SPS/ABM/Techn.Note 79-9 3rd October 1979

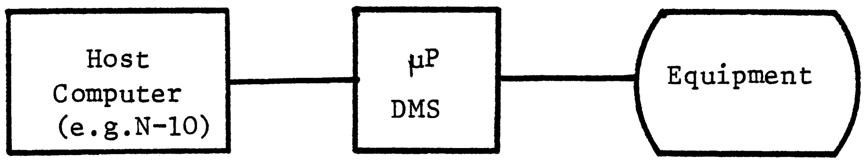
PROPOSAL FOR THE USE OF DATA MODULE SUBROUTINES ON MICROPROCESSORS

J. Altaber, F. Beck, L. Burnod, E. d'Amico

1. Principles

The advantages of a decentralized control system using the multiprocessing facilities have already been demonstrated.

A further step in this direction can be reached by moving to a microprocessor attached to a host computer the management of a task previously done in the host. For the SPS control system, this can be done by putting a Data Module Subroutine into a microprocessor which will have full control of the associated hardware.



The autonomy of such a system allows several DMS's to act in parallel; it increases the programming flexibility at the DMS level and avoids the tedious generation of the complete system when only a minor modification is done on a DMS.

From the user's point of view the interest of this system is only valid if some basic requirements are satisfied, such as:

- The coding must not be written in μP-type language. The fast development of the μP technology imposes the use of high level language, like NODAL, which is independent of the hardware used. In addition such a language, which may be either interpreted or compiled, allows a more comprehensible coding and in its interpreted form it provides easier debugging. Once compiled and tested, it can be loaded into an EPROM, to avoid dangerous modification by non-specialists and to avoid the need for repeated down-line loading.

- The communication package between the μP and the main control system (N-10 computers) has to be transparent for the user. All existing access facilities to the DMS will be kept such that

  > SE DMS (N, ≠PTY) = A

  > T DMS (N, ≠PTY)

  and all call statements > CALL DMS(A, "R", N, ≠PTY)

  N = Equipment Number

  PTY = DMS property

  A = Array

  'R' = Read

  'W' = Write.

- Specified NODAL elements which are used in the microprocessor when it executes the DMS can be so identified at generation that they will also be accessible by a NODAL program running in the host computer.

- Connections between μP and equipment hardware will use the existing interfaces CAMAC or MPX.

- Some DMS's require the execution of programs (called from now on RT-programs) on the occurrence of specified SPS cycle events. This implies that the DMS writer can use high priority software levels in the microprocessor and that the compiler provides a very efficient object code to satisfy response time requirements on the handling of these events.

- To allow easy diagnosis and repair clear hardware and software interfaces will have to be defined between the user's equipment and DMS coding which, as usual, will be under the responsibility of the user. On the other end, the μP hardware and software package, up to the interface will be under the responsibility of the control group people.

When all these requirements have been met the advantages of such a system become obvious: independence relative to the advance of technology for either the μP or the computer, facilities for testing or improving the system while the equipment protection and application programs stay unchanged.

At the stage of the $p\bar{p}$ project, several new DMS's have to be written and others improved. Most of them are linked to the SPS cycle real-time, such as:

PAPOS[1)] acquisition of the 'BPC' beam position monitors (directional couplers)

PAPI[1)] acquisition of the 'AES' beam intensity monitors

QMEAS[1)] Q measurement in $p\bar{p}$ mode

MISEM[2)] acquisition of the secondary emission monitors in TT10 and TT60.

PAGEF[3)] control of function generators in $p\bar{p}$ mode.

A preliminary study has been carried out to implement the above principles for one of these new DMS PAPOS, by using the microprocessor CAMAC ACC SPS 2420 and its associated software facilities.

