ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Milano

INFN/BE-91/08
8 Luglio 1991

S. Banfi, P. Guazzoni, G.F. Taiocchi, L. Zetta:

CAP - A COMPUTER AIDED PULSER

Servizio Documentazione
dei Laboratori Nazionali di Frascati

INFN - Istituto Nazionale di Fisica Nucleare
Sezione di Milano

INFN/BE-91/08
8 Luglio 1991

CAP - a Computer Aided Pulser

S. Banfi[1], P.Guazzoni[2], G.F. Taiocchi[3], L.Zetta[2]

---

(1) Dipartimento di Fisica - Università degli Studi - Milano - *present address*: L.G.S. - Milano

(2) Dipartimento di Fisica - Università degli Studi and Istituto Nazionale di Fisica Nucleare - via Celoria 16 - 20133 Milano - Italy

(3) Ditta Takes - Ponteranica Bergamo - Italy

ABSTRACT

The pulse generator described in the present paper provides up to four analog signals, for testing nuclear electronic devices, using a data base of *true* data stored in a host computer, serially connected.

1) INTRODUCTION

The use of pulse generator for testing electronic chains for nuclear experiments is undoubtly useful. However even in the case of special devices(1), that are able to give dozens of different pulses, the simulated environment can describe only approximatively the experiment. This is true specially for the experimental resolution that, in general can not be taken into account in a simple way.

This was the main reason that induced us to design, build and test a different kind of pulser, CAP, a Computer Aided Pulser.

This pulse generator can be schematically described as a four-way digital to analog converter connected, via RS232-port, to a host computer. In fact CAP can give in output up to four independent analog pulses (0÷4V) of chosen shape, corresponding to the digital words stored in the host computer as an up to four-dimensional matrix, or to a four parameter event, directly read on the magnetic tape by the host computer. If a lot of four parameter events have been collected during an on beam experiment, and stored on a magnetic tape, these row data can be used as a data base for CAP.

It works as slave peripheral for the host computer, in need of a terminal, to initialize the computer section and to run the proper code.

2) GENERAL DESCRIPTION

2.1)*The front panel*

CAP is a two-units standard NIM module. A preliminary description can be made following the front panel (fig.1).
Five BNC connectors are used for the analog outputs: four corresponding to the four parameters of each event, and another that is doubled from the fourth, with independent attenuation. It can be used for service purposes.
Five ten turn potentiometers allow exact calibration of each analog channel. A two position toggle switch is used for choosing the working mode (SERial in connection with the host computer) and TEST. In the latter case twelve switches allow to build a test word of up to 12 bits (0÷4095).
Another toggle switch is used for the Sliding Scale correction.

2.2)*The digital input data*

The standard procedure for the asynchronous serial data transmission requires that the level on the communication line is high (with no characters transmitted). The transmission of a character implies the transmission of a train of bits. Because of the first of these, the signal goes low to syncronize the receiver; afterwards, with the chosen frequency, the character bits (first the MSB, more significant bit), the parity, the stop bits, needed to restore the initial conditions, will follow.
The RS232 standard provides, in addition to the *send* and

receive data signals, also for the *request*, *clear to send* and *data terminal ready*. Of all these signals CAP interface uses only receive data, clear to send and data terminal ready. This is allowed, because of the speed of CAP in data processing, higher than that of the transmission line. In this way the handshake is led by the terminal, while CAP is connected to the auxiliary port.

CAP requires that the input digital data arrive as a string of up to four words, each of four ASCII characters, in octal representation (three digits per character), starting with the most significant digit. To signify the end of each event of up to four parameters, a control character is needed. We chose a slash(/). When the control character is recognized, the data output begins.

2.3)*The working mode*

In the serial mode, CAP works connected to the auxiliary port of a terminal, used as master peripheral.

Inside CAP a simplified unidirectional interface, connected to the RS232C port at 9600 Baud, provides the necessary formatting and control for interfacing between serial (input) and parallel (output) data. A more complex bidirectional one is described in ref. 2.

In ASCII code the characters corresponding to the octal values 0÷7 have a binary configuration corresponding to that of the numbers 0÷7 with in addition the bits 5 and 6 always high, while bits 4 and 7 are always low. Otherwise, no other character has such combination for bits 4,5,6,7. So, we can use the content of these two bits to distinguish between octal numerical values (true data in our case) and the other words. In addition in this latter case, we can look for the control character "/" used as indicator of the end of a string representing one event. In this way we can transmit events of different number of parameters, from one to four. When a number is recognized, the three less significant bits of four subsequent characters are stored, in parallel way into four latches, giving rise to a 12-bit word; and so on up to the end of the event. The recognizing of the slash provides for the contemporary output of all the pulses. In addition CAP can generate, in TEST mode, a parallel word of up to 12-bits giving rise to an analog pulse up to 4 V, the same for each output, to be used for calibration.

