executive part of the LMO reads the pressure of one of the vacuum pumps and stores the value in a table every 20 milliseconds. The vacuum checking program tests the pressure of the vacuum system and gives a message on the console, if the value of a pump is higher than a certain threshold. A pump with bad vacuum will be flagged in order to avoid repeated generation of the message. If the pressure comes down below the limit the program clears the flag.

4.2.6 Vacuum Report

In addition to the LINAC in-run report program this program is specially used by the vacuum operators to track the pressure of the vacuum system after a shut down period. The operator can address each pump, or group of pumps, and get the pressure on the teletype or the display.

5. Discussion

5.1 Extended Applications

The ARGUS 500 is now not fully loaded. The reasons for not loading it further seem to be:

1. Hardware to perform setting ('Stellen') functions is very expensive.
2. The available manpower is absorbed by other, more urgent projects.

But both reasons need not the prohibitive forever.

5.2 Reliability

During the first two years we had much trouble with the computer. There were 35 major hardware faults. However, after the burning out of a main power supply the computer ran satisfactorily. The following data are taken from the interval May '71 to May '72.

We defined two availabilities.

1. Availability for the programmers =

$$\frac{\text{hours of the year - hours of breakdown}}{\text{hours of the year}} = 99.8\ \%$$

and

2. Availability for the process =

$$\frac{\text{hours of the year - hours of breakdown - hours of test}}{\text{hours of the year}} = 84.7\ \%$$

Another important expression is the mean time between failures (MTBF).

$$MTBF = \frac{\text{hours of the year - hours of breakdown}}{\text{breakdowns}} = 8640$$

The software system has broken down 5 times during the above mentioned time interval, for reasons which could not be identified.

5.3 The Operator's Language

The LINAC Language, as far as it is implemented, seems to fulfil all demands. Nevertheless it might be useful to be able to invoke a much used but complicated command with a short command. This could be accomplished by implementing the command STORE. STORE should store a command on disk under some name to that it could be called and issued using that name.

We feel that in applications like the present one the problem of how to facilitate a simple, convenient communication between operator and process cannot be overestimated. It is the vital question when the system is judged from the operator's (usually not a computer expert) point of view.

ACKNOWLEDGEMENTS

In the beginning of the project Mr. S. Pätzold also belonged to the team. He later left this project to work on the storage ring computer project.
The authors thank Mr. A. Warman for his advice concerning the English language. We are also indebted to Miss Tödt for her sterling effort in reading and typing the manuscript.

APPENDIX

The LINAC Language LILA

The definitions in the following part are thought to be a programmer's tool to design the interpreter and not to tell the operator of the LINAC how to construct a command whereas the additional rules are important for both the programmer and the operator. As the language is not a programming language the operator has to choose the names and verbs out of a given set except labels, which can be constructed as the operator wants.

A.1 Syntax

To describe the syntax of the language the following characters are used:

|	or
||	concatenation
(...)	the syntactical unit within the parenthesis is optional
::=	defined as
<...>	the syntactical unit within the brackets is to be replaced by a characterstring.

Parenthesis and brackets serve only to describe the syntax, they do not appear in an actual command. Signs, comma, colon and periods are to be written as shown. Keywords are indicated by capital letters.

name	::= <1 to 4 letters>
operator	::= <verb>|<name of a program>
verb	::= <a number of letters, the first three only, being significant>
keyword	::= <a number of letters, the first three only, being significant>
groupname	::= <1 to 3 letters>
element	::= <groupname || 1 to 3 figures>
list of elements	::= <element>|<element><,list of elements>
list of groups	::= <groupname>| <groupname><,list of groups>

```
value                ::= <1 to 4 figures>|
                         <0 to 4 figures><.0 to 4 figures>|
                         <value><A>|<+value>|<-value>
list of values       ::= <element><=value>|
                         <list of values><,list of values>
operand              ::= <list of elements>|
                         <list of groups>|
                         <list of values>|
                         <operand><,operand>
time clause          ::=  AT <t>|
                          (AT <t>) EVERY <dt> (UNITIL <t1>) |
                          (AT <t>) EVERY <dt> (<i> TIMES)
t, t1, dt            ::= <hh>(<:mm>(<:ss>))H|
                         <mm>(<:ss>)M|
                         <ss>S
hh                   ::= <integer numbers from 1 to 24>
ss, mm               ::= <integer numbers from 1 to 60>
i, i1, i2            ::= <integer number  without sign>
destination clause   ::=  UPON <name of device>|
command              ::= <name of program>|
                         <verb> <operand>|
                         <verb> <operand> <clauses>|
                         <verb> <clauses> <operand>
labeled command      ::= <name:> <command>
```