2. Application to PAPOS

The directional couplers will be used in $p\bar{p}$ operation to provide the position of both the proton and antiproton beams. The acquisition hardware will be essentially the same as for the electrostatic pick-ups presently used, i.e. CAMAC scalers will count the frequency provided by V/F converters. New features will be the switching protons/antiprotons, the bunch selector (TBS) and the larger data gathering flexibility permitted by the use of a 'sequence'. The latter will help to cope with the special requirements of $p\bar{p}$ operation: a long injection flat bottom, followed by acceleration and low beta build-up, and finally the storage period lasting 24 hours. App. 1 shows a possible master timing scheme for $p\bar{p}$ operation.

The mode of operation foreseen is as follows:

Acquisition

Each acquisition will be triggered by a timing module or by programmable timer. The sequencer will control the gating of the scalers and send an interrupt I1 to the μP. The interrupt I1 will start a real-time program RT1 which will carry out the following functions:

a. read and clear the scalers,
b. examine the next element in the sequence. If it exists, it prepares the element by programming the TBS module and/or the $p\bar{p}$ switch (controlled by a MPX module). Otherwise exit.

If the sequencer execution is controlled by software, it activates the sequencer which will eventually issue another I1 interupt.

Reset

The PAPOS acquisition system reset will be activated by one of the following timing pulses: the standard 'reset' pulse of the master timing at the end of each machine cycle (whether simulated or not), and the 'start coast' and 'stop coast' events.

The real-time program RT2 will react on this system reset. Its tasks are:

1) re-initialise interested CAMAC modules,
2) check if all the requested acquisitions have been carried out correctly,
3) load the new 'sequence' for the next machine cycle and - if required - reprogram the TBS module and/or the $p\bar{p}$ switch, corresponding to the first element in the 'sequence'.

DMS PAPOS

The data module PAPOS itself will provide standard properties as ≠CON, ≠GA1, ≠SWI and so on. The 'sequences' will be entered via the property ≠SEQ (using the CALL format).

The large number of acquisition properties will be somewhat reduced by suppressing 'difference' properties and coding the property name of ≠XYZ, where

X is P or A, according to proton beam or antiproton acquisition,

Y is the bunch number in hexadecimal, or T when the sum of all bunches is required,

Z is the 'sequence' number.

It is proposed to store the complete DMS in the ACC, the associated data-table and property table as well.

It is important to note that no provision for re-entrancy is foreseen in the coding of PAPOS. Conflicts between NODAL levels in the NORD-10 (imex, execute, schedule, interactive) will be solved by the NORD-10 monitor.

Appendix 2 shows the functional block diagram for PAPOS.

3. Responsibilities and the time scales

The proposed scheme for DMS implementation in the ACC requires the provision of the following software:

1) In the NORD computer:

An off line NODAL compiler for the TMS-9900 used by the ACC.

Facilities for accessing:

-DMS in its different forms (DMS),
-service routine for the DMS typically LISDM (SDMS),
-hardware which is under full control of the ACC typically for MPX driven equipements, it requires CAMPX subroutine (HARDW),
-specified Nodal elements of program running in the ACC (EXAM),
-error messages detected by the ACC (ERROR).

2) In the ACC:

-A dispatcher which routes the request to the appropriate module and passes parameters.

-A module for each request and the necessary facilities for these modules to collect data from the N10 and pass results back to it.

-RT-program for servicing external interrupts from machine cycle events.

-Tests programs which run in background and check the ACC hardware and user equipements.

The modules arrangement is shown in Appendix 3.

A * indicates modules which are provided by the Control group while modules which are underlined are the responsability of the users.

A Nodal compiler for the ACC is already available and a user's guide is in publication.

From now on the Control group will provide the users with all information required for writing software modules for the ACC with the mode of operation for getting data, parameters and sending results.

A preliminary working package, not including SDMS+EXAM+ERROR and background test will be delivered by the end of October.