2.4) *The sliding scale correction*

Regarding the accuracy of the analog output pulses, CAP can also introduce a sliding scale correction.(3,4,5).

In fact the use of commercial digital to analog converters may introduce a poor differential linearity, because of the channel width error. For a differential nonlinearity of 1% and a resolution of 4096 channels, the DAC output must be accurate to 2.5 ppm of the full scale range. So the required differential linearity can be achieved using the sliding scale method which consists in subtracting a random level from the analog signal, ranging up to a few bits, while adding the equivalent digital code to the digital datum.

In our case we chose a value up to 127 (7-bit word) to add to the digital incoming datum. The result has to be in any case less than 4095. This means a reduction of 1/32 of the full allowed range. Then, after the digital to analog conversion, the same value, separately converted into analog pulse, is subtracted to the analog datum. The sliding scale counter (i.e. the used value) is increased of one unit at the end of the output linear gate (LG) opening.

2.5)*Schematic description*

A schematic description of CAP can be given following fig.s 3 & 4, where, respectively, the block and the timing diagram are shown.

CAP, as previously seen, can work in TEST or SERial mode. In test, the 12 toggle switches are used to set the binary word, in order to be translated into analog pulses, the same for all the outputs.

In SERial, event comes from the host computer, through a RS232 serial line. The serial word, at the beginning, enter an UART (Universal Asynchronous Receveir and Transmitter). A decoder after the UART, recognizes, on the ground of the binary structure, if the character is *good* ("number" or "/")or not.

If it is *good* and it is a number, the event clock starts and the four 3-bit characters are counted and stored in the four 3-bit buffers, used as 12-bit memory.

In this way the first word (A-parameter) is stored, at the 1st internal clock sweep, and so on up to the 4th one (D-parameter). During the handling of a word, a monostable locks the transfer, until the last 3-bit character is loaded.

The delay starts at the trailing edge of the lock monostable, while the LSB/MSB clock starts at the trailing edge of the delay. At the trailing edge of the 1st clock sweep, the LSB/MSB monostable of each parameter starts. When it is high, the less significant part of the word is transferred via the 6-bit bus. At the trailing edge of the second clock pulse, the more significant part of the word is transferred.

We also have to remember that the sliding scale adder is enabled at the beginning of the transfer sequence, and it is always ready to work, beginning from this moment. At the same time (in coincidence with the 1st clock pulse leading edge) the DAC latch, corresponding to the 1st 6-bit string of the word, is enabled. At the 2nd clock pulse leading edge, the latch of the second 6-bit string is enabled. And so on up to the 4th parameter of each event. In fact, at the 16th clock pulse (or in case of a number of parameter less than 4, reading the "/" character, representing the end of event), a 3 us delay is enabled. On its trailing edge, the enable pulse for the output LGs starts. They are open for the shaping time chosen for the analog output pulse (≃3 us in our case).

The increment of the sliding scale counter starts at the end of the LG opening. So it is immediately ready to handle a new event. The analog output pulses exit during the open time of LGs. We have to remark that all the output pulses are coincident, because of the LGs for all the four outputs are open at the same time and for the same width.

3) SOFTWARE DESCRIPTION

The main task of the software written for CAP, is to control the data flow from the host computer, choosing from a variety of *true* data structures, residing on disk volumes or software generated, with the supervision of an operator. Actually the whole procedure is driven by a set of commands which can be entered through a terminal, connected to CAP by means of the auxiliary port. Such commands allow to select the particular data file to be sent to CAP, in order to set various operating parameters, or to start, stop and resume at any time the data flow. This is accomplished taking full advantage of full-duplex communication line.

3.1)*Data Management*

The availability of a large and easily accessible data base is an outstanding characteristic of the system.

Two kinds of data structures are actually recognized by the program:

MATRIX: bi-dimensional matrices, where sample data are collected from tapes recorded during on beam experiments, and arranged in events at discrete row and column coordinates;

LIST: each event being a set of up to four parameters, either computed by means of a simulation code$^{(6)}$, or collected on tape or disk during a true experiment.

Matrices are stored in binary format; for good resolution they need a considerable amount of disk and memory space. Event lists are sequential ASCII files storing a smaller number of values in numerical format; they can also be easily created and edited. Furthermore, for electronics testing, the program can generate by itself data values, simulating increasing and decreasing ramps over the full range.

