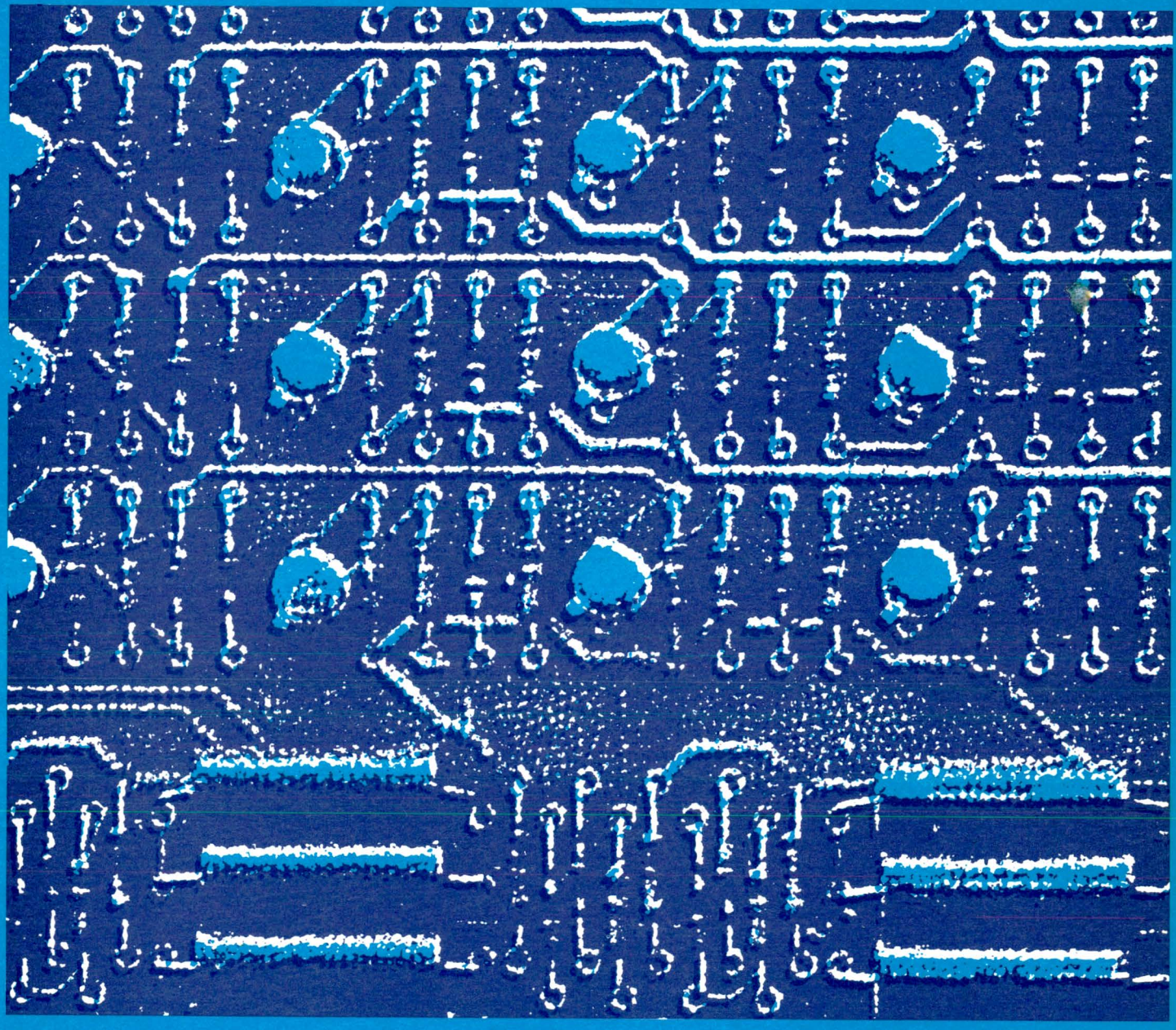
P800M Programmer's Guide 1

Volume V: Small Real Time Monitor



PHILIPS Data Systems

PHILIPS

P800M Programmer's Guide 1
Volume V: Small Real Time Monitor

List of Effective Pages

1 - 24	original
25	revision 1
26 - 30	original
31 - 32	revision 1
33 - 90	original

Original Issue: May 1975 Revision 1: May 1976

A publication of

Philips Data Systems B.V.
Marketing Group Small Computers
Apeldoorn, The Netherlands

Publication number 5122 991 27351

May 1976

Printed in The Netherlands

This handbook is part of the set of P800M manuals for the standard papertape-oriented software:

P800M Programmer's Guide 1 - Volume I: BOM	5122 991 2732X
P800M Programmer's Guide 1 - Volume III: S/W Proc.	5122 991 2733X
P800M Programmer's Guide 1 - Volume IV: BRTM	5122 991 2734X
P800M Programmer's Guide 1 + 2 - Volume II: Instr.	5122 991 2738X
P800M Programmer's Reference Data	5122 991 2743X

Great care has been taken to ensure that the information contained in this handbook is accurate and complete. Nevertheless, should a user find any errors or omissions or wish to suggest improvements, he is invited to write his comments on the sheet provided at the end of the book and send it to:

Mr. D.G.M. Bos or Mr. A.C.P. Debie

NV Philips
MID - S&I/IDS
Building TQV-3
Eindhoven
Tel. 040 - 782563 or 784009

CONTENTS

1. INTRODUCTION

1.1 The Small Real Time Monitor is upward compatible with the Basic Real Time Monitor, which means that user programs running under the SRTM can be used without modification under the BRTM (see 10).

The SRTM is suitable for dedicated computer applications requiring a small and fast monitor.

The SRTM is paper tape oriented and does not use disc equipment. Because of the modularity of this monitor, it is possible to add modules and thus use the corresponding peripheral units, disc excepted.

The system offers the following real time possibilities to the user:

1. connecting user programs to interrupt levels.
2. multiprogramming between user tasks (up to 14 software priority levels).
3. centralized input/output handling.
4. re-entrancy of subroutines.
5. recording the status of each program and providing re-entry to intermediate points in interrupted or suspended programs.
6. for communication purposes between user tasks and the monitor, several socalled Monitor Requests are present.
7. Repetitive or single activation of user tasks at specified time-intervals.
8. Updating of time and date.

1.2 Software Modularity

In order to fit with different configurations, SRTM has been built in such a manner as to make it as modular as possible.
That means, parts of it can easily be cancelled at configuration time, depending on the needs.

Configuration is done on object level, supposing user defined tables have already been assembled.

2. FUNCTIONAL CHARACTERISTICS

2.1 Priority System

The priority system is based on a 64 level structure. These priority levels are subdivided as follows:

1. levels 0 to 47 are hardware priority.
   Programs executed on a hardware priority level are named: Interrupt Mode Programs (IM-)
2. levels 48 to 63 are software priority levels, assigned as follows:
   - 48 is the standard system level of the monitor
   - 50 to 63 are levels assigned to user tasks
   - to be compatible with the BRTM, the use of level 49 is forbidden.

The higher the level number, the lower its priority, i.e. level 0 has the highest, 63 the lowest priority. From this it follows that the hardware interrupt always has priority over software levels.
Programs executed on a software priority level are named: Non-Interrupt Mode Programs (NIM-).

2.1.1 Hardware Interrupt Lines

To these lines (memory locations /0 through /5E) routines are connected which are required to service internal or external hardware interrupts, such as from power failure, real time clock, peripheral devices, etc.
These routines, connected to any of the levels 0 through 47, will, when called through an interrupt, run on the priority level associated to that interrupt. Only a higher priority interrupt will be able to overrule the running one.

2.1.2 Software Priority Levels

This part of the priority system permits multi-tasking. The priority levels 50 to 63 are connected to user tasks. One or more programs may be connected to the same level. These tasks must be activated through monitor requests. It is possible for one task to activate another one.

2.2 Dispatcher

Requests for activation of a program are handled by the dispatcher, a monitor part (on level 48) which divides the central processor time so as to start program according to priority.
The dispatcher can only be entered from IM programs.
Apart from the IM-parts of the various I/O drivers, entries will only be made from the LKM interrupt handling program.
The dispatcher investigates which program has to be started next. This will be either the interrupted one, or in case the interrupted program ran on a software level, a program on a level higher than the level of the interrupted one.

2.3 Start a program

To be started by the dispatcher, a program must be connected to a level. It can be activated in one of the following two ways:

1. by an interrupt, for an interrupt level
2. by an activate monitor request for a software priority level

Since there are several tasks running concurrently, a program passes through various states:

1. inactive	connected to a level but not having been requested.
2. active	the program has been requested and has not been terminated.

3. wait for execution — the program is ready to use central processor time when it has priority.

4. wait for an event — the program gives up control voluntarily. The program has requested an external event (I/O) and waits for its occurence or waits for the completion of a user event.

2.4 Wait for an event

A running program can request the monitor to initiate I/O operations which will be executed parallel to the requesting program. When the running program needs the results of these requests, it can perform a Wait monitor request.
If the results are available, the calling program is restarted. If not, it is put into the "Wait for an event" state and the program will not be restarted before these results are available; central processor control is then given to a program of a lower priority.
A program can also wait for the completion of a users event. In that case a user program is responsible for completing the event on which other programs are waiting.

2.5 Input/Output function

I/O operations are not permitted directly to the user program but they can be initiated through a request to the monitor for peripheral units. I/O modules and tables have been filled up inside the monitor at generation time.
There are two kinds of I/O requests:

1. Basic I/O request: where the monitor does not provide for character checking or data conversion. It will only provide for control command initialization and end of operation signals.
2. Standard ASCII I/O request: some facilities are available, such as error control characters. There is a character checking for end of data and the characters are stored 7 by 7 bits, two characters per word.

2.6 Attach/Detach device

By means of this monitor request a program running at any software level can reserve the use of the specified device to itself; in that case no other program can perform any I/O operations on this device, provided that the other program is also fitted with an attach, which is obligatory.
If the device has already been attached to another program, the calling program is put in wait state until the specified device has been detached.
Programs waiting for attach are served according to their priority.

2.7 Timerfunction

2.7.1 Time-intervals

The term "timer" refers to the various time-intervals on which software level programs can be activated by this module.
These intervals are multiples of

- RTC pulse rate
- standard pulse rate (20 msec)
- 100 msec
- 1 second
- 1 minute
- 1 hour

each with a solution of 2047 counts, so that the maximum time interval is 2047 hours.

2.7.2 Connection to a time-interval

Any program, whether it is a hardware- of software level program can connect a software level program to one or more time-intervals. At connection time the following specifications shall be made:

- basic time-interval
- name of the program to be connected
- offset, i.e. time between connection and first activation
- interval, i.e. time between two consequetive activations.

2.7.3 Disconnection from a time-interval

At any moment a software level program which has been connected, can be removed from the time-interval again.

2.7.4 One-shot

It is also possible to activate a program only once, i.e. after expiration of the offset.

2.7.5 Time updating

Each second the binary representation of the time (seconds, minutes and hours) is updated. These values can be obtained via external declarations.

2.7.6 **Date updating**

As an option updating of the date (day, month and year) can be done.

2.7.7 **Level of the timerfunction**

The timerfunction itself is a software level program, activated by a small Interrupt Mode program.
The level of the timerfunction can be specified by the user.

3. EXTERNAL INFORMATION

This chapter provides the user with the information needed for programming his system using the Small Real Time Monitor; it does not include details on monitor configuration and the use of usertask-monitor interfaces (i.e. Monitor Requests). See therefore chapters 4 and 6.