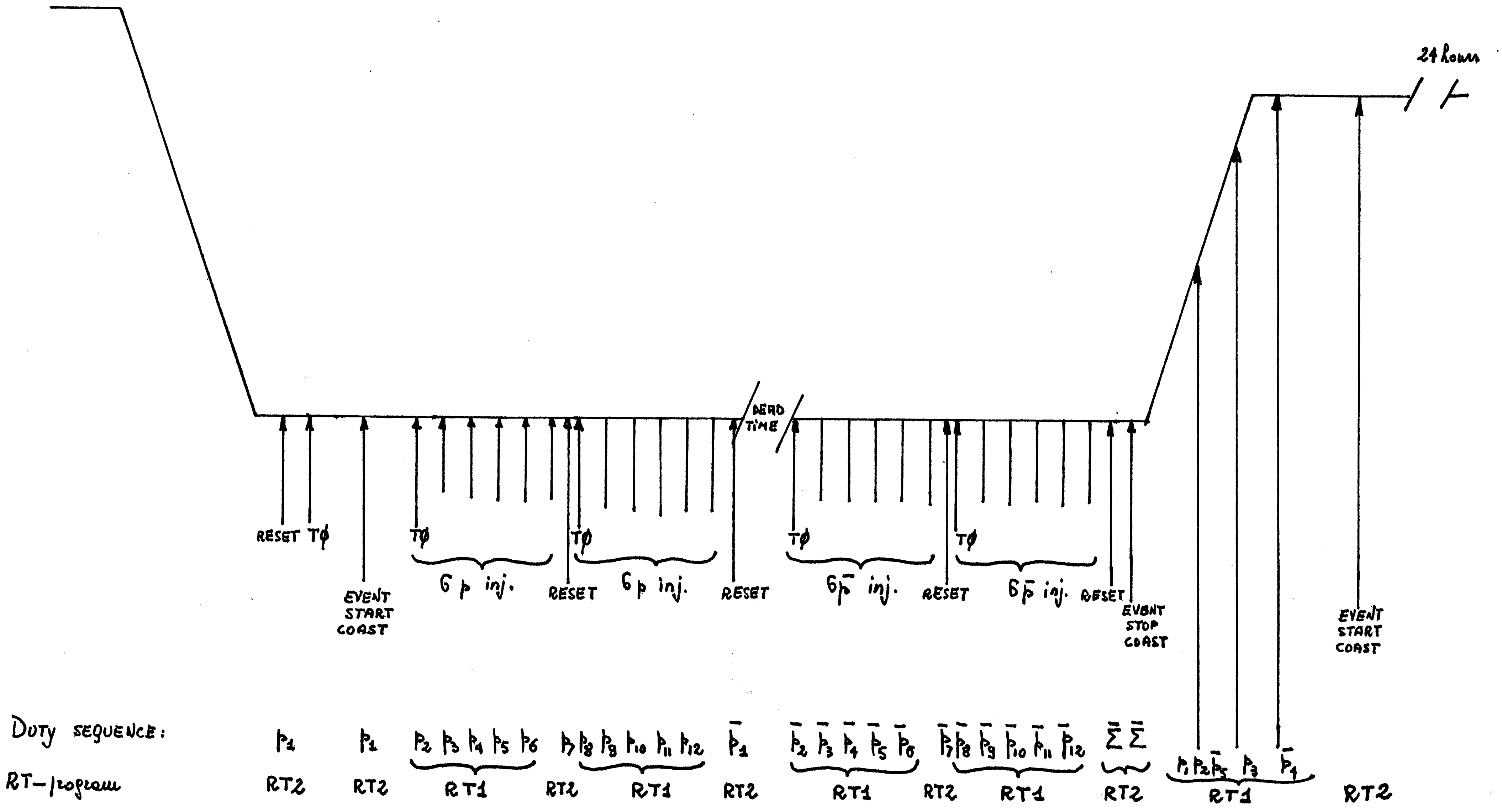
The complete package is due to be released by the end of November.

N.B. When dates are mentioned the implicit year is 1979....!!!!