3.2)*CAP Control*

Data sent to CAP, have to be formatted in events, as previously shown. CAP software is responsible to fetch, from previous data structures, event values, to format them as described and to send the resulting buffer to CAP.

Events are continuously sent until the "end of data" condition is reached. In the meanwhile the receiver line of the link is tested to check if the stop command has been entered at the keyboard: in this case the transmission is paused. At this point some operating parameters can be changed and the transmission resumed or aborted.

3.3) *Design standard*

Some standards have been adopted in designing CAP software with the aim of improving reliability, program diagnosis and modifiability[6]. The whole system has been split up in two parts: the control subsystem and the controlled one. The former has the responsability to activate the whole system, following the external inputs, provided interactively by the operator, and in accordance with the status reached at the same moment by the system. This module performs also all the decisional activity, validating or rejecting input commands. The controlled subsystem is subdivided in as many modules as required by the variety of program performances; the specific modules are organized in a tree structure whose depth depends on the complexity of the task performed. All modules, in the controlled subsystem, can be described in terms of input-output relations.

3.4) *Control and data flow*

The fig.5 shows the main data structures and I/O devices involved, together with the data flow as carried by program modules.

Data sources for CAP, as previously said, are disk files or the program itself in case of test ramp generation. Once selected, the data file is read into the computer memory, to speed up the event fetching. Read and fetching operations are performed by specific modules corresponding to the particular data structure to act upon: matrix or list.

Regardless to data source type, digital values of fetched events are then arranged into CAP events, i.e. from one to four 12-bit words, followed by the end of event ("/"). Such event represents the unit recognizable by CAP for analog conversion; a module solves all the format and handshaking requirements needed to perform output to CAP. Moreover, the data exchange between the operator and the control subsystem (actually, on behalf of the various modules of the controlled subsystem) is always active, allowing command input and status output.

With respect to the shown diagram (fig.5) the terminal (communicating directly with the control subsystem) and CAP share the same physical link to the host computer. The software fits the diagram, providing virtually distinct and parallel I/O operations.

Following the adopted standards of design, all the control flow is driven by the control subsystem module, directly interfaced to the terminal for the input of commands. This module keeps also trace of system evolution, by means of a number of predefined status, as determined by command sequence and conditions, reported by the controlled subsystem.

Each command is checked for consistency with the current system status, and executed activating the involved modules of the controlled subsystem, in the proper sequence. Command validation is table driven: for each status allowed commands are presented on the terminal.

3.5)*Implementation notes*

The control subsystem is implemented as main program. It is composed of a section where prompt is displayed, with help if required. At the terminal, command input is entered and checked. If successful, one following section is activated, where control branches, on the ground of the input command. Then the controlled subsystem is allowed to start , by means of subroutine calls.

All the above sections are nested in a loop, which can be exited with the "end" command. Inside this loop, the keyboard is also polled to allow the continuous control before the end of data; special care is put in order to avoid of altering the data sent to CAP over the shared link

The program is written in FORTRAN and also uses some VAX-VMS System Services call for special functions such as keyboard polling and time synchronizing. It is composed of a single source module of about 2000 statements, and produces a single executable image running on any VAX or VMS-system computer.

4) TESTING

CAP was preliminary tested in order to know its performance either from hardware or software point of view. Secondly, it was used to test known analog calculators, to verify the possibility to use it in new apparatus testing.

4.1)*Performance*

To verify CAP performance, different sets of tests were made, in order to explore its behaviour in various conditions.

At the end, a set of 40 equispaced (100mV to 4V, step 100mV) digital quadruplets were sent for an one week test to CAP and its four output pulses were registered into a SILENA CATO MCA(7), with an ADC-sensitivity of 1 mV/ch, using MUDA(8), a Mixer Unit for Data Acquisition. So it was possible to test at the same time the different parameters of CAP. The obtained results, shown in the following, are the same for the four outputs.

<u>Stability</u>: better than 1 mV/day

<u>Resolution</u>: mean resolution over 40 different peaks, better than 1.5 mV, without , better than 2 mV with sliding scale correction.

<u>Integral nonlinearity</u>: less than $\simeq$ 0.001% over the full range without and with sliding scale correction.

<u>Differential nonlinearity</u>: less than 0.2%over the full range without and $\simeq$ 0.1% with sliding scale correction.

Regarding to the output pulses, as previously seen, the amplitude is continuously variable in the range 10mV ÷ 4V, with a shaping time of about 3 us (fig.6).

A test was also made using the software ramp generator. Also in this case the results are in good agreement with that previously exposed.