3.1 Interrupt Mode Programs

When an interrupt signal is raised, an automatic link is performed through one of the locations in:

3.1.1 The Interrupt linkage table

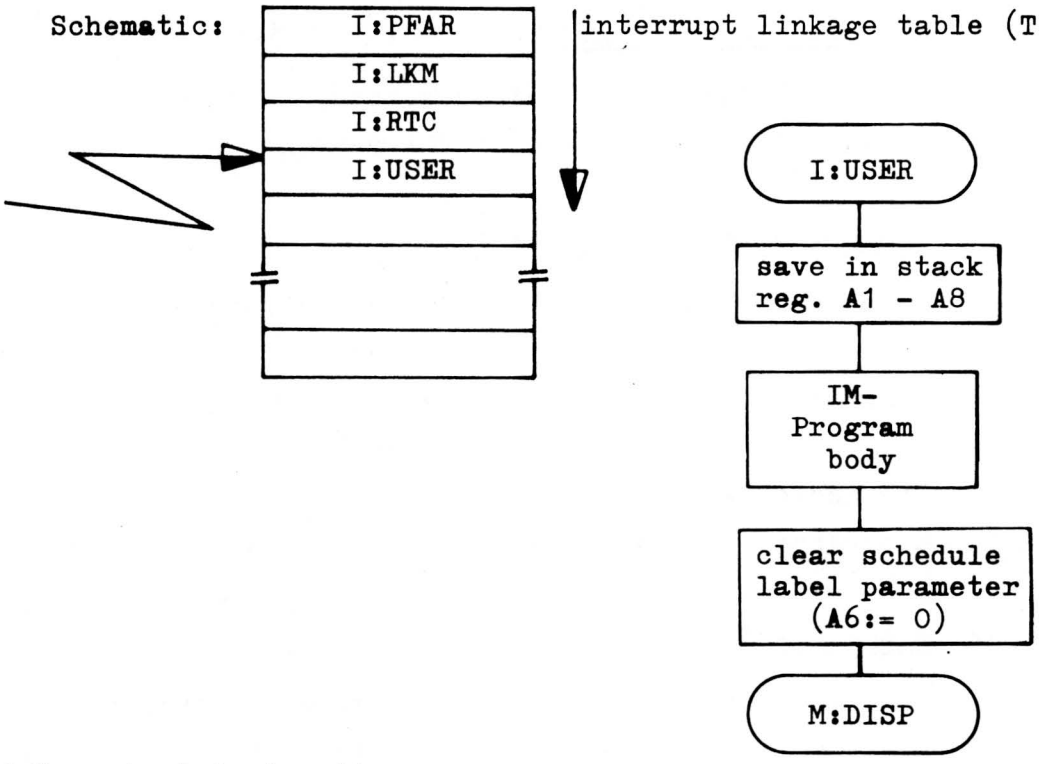
This table has to be loaded absolutely from loc. /0 to /5E. It has to contain the startaddresses of the IM programs connected to the various interrupt levels. Location /0 represents level 0, the other ones are successive.
The table as it has been constructed now can be updated according to hardware specifications (see configuration 6.1).

3.1.2 Stack Overflow

When an interrupt is raised, program counter, PSW and registers A1-A8 are stored in stack -A15. If this stack area is not big enough, an interrupt will be raised as soon as the stackpointer A15 exceeds the last location, i.e. /100. This interrupt is connected to the same line as the internal LKM. In the IM-program connected to this line, the stack overflow interrupt will cause a HLT.

3.1.3 Programming rules

-In IM-programs only registers A1 to A8 can be used.
-All IM-programs have to start by saving A1-A8 in stack A15.
-Each IM-program shall return to the dispatcher by ABL M:DISP. Before returning, the schedule label parameter in reg. A6 shall be cleared (BRTM compatability).



3.1.4 Non-wired instruction

If a non-wired instruction is executed, the so-called trap-action is started. This action:
- does not bother about inhibit mode
- has higher priority than any level
- stores P and PSW in A15-stack
- does not change level in PL-register
- puts CPU in inhibit mode
- performs indirect branch via loc. /7E to traproutine

The traproutine saves A1-A8 in A15-stack, checks stack overflow and prepares the parameters for the main simulation module labeled M:A00. In here the instruction is analysed and a simulation routine called for. At last return to user program. Re-entrancy of the main module is provided by a particular stack, labeled T:SVR, declared in T:SYS.

3.1.5 Power failure / automatic restart

Hardware

At power failure an interrupt is raised on line 0, the current PSW and PC are saved in stack A15 and a link is performed via loc. /0.
The last instruction of the following software action will be a halt.

The automatic restart option performs a master clear as soon as power is restored. Without saving anything into stack A15 a link is performed via loc. /0 (PL=63).

Software

Several actions are possible. The one implemented in this monitor is reinitialisation of the whole system.
Another solution, however rather complex, is to continue from the breakpoint.
It should be stressed that it is not possible to restart the system in such a way that the running I/O will continue from its breakpoint too, for all controllers were reset at master clear.
To avoid that programs, which requested I/O, will wait forever (I/O shall not be completed), it is necessary to perform a Reset Event for all requested I/O.

Moreover the automatic restart program may set a flag in the ECB-statusword to tell the user program that I/O request has failed because of power-failure. Of course this requires further action in the user program.

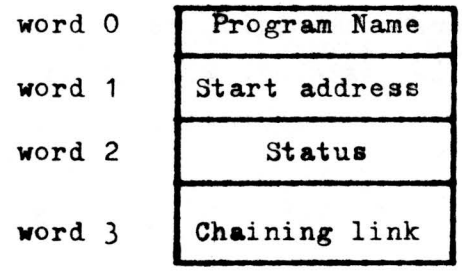
Special precautions shall be taken that a driver is not restarted in a critical section.

The restart measures shall not be performed than after switching to level 48.

3.2 Software Level Programs

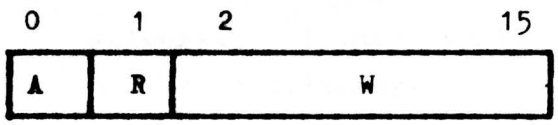
3.2.1 Program Control Table

The dispatcher uses the Program Control Table (PCT) containing all information about a software level program which has been loaded into memory. This table has the following layout:



where

word 0 contains the program name (2 ASCII char.)
word 1 contains the start-address of the program
word 2 contains the program status



active bit bit 0 = 1 program in active state
repeat bit bit 1 = 1 program requested in active state
wait bits bits 2-15 = address of Event Control Block program in wait State
word 3 contains a chaining link to the PCT of the next program on the same level

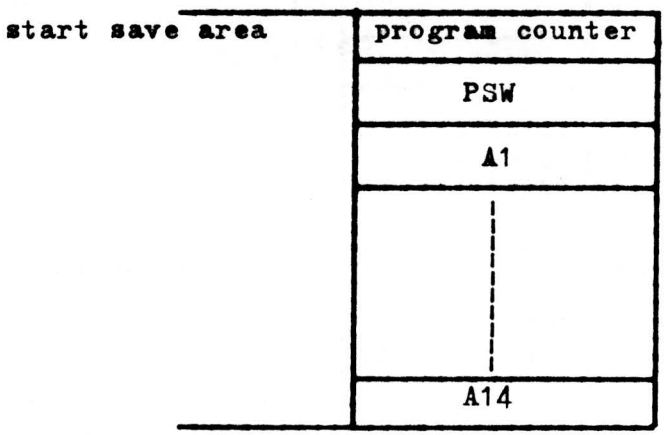
The PCT-status word address of one of the programs chained together is placed in the Software Level Table, thus identifying the level of these programs.

3.2.2 Software Level Table

The priority level of a Software Level Program is specified by the position of its PCT-status word address in the Software Level Table.
Each location in this table represents one of the levels 50-63, the first location being related to level 50, the other ones being conform.
The dispatcher modifies this location as to contain the PCT-status word address of that program on a level, which is active and ready to be executed.

3.2.3 Programming Rules

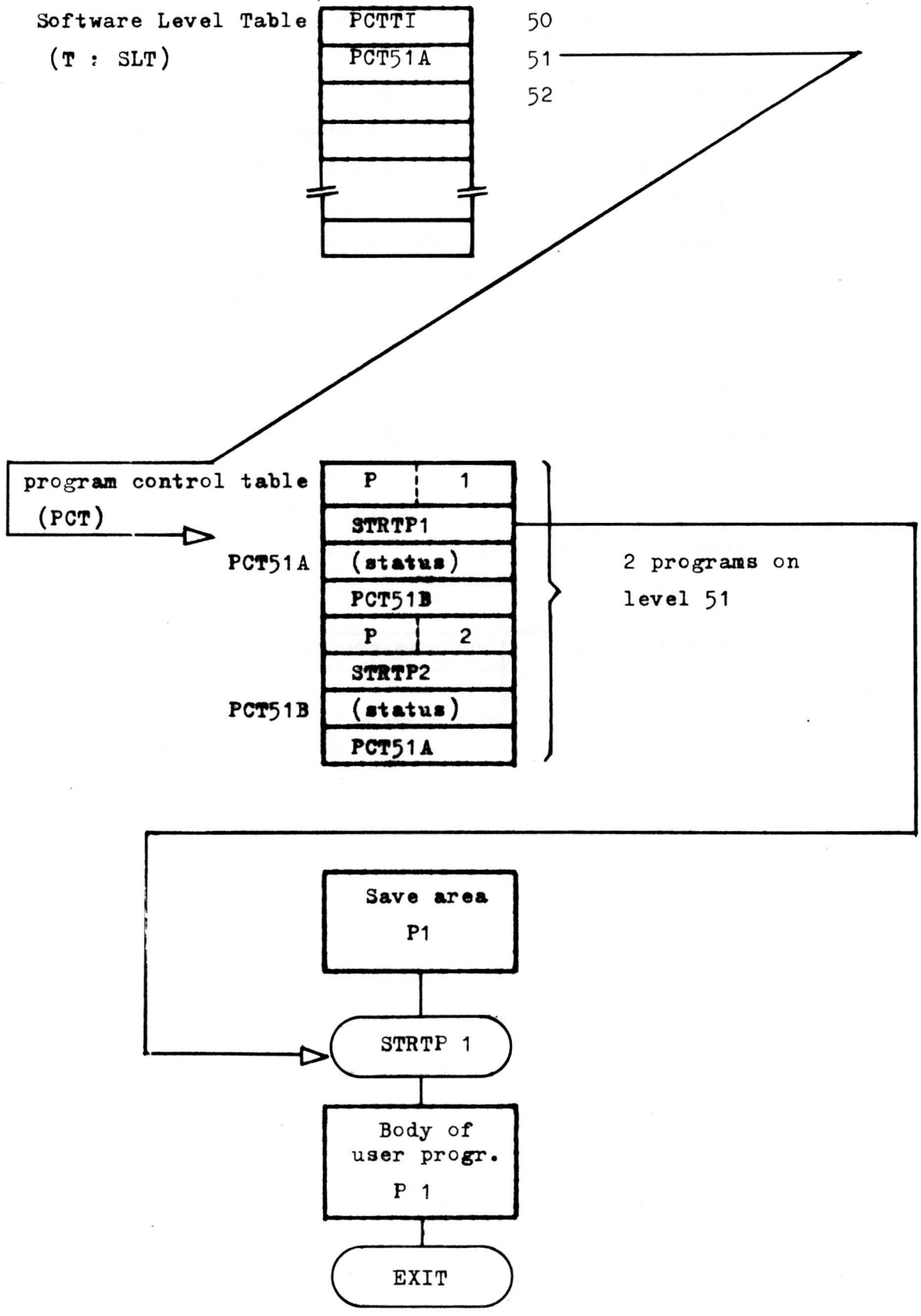
- A Software Level Program shall be preceeded by a Save Area, which length is 16 locations. This Save Area will be filled by the dispatcher when a program on a higher software level becomes active and is ready to be executed.
The contents will then be:



start user program (same loc. as defined in PCT)

- A Software Level Program shall finish with an Exit Monitor Request (see 4.2). This Exit shall only be given if all I/O events have been completed (see 4.6). So, if necessary a Wait Monitor Request (see 4.3) has to be given as to wait for I/O completion before an Exit is done.

Schematic


Software Level Table
(T : SLT)
PCTTI
PCT51A
50
51
52
program control table
(PCT)
P
1
STRTP1
PCT51A
(status)
PCT51B
P
2
STRTP2
PCT51B
(status)
PCT51A
2 programs on
level 51
Save area
P1
STRTP 1
Body of
user progr.
P 1
EXIT

3.2.4 Use of Subroutines

- Subroutines may be called by using A15 as stack-pointer, as well as any other register. However, in the latter case a separate stack shall be declared per software level program. It is not allowed that any other program uses the same stack.