Additional rules are:

1. The different parts of the command have to be seperated by at least one blank.
2. Carriage return and line feed do not replace a blank.
3. A blank is optional immediately following a comma, in front of and immediately following the equal-sign and the single letter A (standing for Ampere) used solely in connection with values.
4. A blank may not be placed within names, elements or keywords, within a time or in front of a comma.
5. Only the first three letters of verbs and keywords are significant, all following letters are optional and are ignored by the interpreter.
6. All commands with a time clause should have a label, if it is missing, one is generated by the interpreter.

A.2 Language Elements

Keywords are the following words:
AT EVERY UNTIL TIMES UPON TO FROM ON OFF LABELS IS CORE

The keywords are the tool to recognize a clause or certain operands.

The following list shows all possible commands, clauses in parenthesis are optional.

ANALOG	<element> TO <i>
CANCEL	<name of a label>
CHECK	<operand> (<time clause>)
CLEAR	
DIGITAL	<element> TO <i>
DISPLAY	<operand> (<time clause>) | <label> | LABELS
DUMP	i1 - i2 | CORE (UPON <device>)
LOAD	<name of a program>
PRINT	<operand> (<time clause>) (UPON <device>)| <label> | LABELS
PUNCH	<operand>
SET	<list of values>| IS ON|OFF
TIME	
STORE	<name>
EXECUTE	<name>
<name of program>	

Restrictions

Some commands are not implemented at the moment (July 1972). They are:
ANALOG DIGITAL DUMP LOAD TIME STORE EXECUTE

REFERENCES

1.) Ferranti; Software Information Bulletin No.19
B. Gorman; Standard Small Machine Organiser, Manchester 1968

2.) S. K. Hawry et al.; The SLAC Beam Switchyard Control Computer
IEEE Transactions on Nuclear Science INS-14, No.3, June 1967 P.1066ff.

3.) H. van der Beken, W. Remmer;
The Process Control Computer of the CPS, MPS/CO Computer 69-2, May 1969.

4.) Ferranti Ltd, Technical Memorandum; B. Gorman; Reentrant Programming Techniques for the ARGUS 500, Manchester, 1968.

5.) PDP-9, Advanced Software System Monitors Digital Equipment Corporation, Maynard, Mass. 1968.