4.2)*Experimental Results*

For the second set of tests, CAP was connected to various systems to control its behaviour in testing complex apparatus.

Among other tests, CAP was used in connection with an analog identifier[9] for light particles, using simulated data[6] for the reaction $(p+^{27}Al)$ to obtain light particles up to ^{4}He at incident proton energy of 27 MeV, using a SSD telescope with the thin detector 0.1 mm thick. The identification spectrum (PIF) obtained is shown in fig.7.The upper part of this figure represents PIF computed by means of the analog identifier of ref. 9, using the E and DE pulses coming from CAP. The lower one shows PIF software computed and transmitted through CAP to MCA. It is possible to see that in the first case the width of peaks is larger than in the second one, due to electronic noise in analog PIF computing.

We can conclude that CAP is a quite simple, reliable, and useful tool in complex apparatus testing, where the careful reproduction of the experimental environment is required.

The authors aknowledge the helpful cooperation of Mr. P.L. Dellera in the figure drawning.

This work was partially supported by a grant of the Italian Ministry of Scientific Research.

R E F E R E N C E S

(1) P. Guazzoni, P. Michelato, G.F. Taiocchi, L.Zetta
Nucl. Instr. and Meth. 185 (1981) 219

(2) W. Bragagnini, P. Guazzoni, M. Pitalieri, G.F. Taiocchi, L. Zetta - Report INFN/BE - 90/09

(3) C. Cottini, E. Gatti, V. Svelto
Nucl. Instr. and Meth. 24 (1963) 241

(4) S.G. Gobbur, D.A. Landis, F.S. Goulding
Nucl. Instr. and Meth. 140 (1977) 405

(5) M. Brendle - Nucl. Instr. and Meth. 144 (1977) 357

(6) P. Guazzoni, G. Sechi, L. Zetta
Nucl. Instr. and Meth. 227 (1984) 526

(7) Ditta Silena - Cernusco S/N Milano - Italy

(8) P. Guazzoni, G.F. Taiocchi, L. Zetta
Nucl. Instr. and Meth. in publishing
Report INFN/BE-90/06

(9) P. Guazzoni, P. Michelato, G.F. Taiocchi, L. Zetta
Nucl. Instr. and Meth. 200 (1982) 323