- If a subroutine contains a Wait Monitor Request, it is forbidden to use A15 as stack-pointer. Another stack has to be declared, any register can serve as stack-pointer (see previous point).

- Re-entrant Subroutines

  Subroutines are re-entrant if:

  a) for intermediate data storage only registers are used, thus not any memory location.
  b) No instructions are composed internally, or composing and executing is performed in Inhibit Mode.

3.3 Timer structure

3.3.1 Scheduling

The timerfunction itself (activation of programs, updating of time and optionally updating of date) is scheduled by the monitor as a normal software level program. This program is activated by the Interrupt Mode part of the RTC-interrupt.

3.3.2 Timer-items

Each basic time interval has at its disposal an itembuffer, in which the timer-item is stored at the moment a connection is done.

One timer-item consists of:

- indication of program to be activated when time is due,
- offset (no. of counts after which the specified program is activated for the first time),
- interval (no. of counts between two activations).

3.3.3 Itembuffer identification block

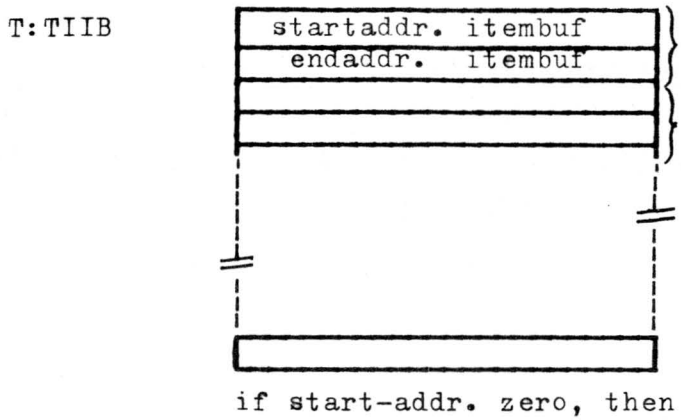
An itembuffer is identified by its start- and end-address in the itembuffer identification block. In this block start- and end-addresses are grouped following the time-interval codes.

3.3.4 Time-updating

To be able to update the time on each clock-pulse, a timer-interval table holds the counts and presets per interval. The preset is the negative number of counts before an interval is expirated. The counter is incremented from negative preset towards zero.

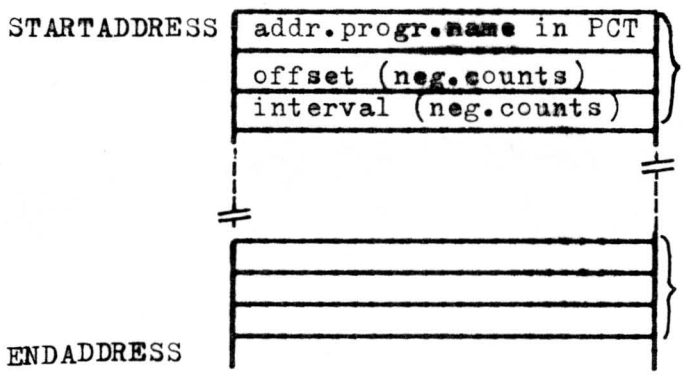
3.3.5 TABLE FORMATS

- ITEMBUFFER IDENTIFICATION BLOCK

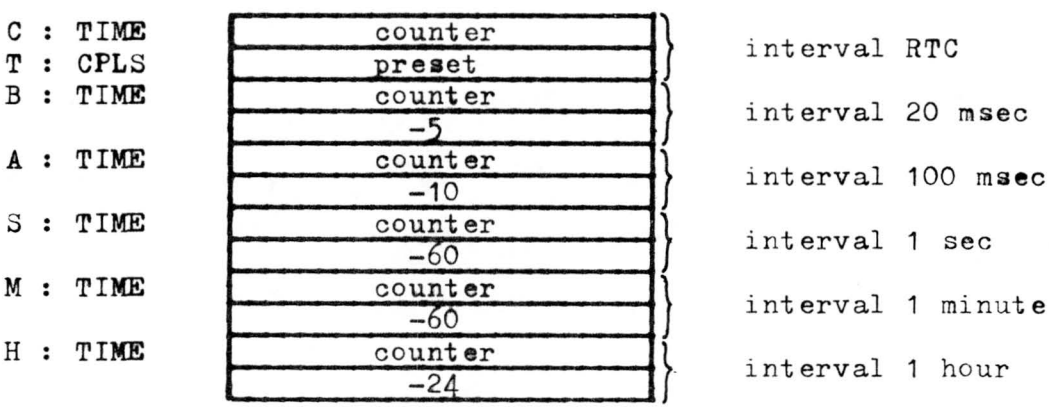


if start-addr. zero, then timer absent

- ITEMBUFFER



- TIMER INTERVALS



3.3.6 Date-updating

In an optional routine the date is updated.
This routine will be entered once a day.
The binary counts of day, month and year (to be initialized via the externals C : DAY, C : MONT and C : YEAR) are updated.
Hereafter the binary values which have been changed, are transcoded to 2 ASCII characters (obtainable via the externals W : DAY, W : MONT and W : YEAR).

4. MONITOR REQUESTS

Monitor requests are performed by the user program through a link to monitor (LKM) instruction. The nature of the requested function is specified by the word following the LKM, which is a numeric identifier (coded by a DATA directive). A7 and A8 registers are used to specify parameters.
Only the Activate MR may be used in Interrupt Mode Programs. The other ones are only allowed in Software Level Programs.

4.1. Activate a program

```
Format  LDKL   A7,  PRNAME
        LDKL   A8,  DUMMY
        LKM
        DATA   12
```

PRNAME points to the name in the PCT of the program to be activated. Register A8 points to a dummy parameter block of 2 words to be compatible with BRTM. This request can be made by a program running at any level, to activate a software level program. Depending on their priority levels, both the calling and activated program may be processed concurrently. If the program is already active the request is recorded in the status word (repeat bit). In case more requests occur only the first one is taken into account.

4.2 Exit

```
Format  LKM
        DATA   3
```

This request is used to specify the end of a software level program.

If another activation request is recorded for this program (repeat bit) the program remains active. Otherwise, the program becomes inactive, central processor control is given to the next active program on the same level. If no other program on this level is active, control is given to the next level.

4.3. Wait for an event

```
format      LDKL    A8, ECBADR
            LKM
            DATA    2
```

where ECBADR gives the address of the Event Control Block (see I/O request). The first bit of the ECB is the event bit. If this bit is set to 1, the event has been completed. This request causes a program to stop and wait for the completion of an event (I/O or user). If the event has occurred, the program is continued. If the event has not occurred the program is put in wait state, to be restarted when the event has occurred. Control is given to the next active program on the same level. See also 4.7.

4.4. Attach a device to a program

```
format      LDK     A7, 1
            LDKL    A8, DEVBLK
            LKM
            DATA    14
```

A7 should be loaded with constant "1" for compatability reasons.

DEVBLK points to a one word block containing the file code of the driver to be attached. It is allowed that DEVBLK refers to the file code in the first word of the ECB.

If the device requested for has already been attached (which is indicated by bit 0 = 0 in the attach-word of the DWT), the requesting program is put in wait state. This wait is performed on the just mentioned bit.
If the device is detached, the highest level program waiting for it will have control.

Att: If one of the ASR-functions (keyboard, reader, punch) is to be attached, all 3 functions should be attached, for they are all performed on the same device.

4.5 Detach a device from a program

```
format      LDKL    A8, DEVBLK
            LKM
            DATA    15
```

DEVBLK is specified as under attach (4.4.)

Use: By means of this request a device which has been previously attached to a program through an attach request, is detached from a program.

4.6 I/O request

```
format      LDK     A7, ORDER
            LDKL    A8, ECBADR
            LKM
            DATA     1
```

Through this request the user can ask the system to start a certain I/O operation on a peripheral device. The parameter in the A7 register has the following meaning:

ORDER specifies which I/O function is requested, by giving one of the following combinations of 2 hexadecimal digits:

01 basic read
02 standard read
05 basic write

For each of these request orders, the following specific information applies to the various peripheral devices:

1. Basic read (01)

operators' typewriter: all characters are entered on 8 bits until the requested length is reached.

ASR tape reader : same as for the keyboard. The reader stops one character after an X-off code is read.

high speed reader : all characters are entered on 8 bits, without checking or special features, until the requested length is reached or end of tape is encountered.

2. standard read (02)

operators' typewriter: ASCII characters are entered on 7 bits, with the following special features:

- the special characters, coded from X'0' to X'1F' are ignored
- the code X'7F' (rub-out) is ignored
- code X'5F' (←) can be used to delete the proceding character. If more ← are used consecutively, an equal number of preceding characters will be deleted
- code X'5E' (↑) is used to delete the line preceding it, up to the next carriage return.
- for reasons of compatability with BRTM, input character "\" shall not be used.

- **all characters entered above the req. length are ignored.**
- code X'0D' (carriage return) indicates end of block. It is the last character to be entered. It is not transmitted to the users' buffer.

ASR tape reader : For ASII characters, the same features apply as for the keyboard. The code for carriage return must be preceeded by the code X-off.

high speed reader : same as for ASR tape reader.

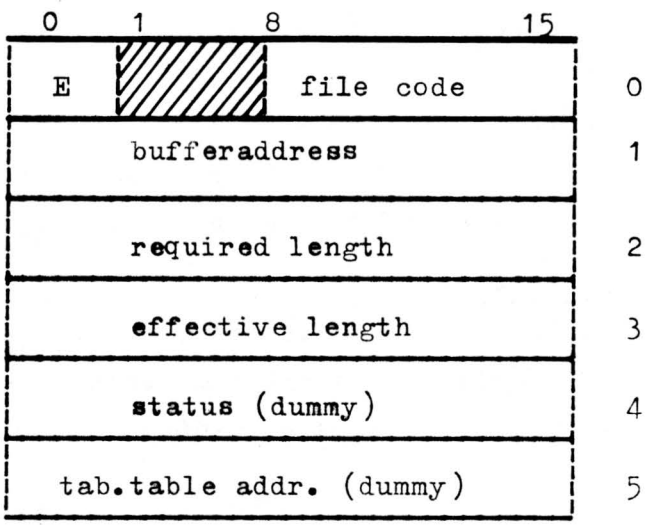
3. Basic write (05)

operators' typewriter: all characters are output without checking or special features.

ASR tape punch : same as for printed output.

high speed punch : same as for ASR tape punch.

4.6.1 The Event Control Block (of which the address must have been loaded in the A8 register) has the following format:



word 0 bit 0 = 1 end of operation has occurred for this ECB (event bit).

bit 8 - 15 = file code.

file codes are assigned to the various devices to provide for distinguishing and addressing them (see 4.6.2).

word 1 address of the user buffer.

word 2 requested length to be read or written (in characters)

word 3 effective length which has been transmitted (in characters)

word 4 statusbits, dummy

word 5 tabulation table address, dummy

Note: 1. A return to the calling program will be made directly after initiation of the device. The user program can give a Wait monitor request later on for synchronisation.

2. If the device is busy, the calling program is automatically put into "Wait for an event" state, until the device is given free. Control is given to the next active program on the same level.

3. All ECB's should be declared in the memory part <16K. This is because at a Wait Monitor Request the ECB-address is filled into the status-word of the calling program, bits 2-15, after shifting this address one position to the right.

4.6.2 File Codes

The file code consists of two hexadecimal characters.
It refers to a device via the:

File Code Table

This table connects a device work table (in fact a device) to each file code. The start-addresses of the DWT's have been placed in the file code table according to the corresponding file code.