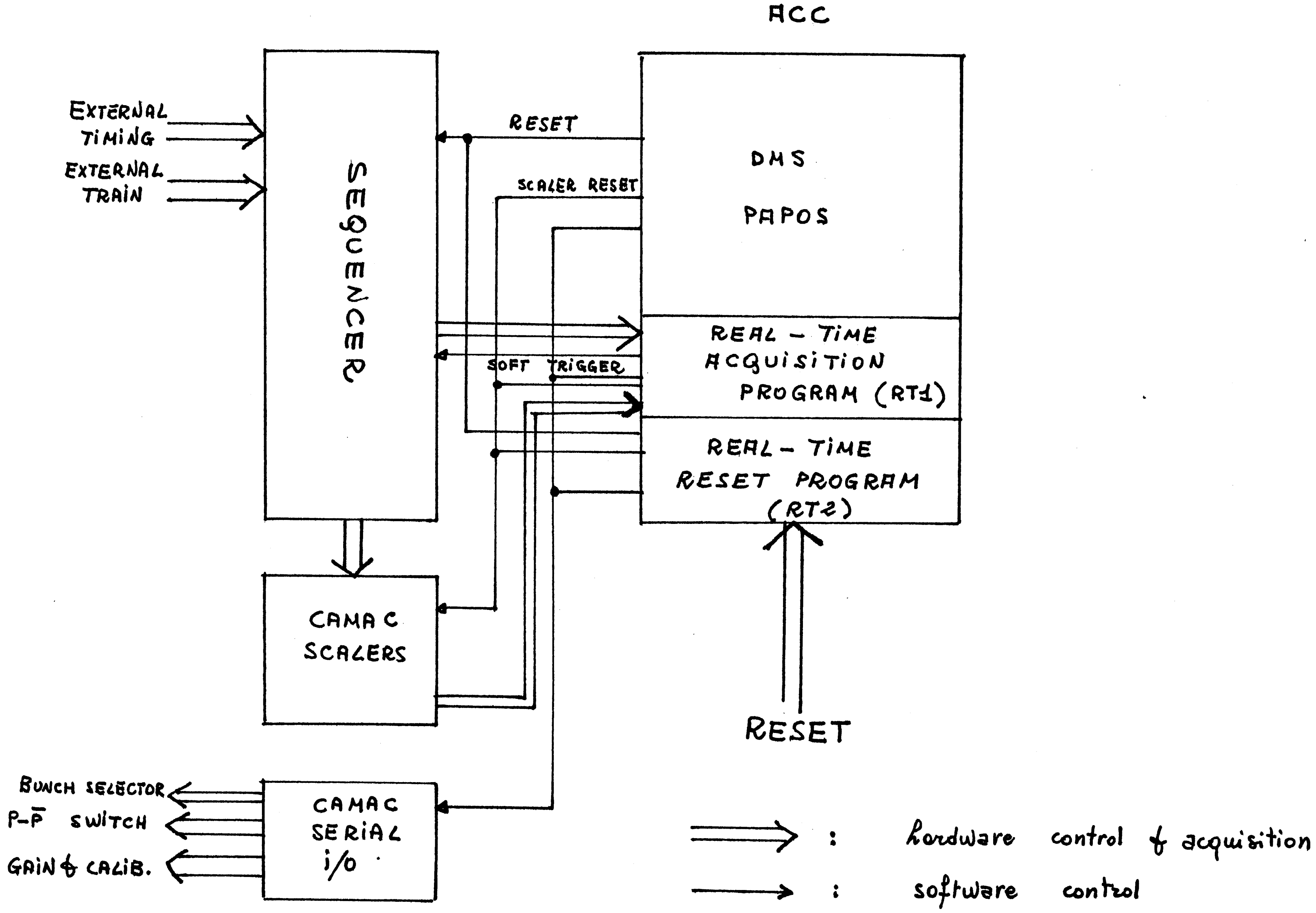
References

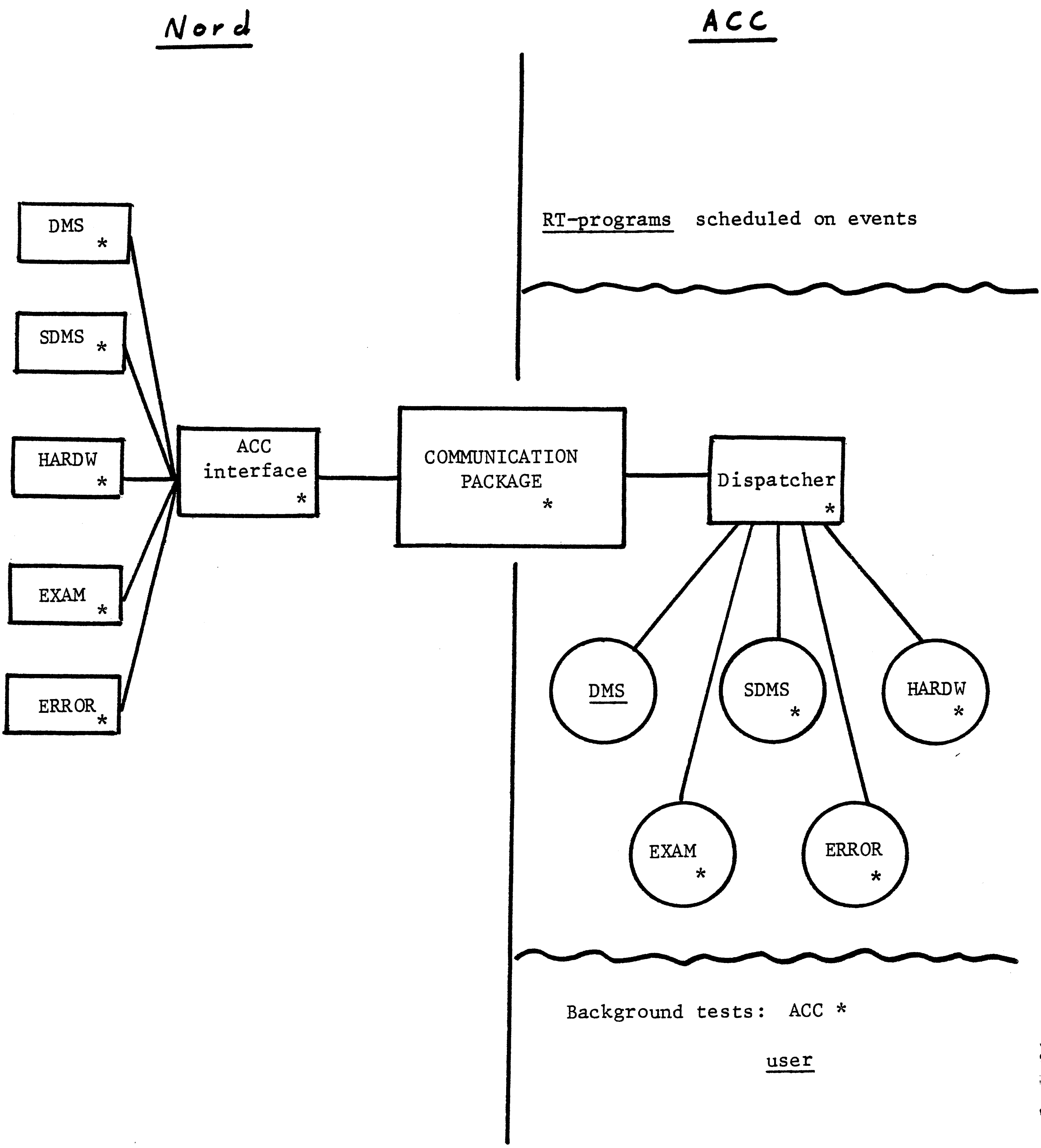
1) SPS/ABM/LB/Note 79-6, 18.6.1979.
2) J. Bosser, J.H. Dieperink. Private communication.
3) SPS/ABM/GM/Note 79-5, 15.6.1979.

REAL-TIME SEQUENCE FOR P-$\bar{P}$ OPERATION



FUNCTIONAL DIAGRAM FOR PAPOS





Modules arrangement