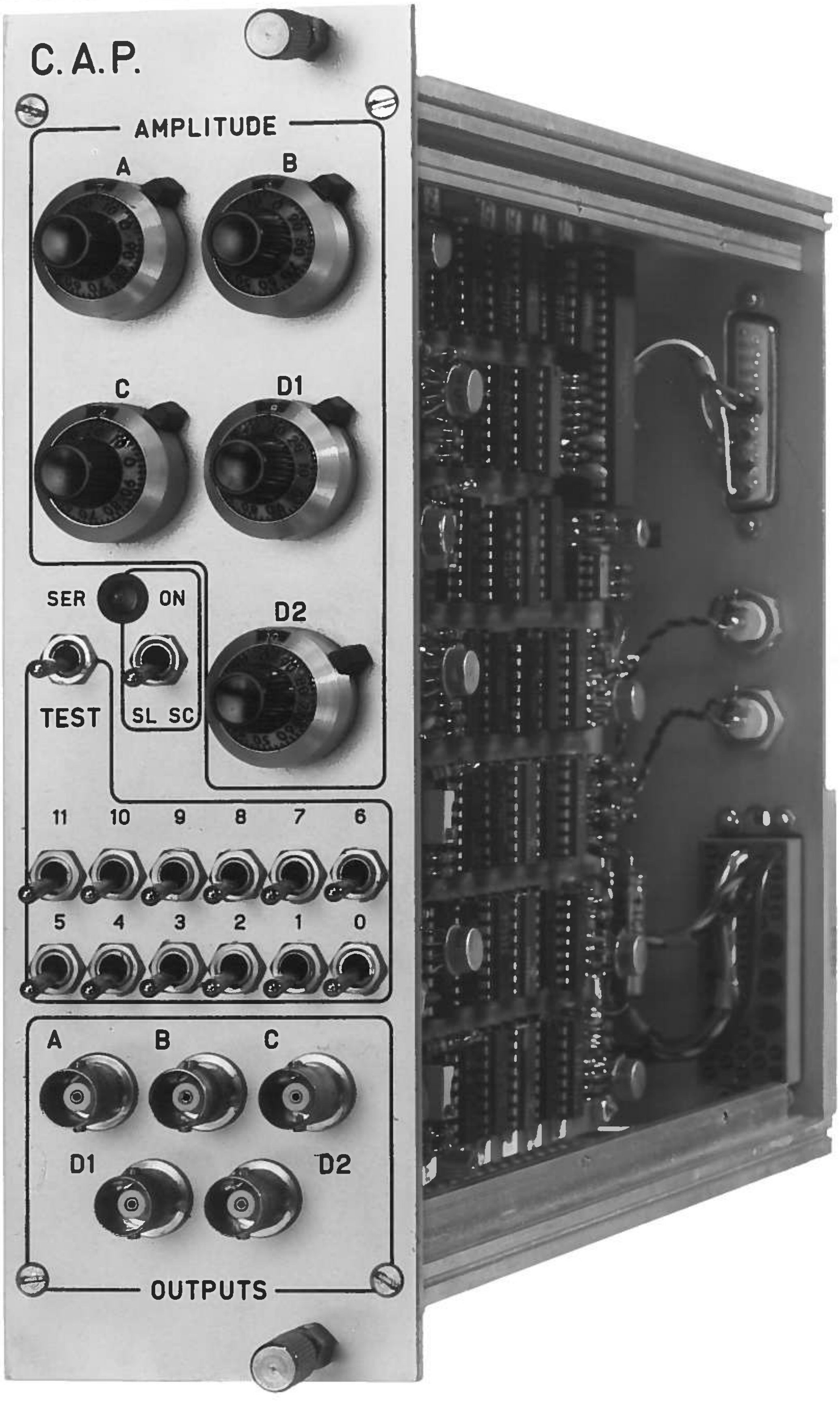

Fig.1 - The front panel

-1-

SCONTORNATO

$512_{10} = 1000_8$

PARAMETER STRUCTURE

0	0	1	0	0	0	0	0	0	0	0	0
4th CHAR.			3rd CHAR.			2nd CHAR.			1st CHAR.		

RS 232

CHAR. \ BIT N°	7	6	5	4	3	2	1	$\binom{\text{ASCII}}{\text{VALUE}}_8$
1st	0	1	1	0	0	0	0	60
2nd	0	1	1	0	0	0	0	60
3rd	0	1	1	0	0	0	0	60
4th	0	1	1	0	0	0	1	61
	RD 7	RD 6	RD 5	RD 4	RD 3	RD 2	RD 1	UART
OCTAL NUMERICAL CHARACTER	OFF	ON	ON	OFF	X	X	X	

EVENT STRUCTURE (UP TO 4 PARAMETER)