Th The following adjudgement is standard:

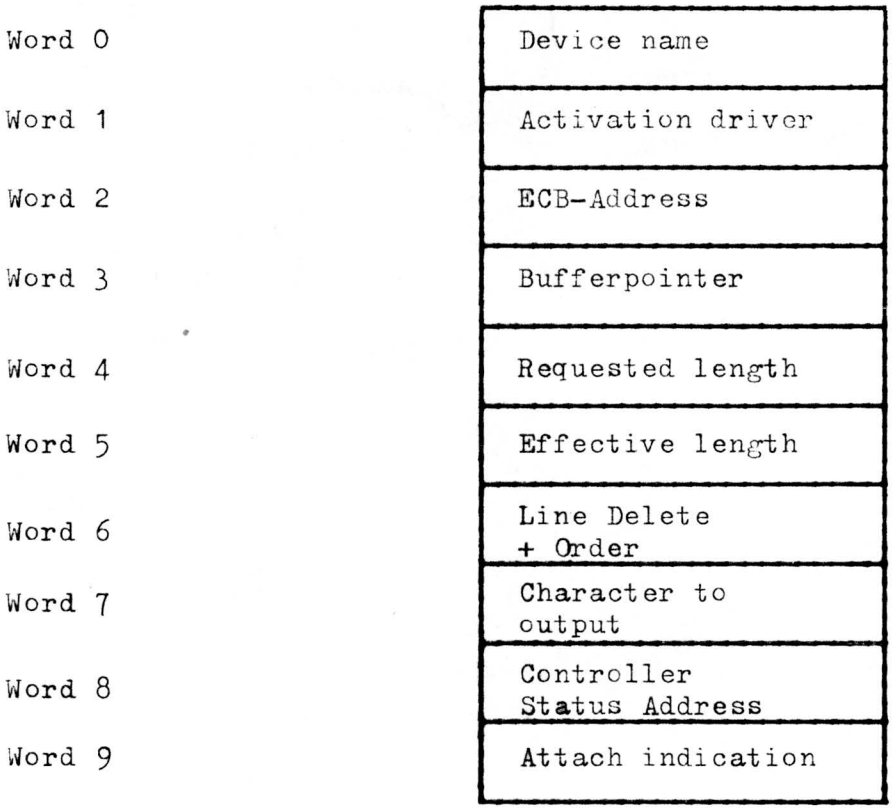
file code:	01	reserved for standard utility programs, which cannot be executed with SRTM.
	02	
	03	
	04	
	05	ASR keyboard
	06	ASR reader
	07	ASR punch
	08	paper tape reader
	09	paper tape punch

4022 151 04001 bl. 625-27

4.6.3 Device Work Table

The start-address of each device work-table is given by the file code in the first word of the ECB, which is pointing to a location in the file code table.

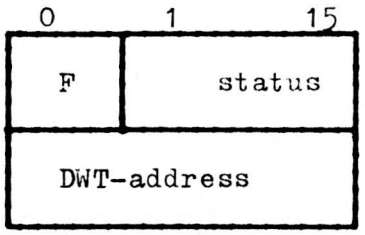
Layout:



Where:

word 0 contains the device name (2 characters ASCII)
word 1 contains the start-address of the driver
word 2 contains the address of the event control block of the calling program
word 3 contains the start-address of the buffer to be handled
word 4 contains the number of characters to be handled
word 5 contains the number of characters which are handled
word 6 bit 0 = 1 : line delete character has been given
bits 13 - 15 : order of the device-handling
word 7 contains the next character to be output

word 8 contains the address of the controller status-block



First blockword:

bit 0 = 1 : controller free
= 0 : controller busy
bits 1 - 15 : statusbits for reader handling

Second blockword:

DWT - addr. : address of the DWT operating now.

word 9 bit 0 = 0 device attached
= 1 device detached
bits 1-15 PCT-address of program to which device has been attached.

4.7 Reset an event

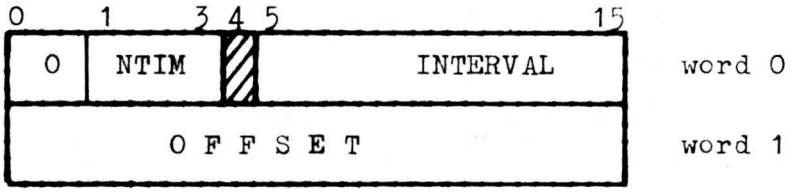
format : LDKL A8, ECBADR
LKM
DATA 18

ECBADR gives the address of a one-word ECB. The left most bit in this word is set to 1 as to indicate the completion of the event.
This feature can be used in cooperation with the Wait monitor request as to test and set inter-program flags.
Setting the flag on which other programs are waiting has to be done by the Reset Event monitor request for BRTM-compatability reasons.

4.8 Connect Timer

format : LDKL A7, <addr. programname in PCT>
LDKL A8, <addr. parameterblock>
LKM
DATA 10
on return: A7 = 0 connection succeeded
A7 ≠ 0 connection not succeeded

4.8.1 The parameterblock has the following format:



word 0 : bit 0 : shall be zero for compatibility reasons with BRTM-timerfunction
bits 1-3 : timernumber, specifying the basic time interval in which counts of offset and interval are expressed (see 4.8.2.)
bits 5-15 : interval, specifying the number of counts in a basic time interval between two consequetive activations.
word 1 : offset, specifying the number of counts in a basic time interval before the first activation.

4.8.2 The basic time intervals are coded as follows:

standard	non-stnd.	
	0	RTC pulse rate (no. of cycles during 20 msec)
0	1	standard pulse rate (20 msec)
1	2	100 msec
2	3	1 second
3	4	1 minute
4	5	1 hour

4.8.3 The one-shot, performing a unique activation of the specified program, can be initiated by setting:

INTERVAL : = 0

After expiration of the specified offset, the program is activated and removed from the timer.

4.9 Disconnect Timer

format: LDKL A7, <addr. programname in PCT>
LDKL A8, <timernumber>
LKM
DATA 11

The timernumber shall be equal to the one with which the connection was done. **On return: A7 = 0 disconnected**
A7 ≠ 0 disconnection impossible

4.10 Get time

4.10.1 Preset

The absolute time in seconds, minutes and hours shall be preset by f.i. a Keyboard Function.
The syntax shall be like this format:

ET ␣ <hour> , <minute> , <second>

4.10.2 Get time

The momentary absolute time can be fetched at any moment, using the following external labels:

S : TIME for complementary counts of seconds.
absolute seconds : (S:TIME)+ 60.
M : TIME for complementary counts of minutes.
absolute minutes : (M:TIME)+ 60.
H : TIME for complementary counts of hours.
absolute hours : (H:TIME)+ 24.

4.11 Get date.

4.11.1 Preset.

The date in days, months and years shall be preset by f.i. a Keyboard Function. This function has to fill counters C:DAY, C:MONT and C:YEAR with the date of yesterday and then call for the date- update routine M:DATE.

The syntax shall be like this format:

ED ␣ <year>,<month>,<day>.

4.11.2 **Get date.**

The daily date can at any moment be obtained via the following external labels:

W : DAY 2 ASCII characters for day-number
W : MONT 2 ASCII characters for month-number
W : YEAR 2 ASCII characters for year-number.

5. INITIALISATION OF MONITOR AND USER TASKS

The Monitor is initialised by module M:INIT.
To initialise the User Tasks, the system start-up is activated, which is a user SL-program.

System start-up

This SL-program has been connected to level 50. It has to be written by the user, and can contain f.i.:

- call for initialisation of timer (U:INTI)
- activations of SL-programs
- connect timer requests
- calls for initialisation routines of user tasks.

6. CONFIGURATION.

To be as flexible as possible, the monitor has been subdivided in 31 modules. Each of this modules can either be updated or rejected according to the users' needs.

If a module is rejected, it is advisable to satisfy the Linkage Editor by labeling the name of the rejected module to a dummy location. This location can be physically the same for all rejected modules.

6.1 System Tables

Identifier : T:SYS

In this module the following tables are included:

6.1.1 Interrupt Linkage Table

Identifier : T:INT

This table has already been constructed for the standard interrupt lines.

6.1.2 Software Level Table

Identifier : T:SLT

This table shall be made individually for each system, taking into account the following rules:

- level 50 shall always be used (for monitor initialisation purposes). Standard : system start-up and timerfunction.
- startaddress shall be labeled by T:SLT, endaddress by T:SLTE, both labels to be declared in the entry list,
- linkages in this table shall point to the statusword in the PCT of the SL-program (external),
- if a level is not used, the linkage shall be zero.

6.1.3 Trap location and simulation storage

Identifier : T:SRA / T:SRV

The linkage table for simulation (T:SRA) is preceeded by the trap location, which has been fixed on loc. /7E.

The save area, used by the simulation main module M:A00, has a depth of two levels (T:SVR). If more levels use simulation, the save area can be extended. Mind the following points:

- save area shall be extracted from its position in front of the stack area and put elsewhere.
  The released positions have to be filled up by a RES instruction, in order to hold the stack area in the right place.
- startaddress shall be lebeled by T:SVR, tablelength by L:TSVR. These labels shall be declared in the entry list.

6.1.4 A15-stack Area

Identifier : STB

The stack-area has been declared at a fixed place, because its endaddress has to be loc. /100.

While extending mind that:

- the startaddress, labeled by STB, corresponds to a memory-address higher than /100,
- the length of the stack shall be at least ten times the number of interrupt lines present in the system.

6.2 Program Control Tables

Identifier : T : PCT

For each Software Level Program in the system, a PCT shall be built. As has already been described (3.2.1) one PCT consists of

word 0 : program name (2 ASCII - char)
word 1 : start-addr. progr. (external)
word 2 : status
word 3 : chaining link

The following rules hold :

- the start-address of the block containing all PCT's shall be labeled by T : PCT, the end-address by T:PCTE. These labels shall also be declared in the entry list.
- the labels on the programname and status-word shall be declared in the entry-list.
- if the program is the last one on a level, its chaining link shall point to the first program on this level (closed loop). So if there is only one program on a level, its chaining link shall point to its own PCT.
  Att: the chaining link shall point to label on the status-word.

6.3 File Code Table.

Identifier : F : CT

This table has already been built for the standard file codes. If it has to be extended, mind the following points:

- start-address shall be the first address of the table minus 2. It shall be labeled by F : CT, the end-address by F : CTEN. Both labels shall be declared in the entry-list.
- the linkages in this table, pointing to the Device Work Tables of the corresponding drivers shall be declared in the external-list.

6.4 Device Work Tables.

Identifier : D : WT

The DWT's for the standard drivers have already been made. The layout has been given in 4.6.3. If the block of DWT's has to be extended, mind the following points:

- contents of the DWT:

  word 0 : device name; shall consist of 2 ASCII characters.
  word 1 : the start-address of the driver; shall be declared in the external list.
  word 2 to word 7 : not to be filled in.
  word 8 : controller status-address; shall be declared in the external list. If a new DWT is added, also a new CSB shall be made (6.7).
  word 9 : not to be filled in.

- start-address of the block of DWT's shall be labeled by D : WT, end-address by D : WTEN.
  Both labels shall be declared in the entrylist.
- start-address of each DWT shall be labeled by the name used in the file code table.
  These labels shall be declared in the entry list.

6.5 Controller Status Blocks.

For each driver (controller) two more locations shall be reserved to enable a check on controller free status.
For the standard drivers the CSB's have already been made.
If a driver is added, a new CSB shall be added to the block of CSB's. Mind that:

- the start-address of the block of CBS's shall be labeled by C : NSTB, the end-address by C : NSTE. Both labels shall be declared in the entry list.
- the first address of the CSB shall be labeled by the name used in the DWT (word 8). This label shall be declared in the entry list.
- both reserved locations need to be filled in.

6.6 Standard Read

Identifier : ASCINP

In case no standard read (order 2) is performed in the system, the extension of subroutine INPUT, handling the standard read order : ASCINP, is obsolete and may not be loaded.

6.7 Timer function

Before loading can be done, the user has to provide for:

1 - the RTC-interrupt linkage. The label I : RTC shall be filled in the RTC-interrupt position of the interrupt linkage table T : INT.

2 - the connection of the non-interrupt mode part of the timer-function to a software level.
This can be done by adding the PCT of the timerfunction to the PCT-pool and filling the PCT-address into the software-level table on the level-appropriate position.

3 - for each basic time interval an itembuffer, reserving 3 locations for each item position.

4 - for each itembuffer an identification, specifying start- and endaddress of the itembuffer.

5 - the RTC-pulse rate (minus no. of cycles during 20 msec) in T : CPLS.