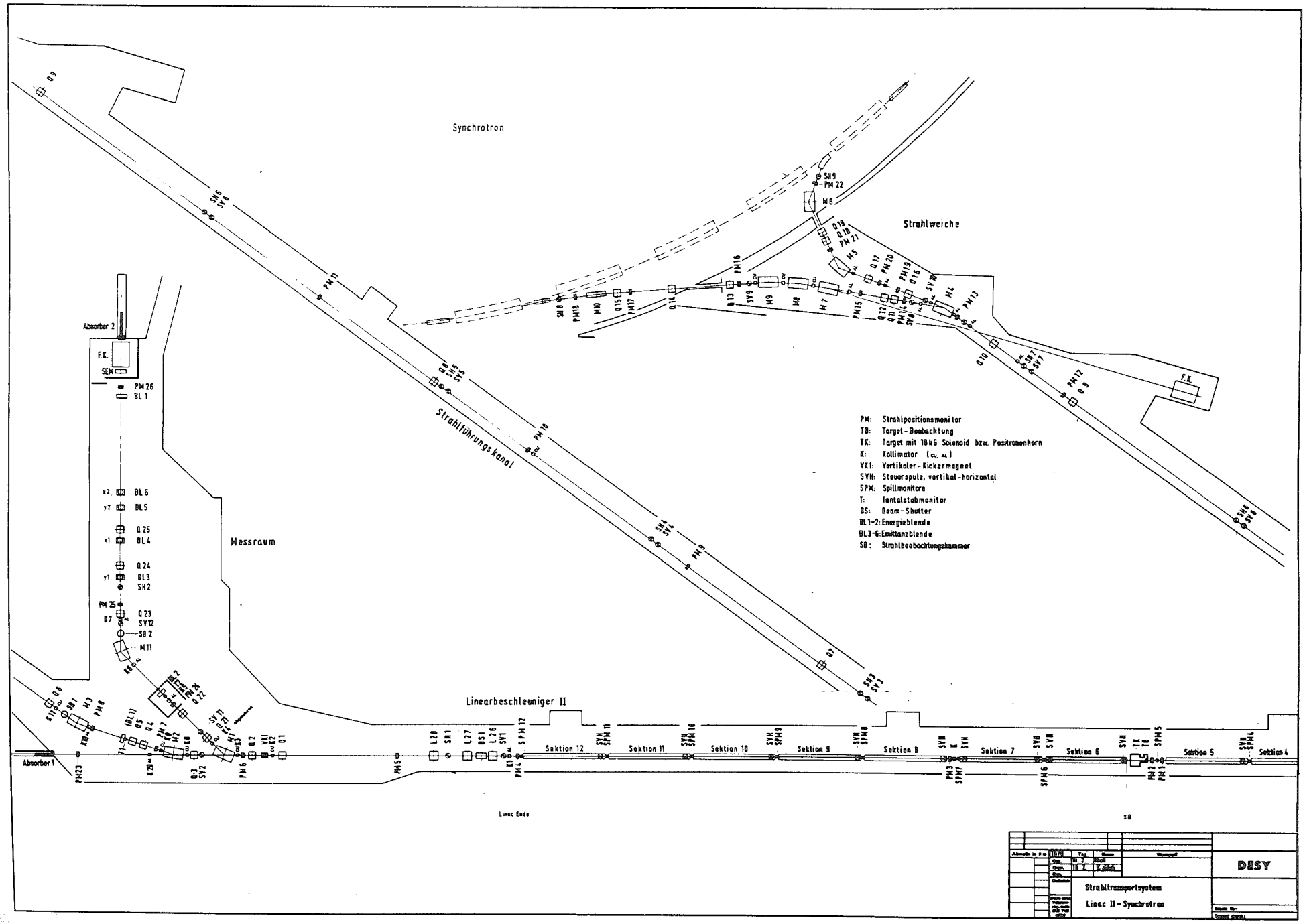
Synchrotron
Strahlweiche
Strahlführungskanal
Messraum
Linearbeschleuniger II
Absorber 2
Absorber 1
F.K.
SEM
PM: Strahlpositionsmonitor
TB: Target-Beobachtung
TK: Target mit 18kG Solenoid bzw. Positronenhorn
K: Kollimator (Cu, Al)
YK1: Vertikaler-Kickermagnet
SVH: Steuerspule, vertikal-horizontal
SPM: Spillmonitore
T: Tantalstabmonitor
BS: Beam-Shutter
BL1-2: Energieblende
BL3-6: Emittanzblende
SB: Strahlbeobachtungskammer
Sektion 12
Sektion 11
Sektion 10
Sektion 9
Sektion 8
Sektion 7
Sektion 6
Sektion 5
Sektion 4
DESY
Strahltransportsystem
Linac II - Synchrotron