A - parameter	B - parameter	C - parameter	D - parameter	/

Fig.2 - The event structure

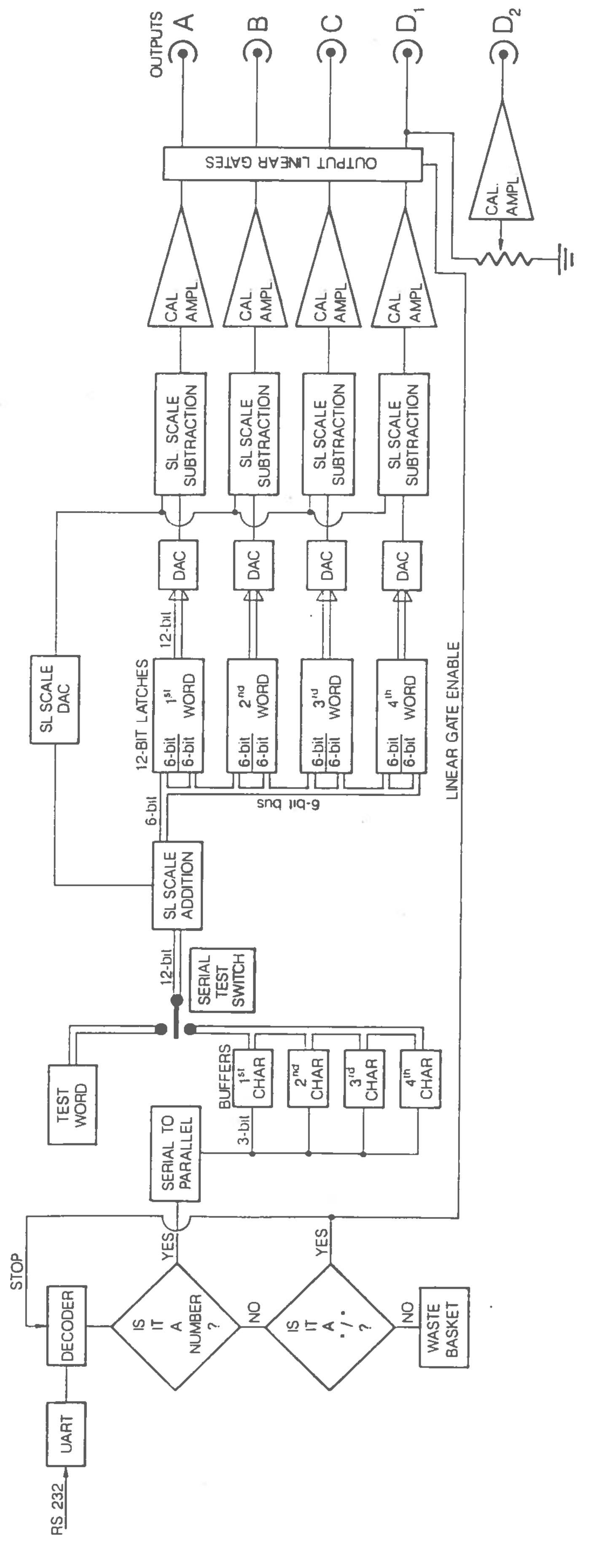

Fig.3 - Simplified schematic