For the standard timerfunction, all these specifications have already been made.
This means that:

1 - RTC-interrupt is on level 2,

2 - timer-function is on level 50 (chained to system start-up),

3 - 3 itempositions in each itembuffer,

4 - all basic-time intervals have an itembuffer,

5 - RTC pulse rate is 20 msec (T : CPLS = -1).

Specification:		to make in module:	
	1		T : INT
	2		T : PCT
			T : SLT
	3		T : TAB
	4		T : TAB
	5		T : TAB

In case date-updating is not required, the dummy date-update routine shall be selected from the object library, in stead of the real update routine.

7. SYSTEM GENERATION UNDER BOM

To generate a Load Module, which can be loaded by the IPL, the following steps shall be made consecutively:

7.1 Configurate and assemble all user Interrupt Mode Programs and Software Level Programs as to obtain the object tapes.

7.2 Update (if necessary) the Monitor Tables and do an assembly as to obtain the object tapes.

7.3 Perform a Link Editing b.m.o. utility program LKE.
The object modules to be linked shall be selected from the object library tapes b.m.o. command S <symbol> .
The first module to be included shall be T : SYS, as this shall be loaded from loc. / 00.

8. SOURCE LIBRARY

The following datatables can be updated according to the users' needs:

T : SYS, system tables,
T : PCT, program control tables,
F : CT, file code table,
D : WT, device work tables,
C : NSTB, controller status blocks,
T : TAB, timer tables.

MEMORY OCCUPATION:
The maximum size of the SRTM, including average datatables, is about 2K.

9. OBJECT LIBRARY

The following survey of the SRTM is related to the object modules of the monitor. The sequence is the same as on the object library tape.

T : LKM,	LKM linkage table
I : TRAP,	trap action routine
I : PFAR,	power failure/automatic restart
I : LKM,	LKM processor
M : AOO,	simulation instr. analysis
MPYMOD,	simulation multiply
DIVMOD,	simulation divide
ADDMOD,	simulation double add
DSUMOD,	simulation double subtract
DSHMOD,	simulation double shift
MLDMST,	simulation multiple load/store
M : INIT,	monitor initialisation
M : DISP,	dispatcher
M : IØRM,	I/O request module
D : RPTR,	driver PTR
D : RDTP,	driver PTP
D : RAS1	driver ASR-keyboard
D : RAS2,	driver ASR-reader
D : RAS3,	driver ASR-punch
I : ASR,	IM ASR drivers
I : NPUT,	basic input
ASCINP,	standard input
Ø : TPUT,	basic output
R : TURN,	finish I/O request
M : WAIT,	wait MR
M : EXIT,	exit MR
M : ACT,	activate MR
M : ATDV,	attach device MR
M : DTDV,	detach device MR
M : RSEV,	reset event MR
FINDWT,	find DWT to file code
M : TIMR,	timerfunction, IM and NIM part
M : CNTM,	connect timer MR
M : DNTM,	disconnect timer MR
M : DATE,	date update routine
M : DATD,	dummy "

10. COMPATABILITY WITH OTHER MONITORS

10.1 BRTM

The SRTM is upward compatible with the BRTM. This means that user programs running under SRTM can be used without any internal modification under the BRTM.

10.2 DRTM

The SRTM is upward compatible with the DRTM, as far as user program written under SRTM are specified as core resident programs or background programs.

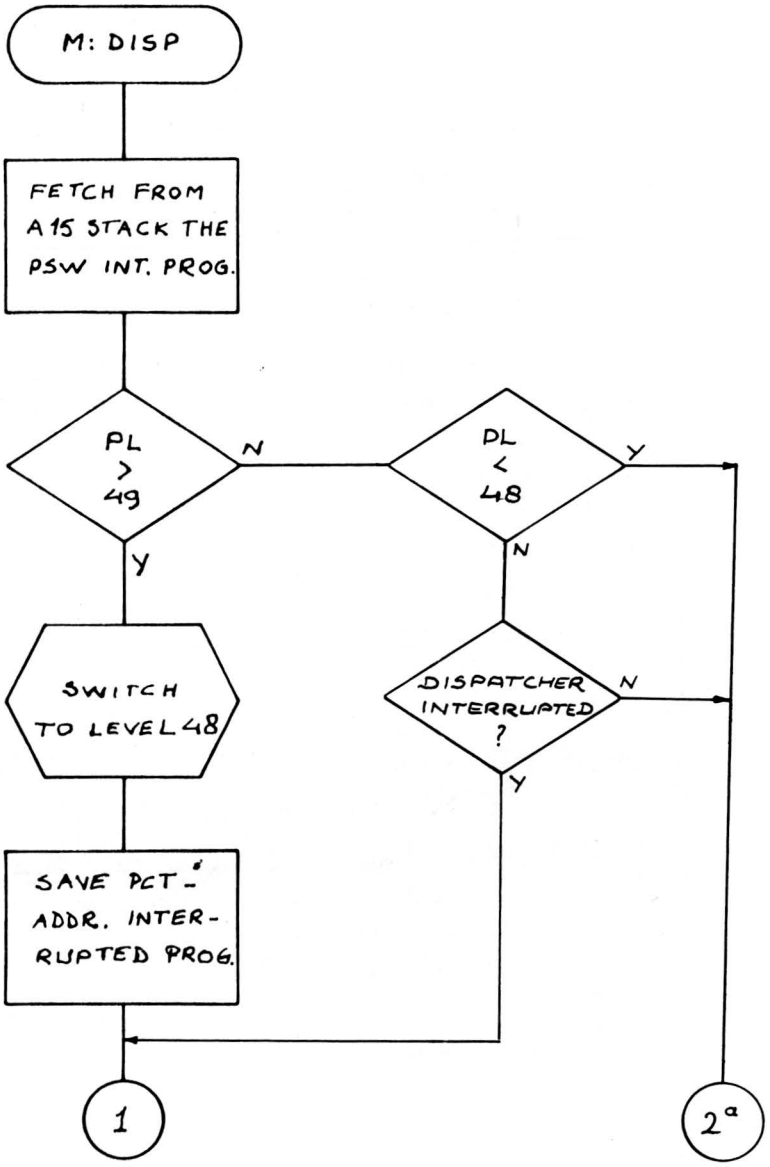
10.3 FORTRAN

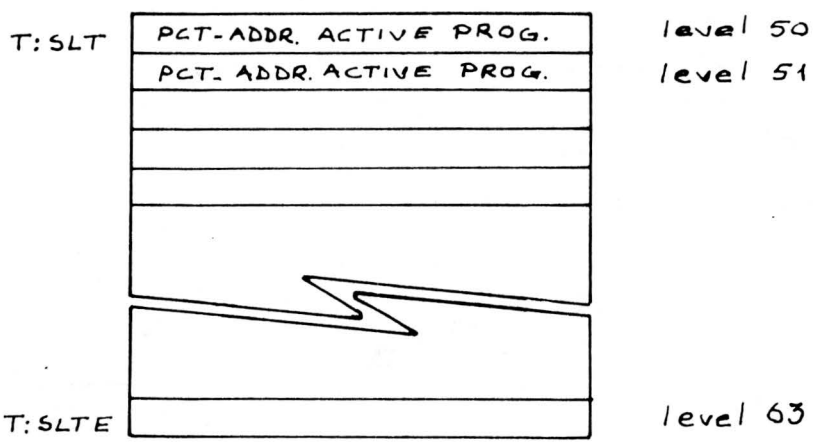
Programs written in Fortran cannot be executed with the SRTM, but require the BRTM or DRTM.