FIGURE 1

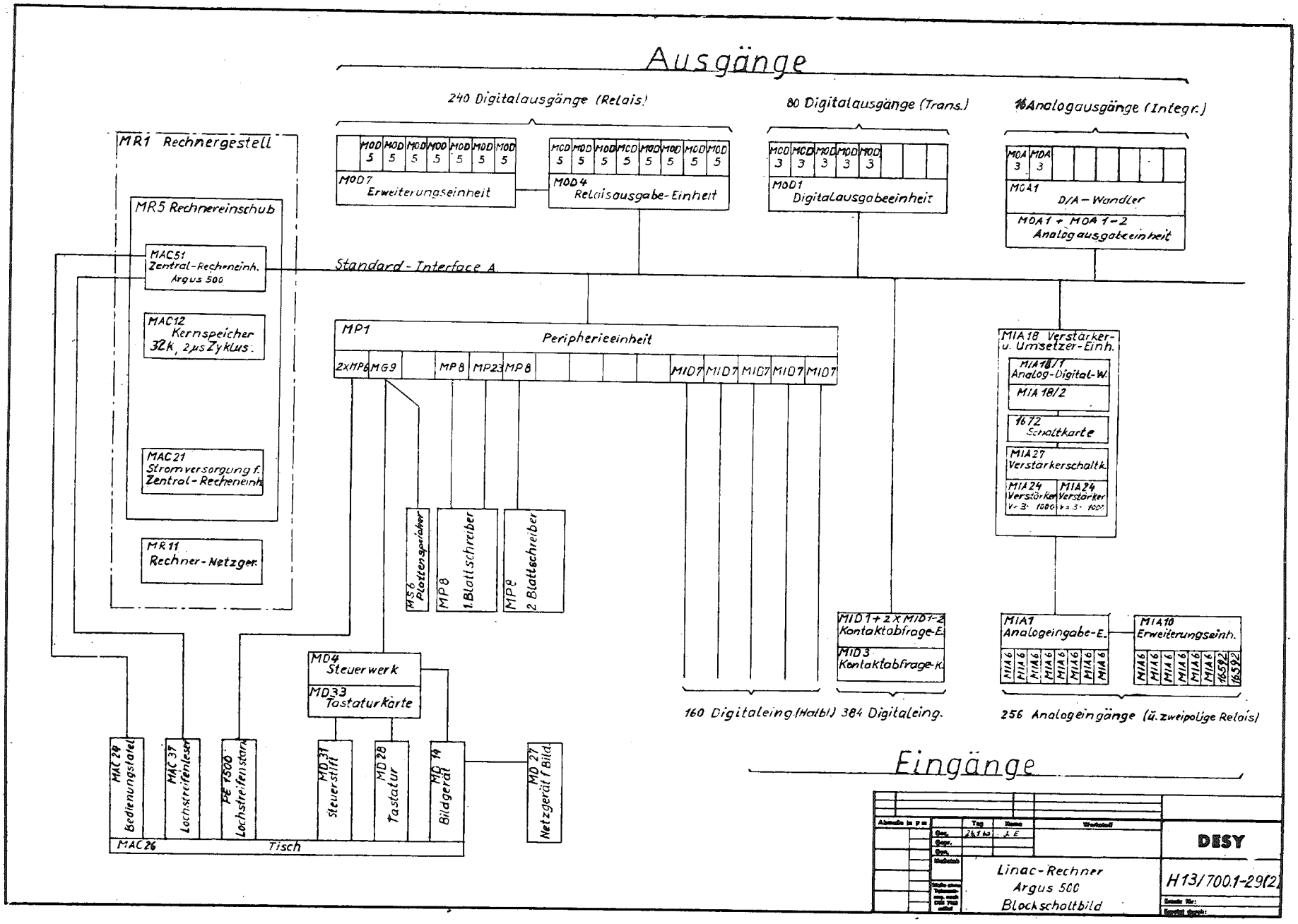
Ausgänge
240 Digitalausgänge (Relais)
80 Digitalausgänge (Trans.)
16 Analogausgänge (Integr.)
MR1 Rechnergestell
MR5 Rechnereinschub
MAC51 Zentral-Rechneinh. Argus 500
MAC12 Kernspeicher 32K, 2µs Zyklus
MAC21 Stromversorgung f. Zentral-Recheneinh.
MR11 Rechner-Netzger.
MOD7 Erweiterungseinheit
MOD4 Relaisausgabe-Einheit
MOD1 Digitalausgabeeinheit
MOA1 D/A-Wandler
MOA1 + MOA1-2 Analogausgabeeinheit
Standard-Interface A
MP1 Peripherieeinheit
2xMP6 MG9 MP8 MP23 MP8 MID7 MID7 MID7 MID7 MID7
MP6 Plottenspeicher
MP8 1. Blattschreiber
MP8 2. Blattschreiber
MIA18 Verstärker- u. Umsetzer-Einh.
MIA18/1 Analog-Digital-W.
MIA18/2
1672 Schaltkarte
MIA27 Verstärkerschaltk.
MIA24 Verstärker V=3-1000
MIA24 Verstärker V=3-1000
MD4 Steuerwerk
MD33 Tastaturkarte
MID1+2xMID1-2 Kontaktabfrage-E.
MID3 Kontaktabfrage-K.
MIA1 Analogeingabe-E.
MIA10 Erweiterungseinh.
MIA6 16592
MAC28 Bedienungstafel
MAC37 Lochstreifenleser
PE 1508 Lochstreifenstanz
MD31 Steuerstift
MD28 Tastatur
MD14 Bildgerät
MD27 Netzgerät f. Bild.
MAC26 Tisch
160 Digitaleing. (Halbl.) 384 Digitaleing.
256 Analogeingänge (ü. zweipolige Relais)
Eingänge
DESY
Linac-Rechner Argus 500 Blockschaltbild
H13/700.1-29(2)

FIGURE 2