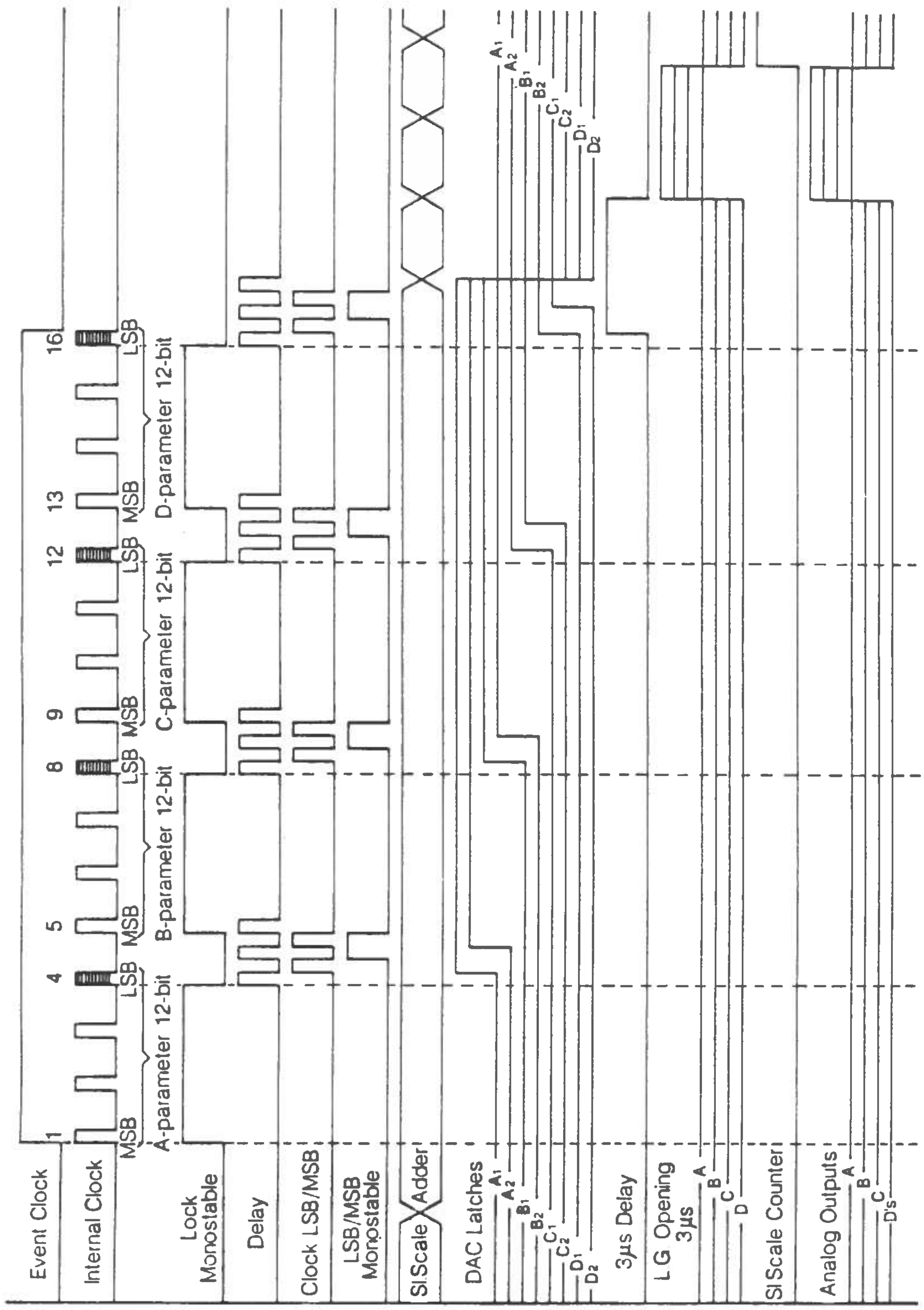

Fig.4 - Timing diagram

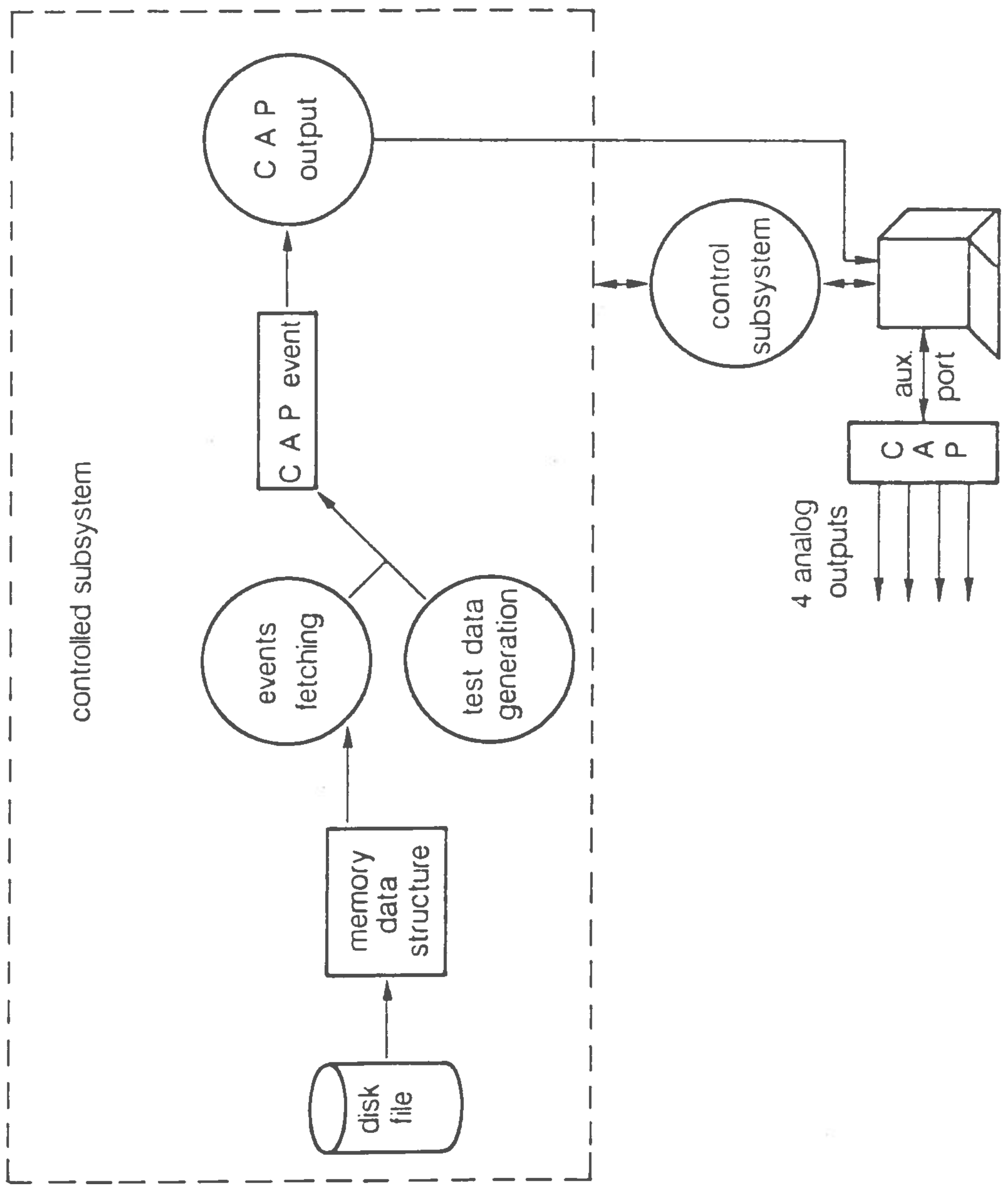

Fig.5 - Program diagram