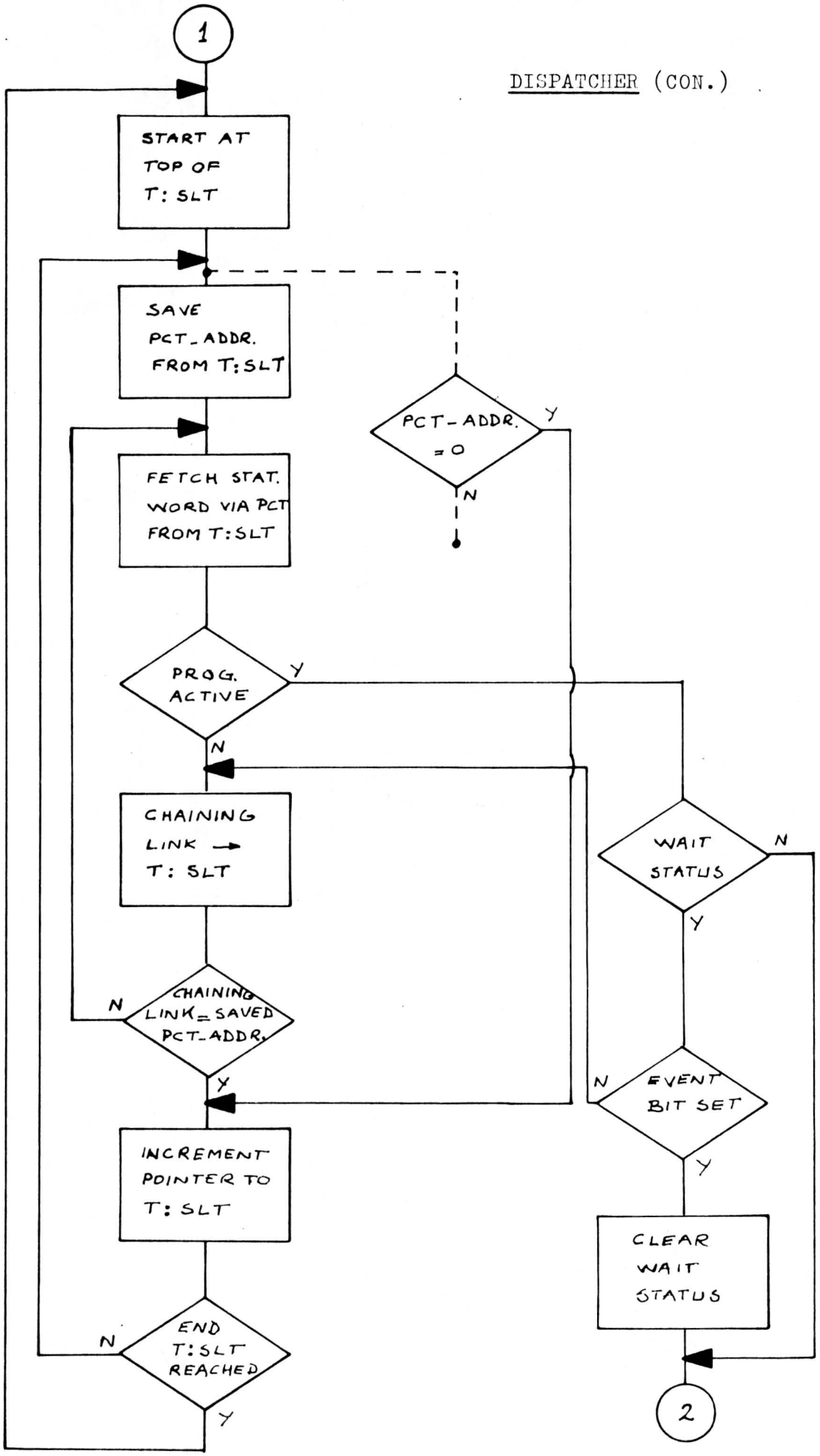
11. BLOCK DIAGRAMS

11.1 Contents

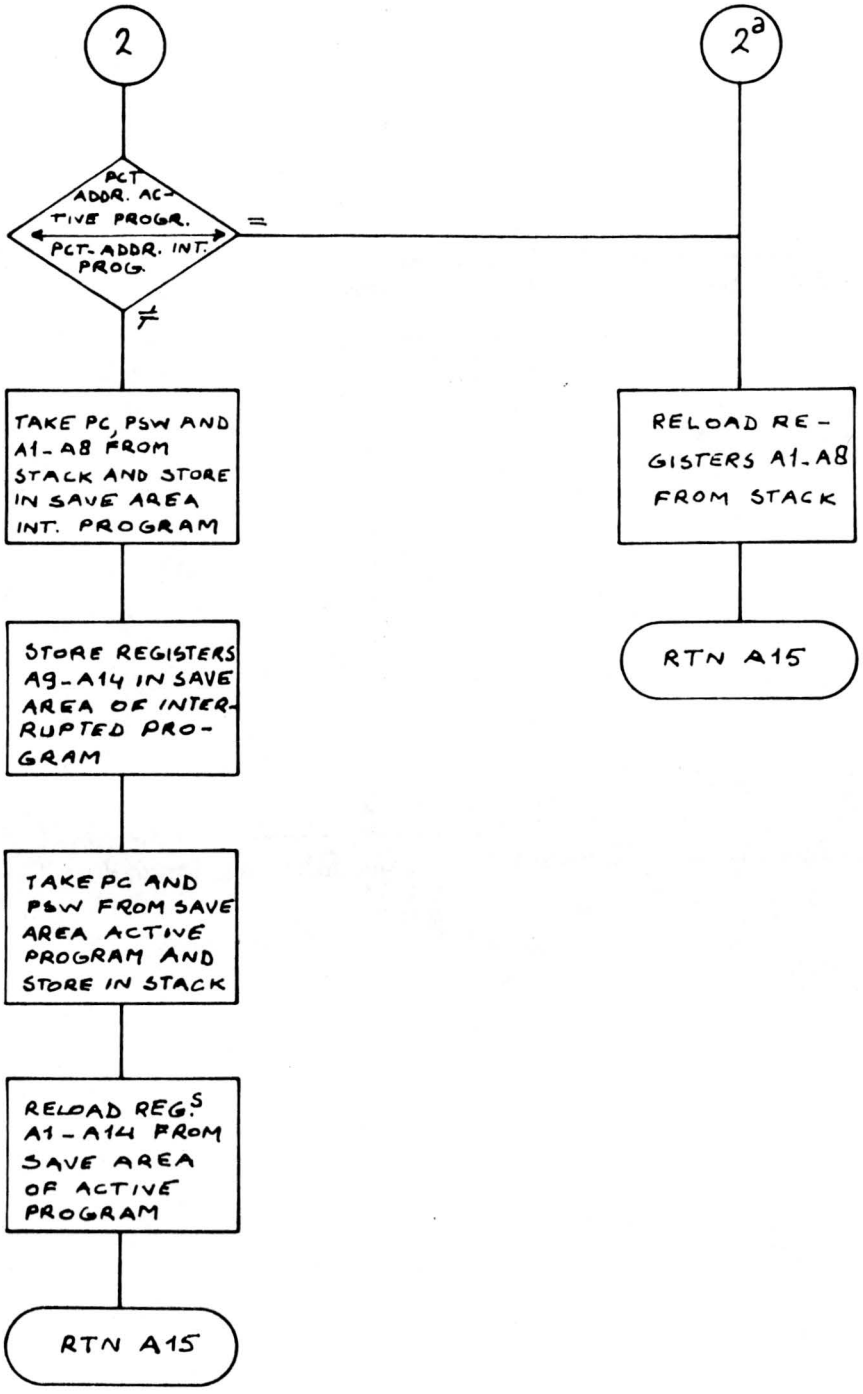
2. DISPATCHER



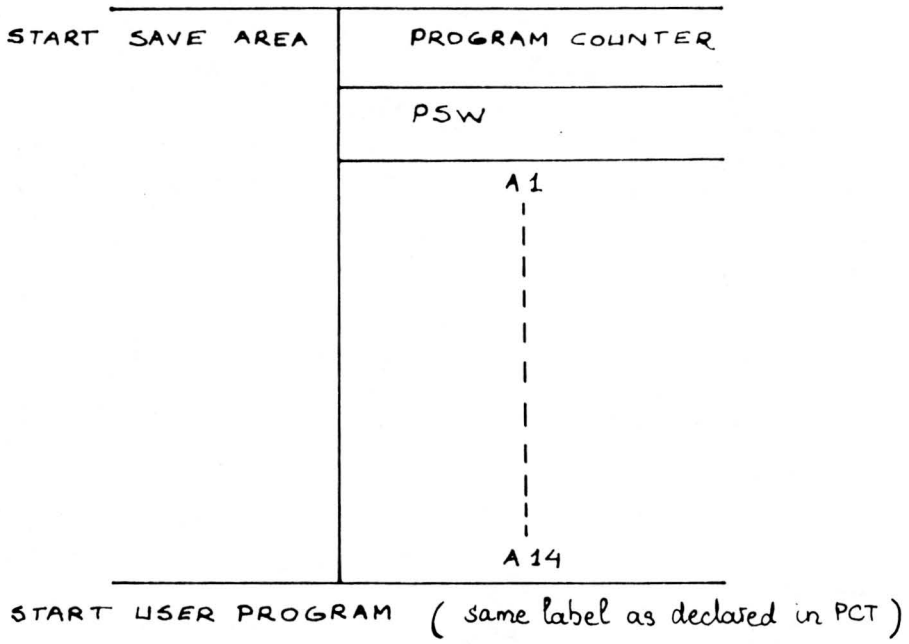



1
START AT TOP OF T: SLT
SAVE PCT_ADDR. FROM T:SLT
PCT-ADDR. =0
Y
N
FETCH STAT. WORD VIA PCT FROM T:SLT
PROG. ACTIVE
Y
N
CHAINING LINK → T: SLT
WAIT STATUS
N
Y
CHAINING LINK = SAVED PCT-ADDR.
N
Y
EVENT BIT SET
N
Y
INCREMENT POINTER TO T: SLT
CLEAR WAIT STATUS
END T:SLT REACHED
N
Y
2

DISPATCHER (CON.)



DISPATCHER (CONT.)
CONSTRUCTION SAVE AREA

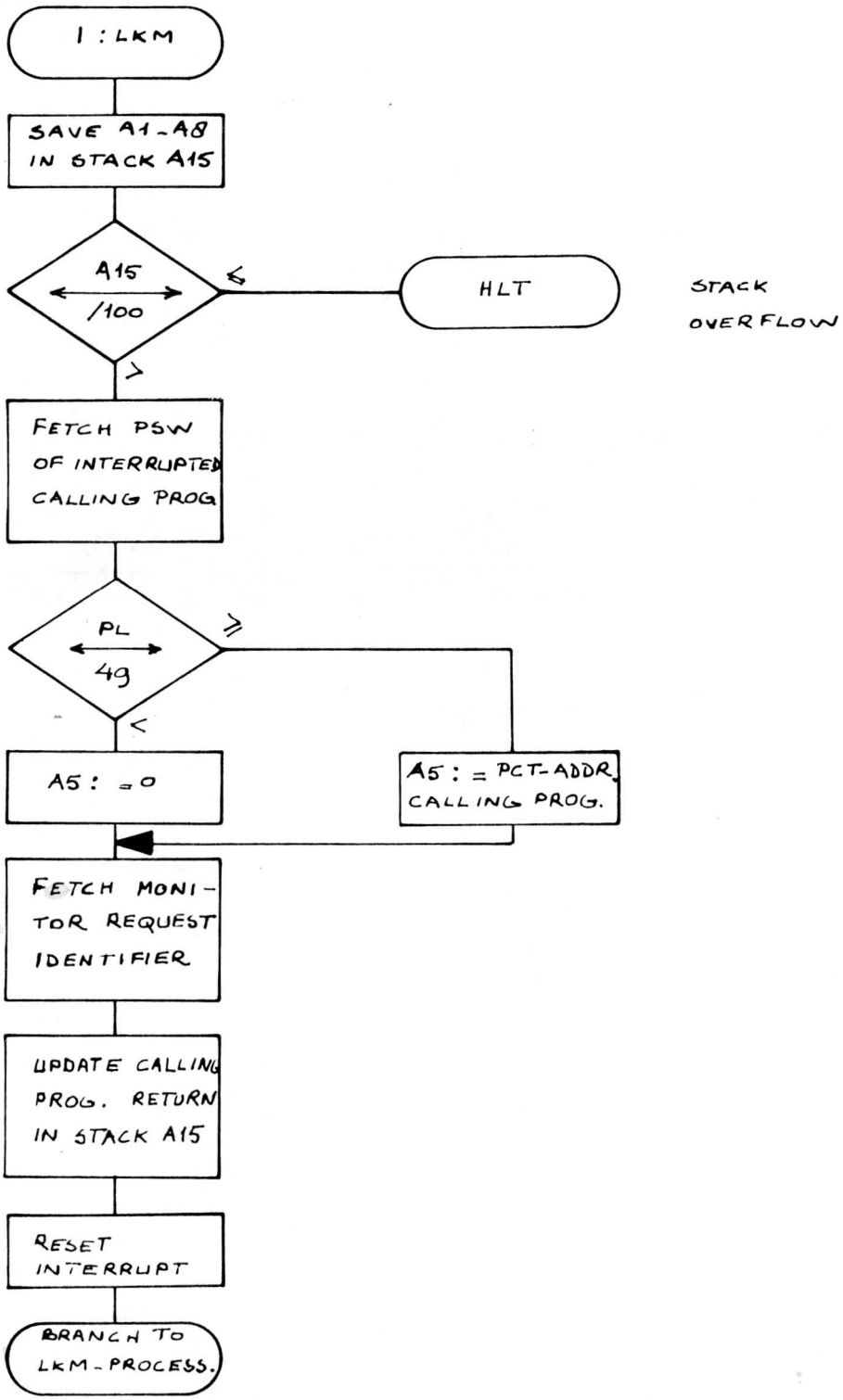


4022 151 04001 bl. 625-48

3. INTERRUPT HANDLING LKM

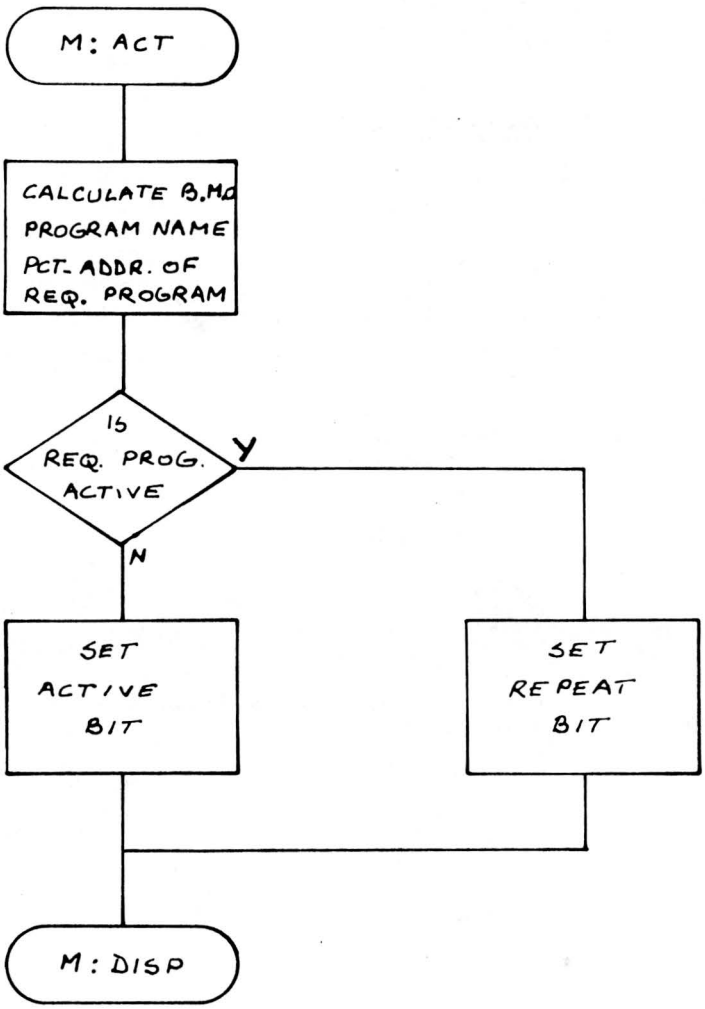
ENTRY: A7: PARAMETER FOR LKM - PROCESSOR
A8: PARAMETER FOR LKM - PROCESSOR

EXIT: A5: PCT-ADDRESS OF CALLING PROGRAM IF PL ≥ 49
ZERO IF PL < 49
A7: PARAMETER FOR LKM - PROCESSOR (UNCHANGED)
A8: PARAMETER FOR LKM - PROCESSOR (UNCHANGED)