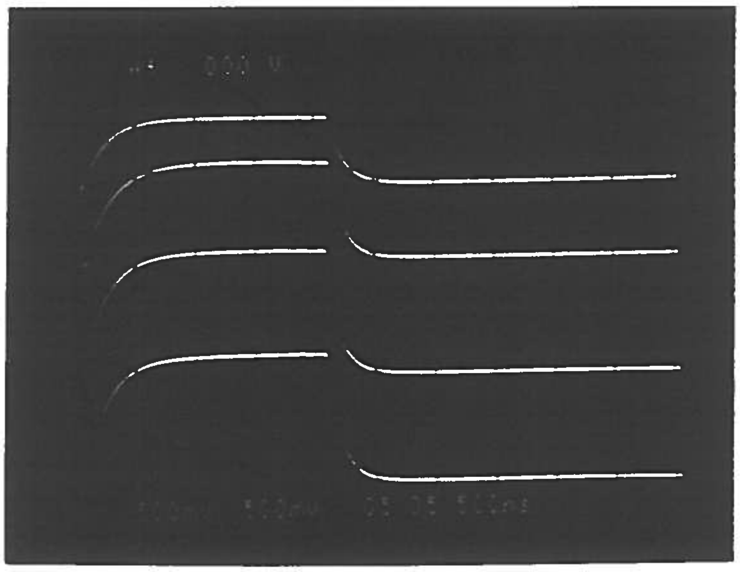

Fig.6 - Output pulses

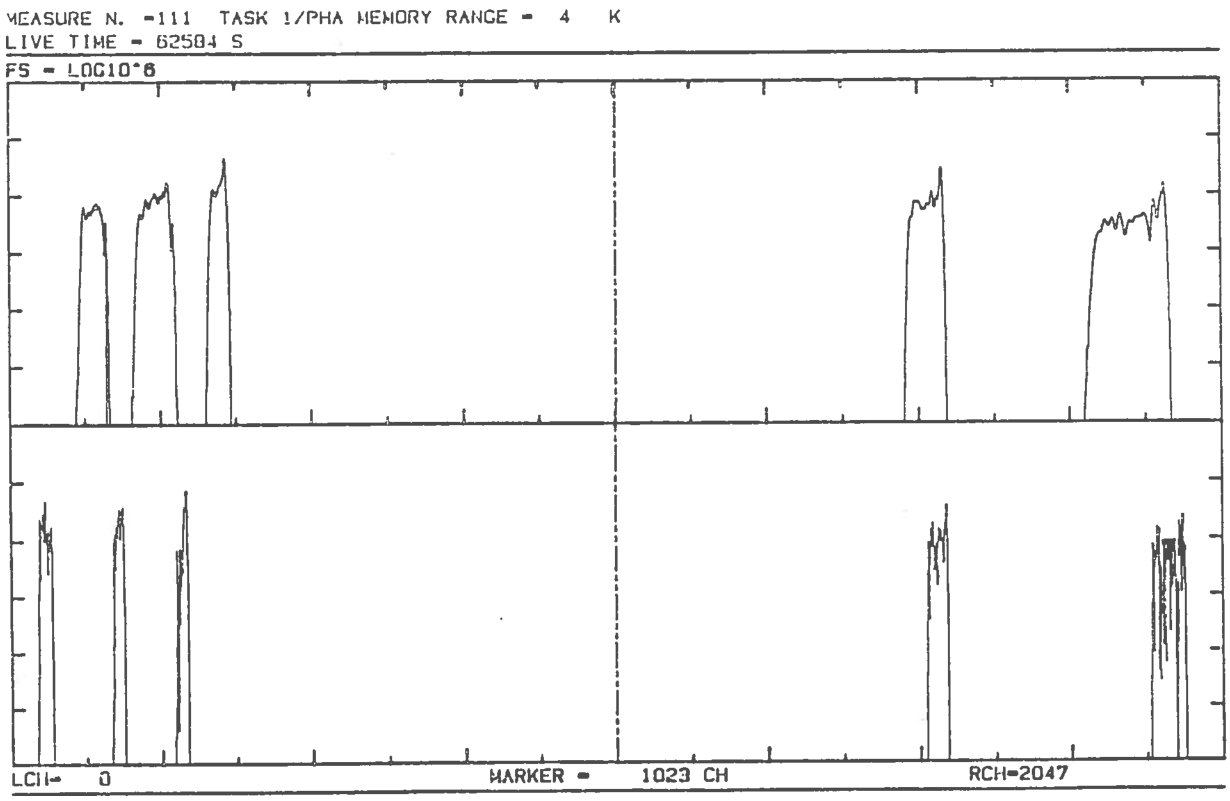

Fig.7 - Identification spectrum for a simulated reaction