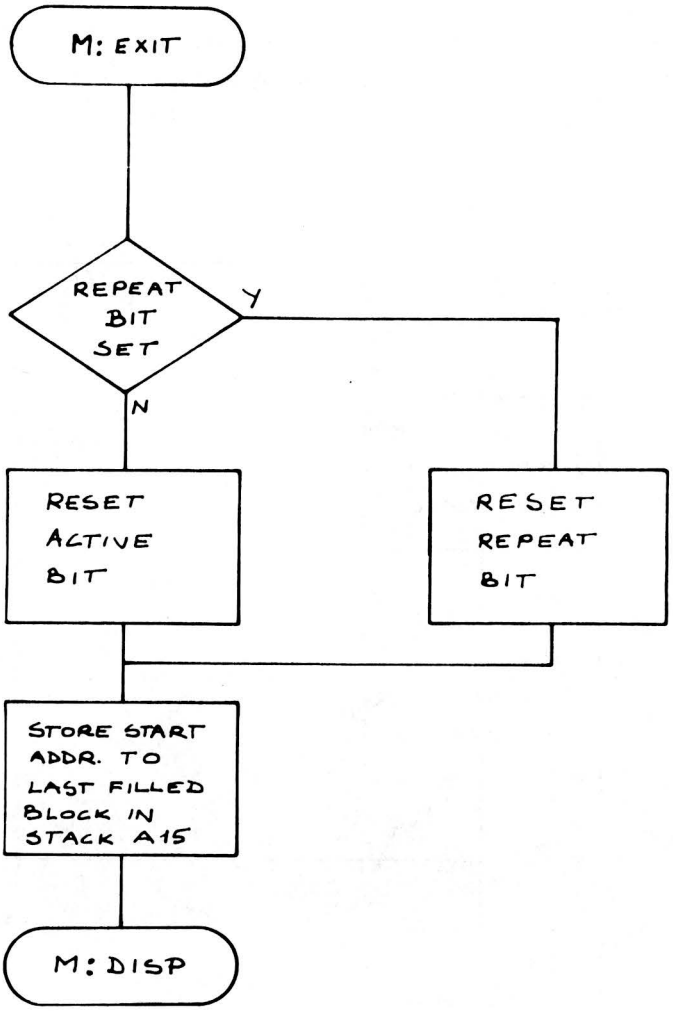


4. ACTIVATE MONITOR REQUEST

ENTRY : A5: PCT-ADDRESS OF CALLING PROGRAM IF PL ≥ 49
ZERO IF PL < 49
A7: ADDRESS OF REQUESTED PROGRAM NAME BLOCK IN PCT
A8: DUMMY ECB-ADDRESS (COMP. REQ.)

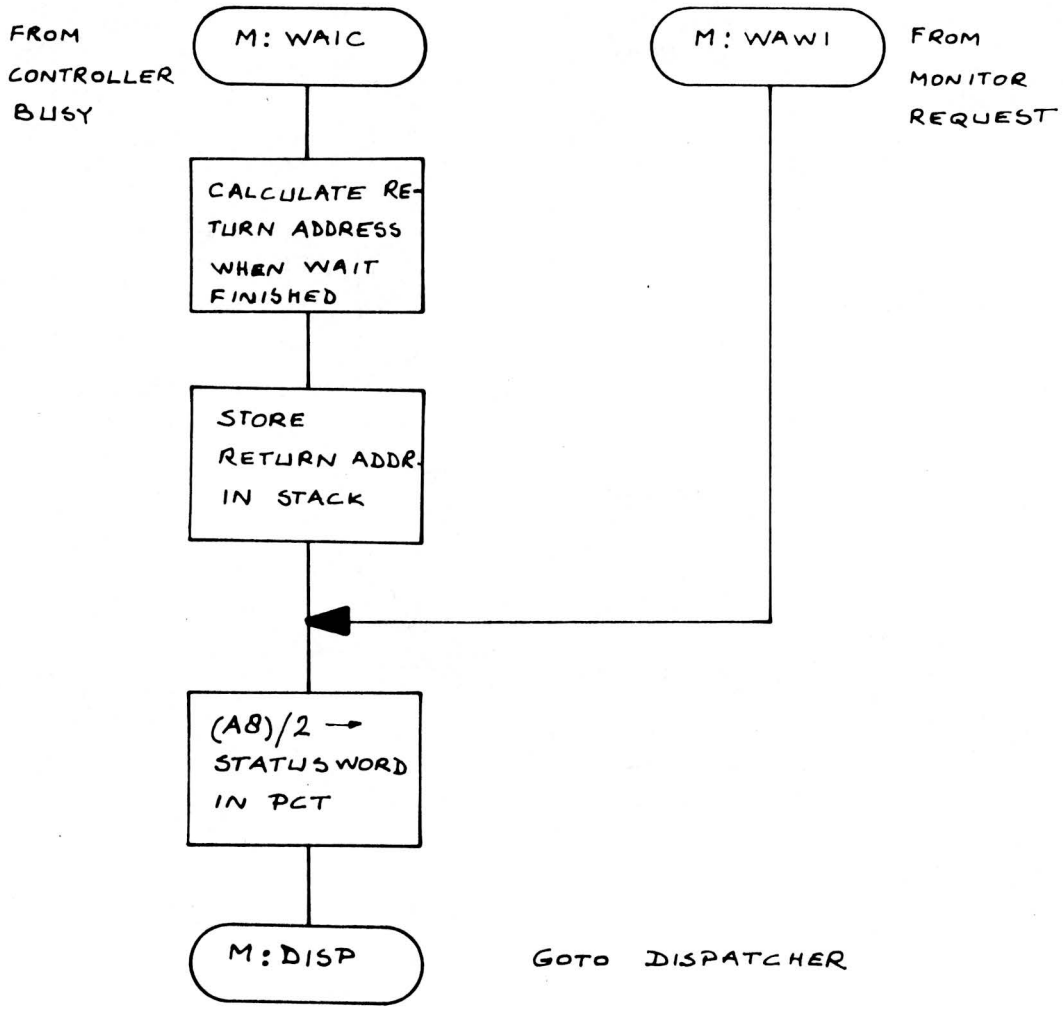


5. EXIT MONITOR REQUEST



6. WAIT MONITOR REQUEST

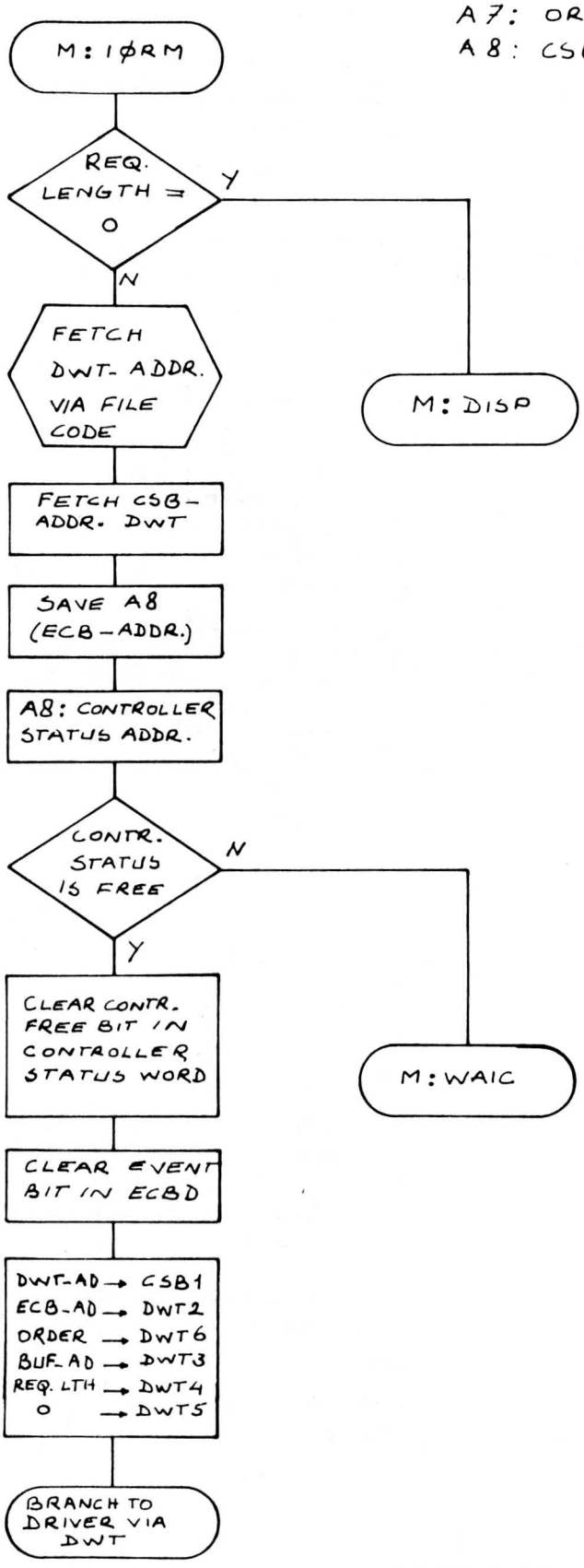
ENTRY: A5: PCT-ADDRESS CALLING PROGRAM
A8: ECB-ADDRESS



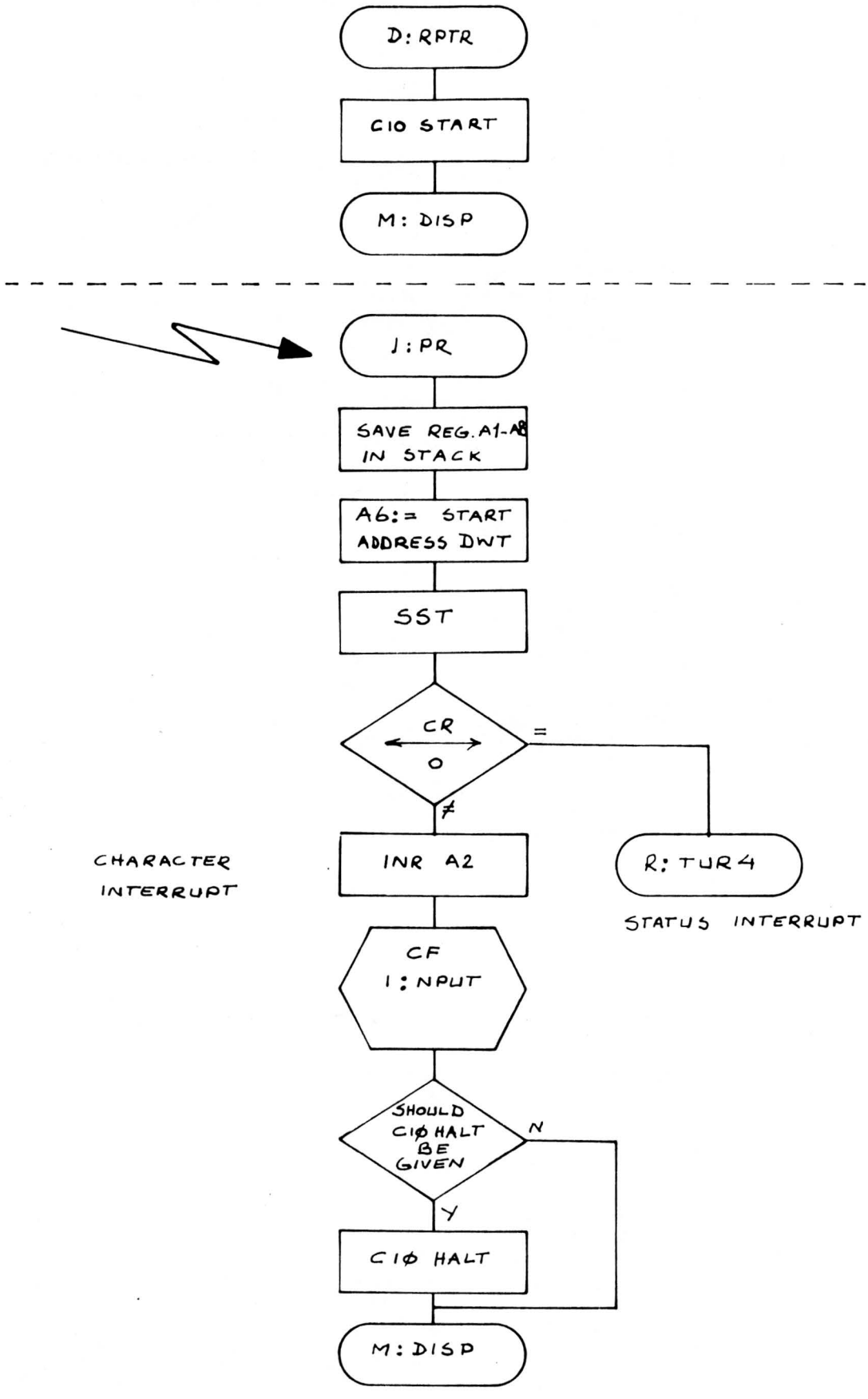
7. INPUT/OUTPUT MONITOR REQUEST

ENTRY : A5: PCT-ADDRESS CALLING PROGRAM
A7: ORDER
A8: ECB-ADDRESS

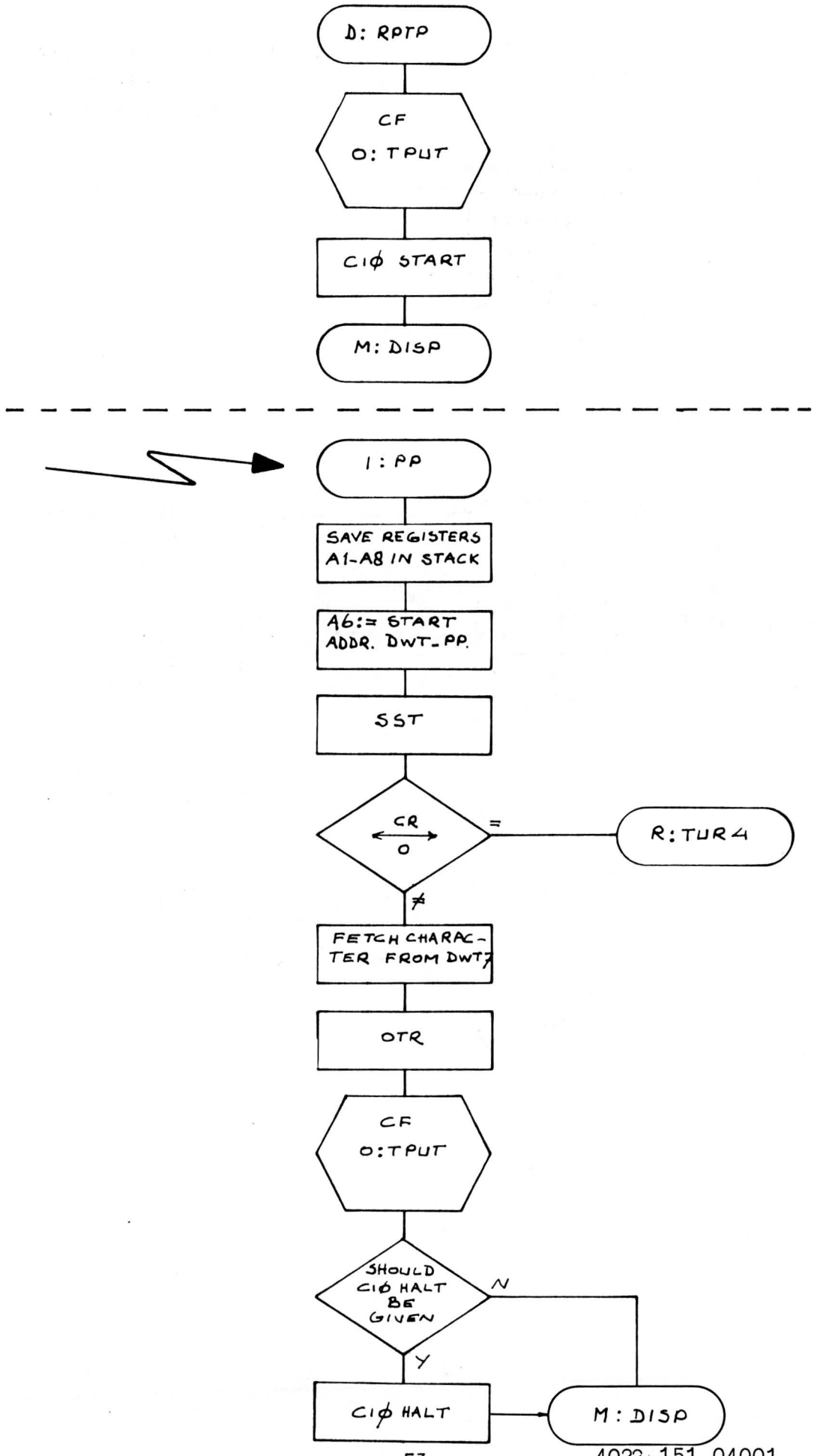
EXIT : A6: DWT-ADDRESS
A7: ORDER
A8: CSB-ADDRESS



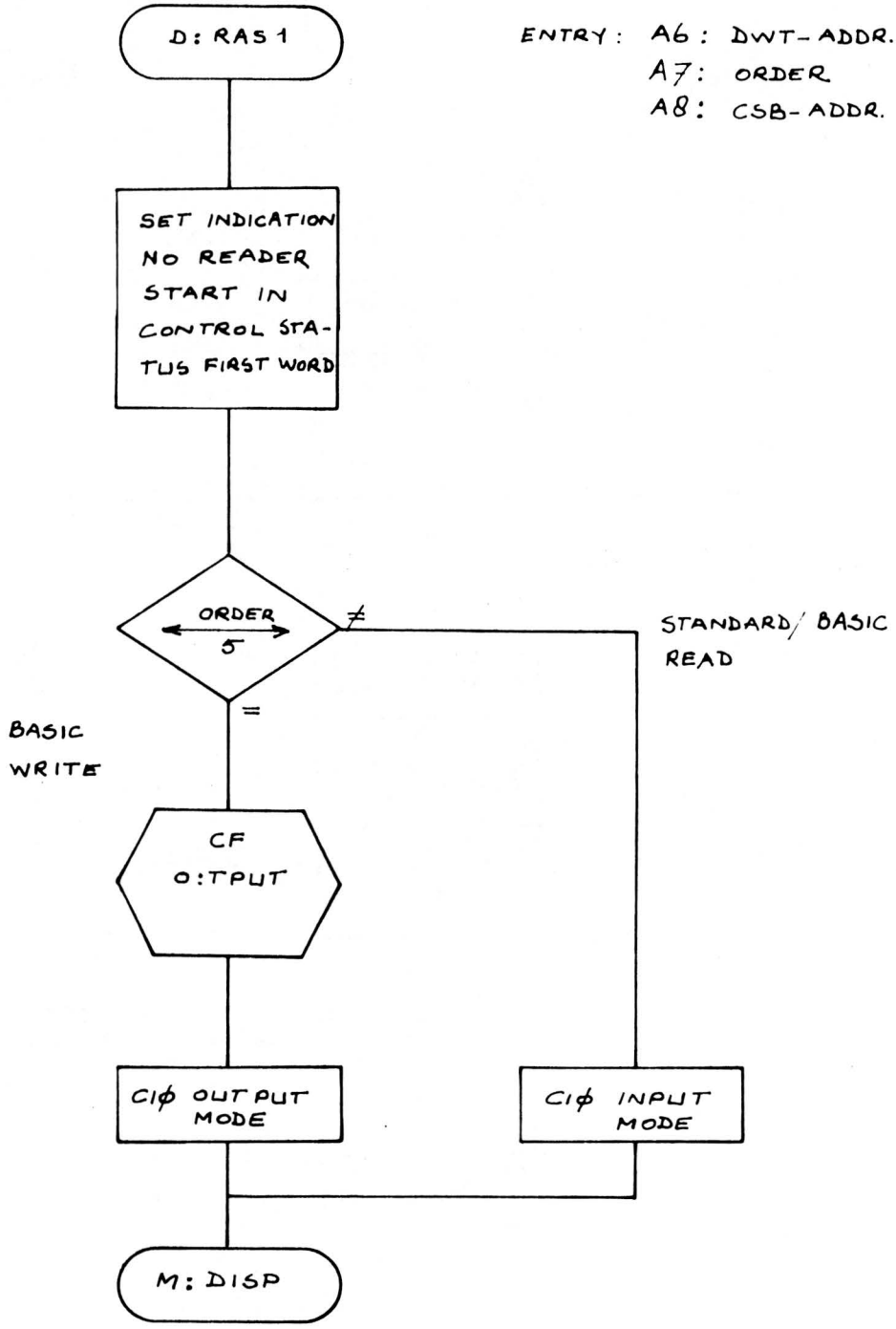
8. DRIVER HSR


D: RPTR
CIO START
M: DISP
I: PR
SAVE REG. A1-A8 IN STACK
A6:= START ADDRESS DWT
SST
CR
O
=
≠
CHARACTER INTERRUPT
INR A2
R: TUR4
STATUS INTERRUPT
CF
I: NPUT
SHOULD CIØ HALT BE GIVEN
N
Y
CIØ HALT
M: DISP

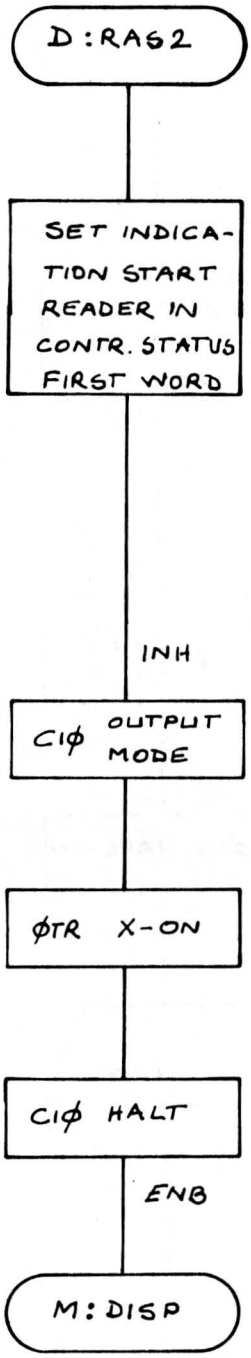
9. DRIVER HSP


D: RPTP
CF
O: TPUT
CIØ START
M: DISP
I: PP
SAVE REGISTERS A1-A8 IN STACK
A6:= START ADDR. DWT_PP.
SST
CR
O
=
R: TUR4
≠
FETCH CHARAC-TER FROM DWT
OTR
CF
O: TPUT
SHOULD CIØ HALT BE GIVEN
N
Y
CIØ HALT
M: DISP

10. DRIVER ASR KEYBOARD

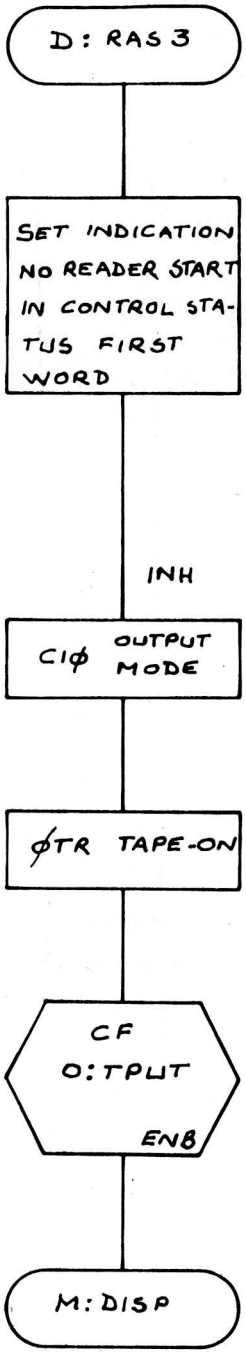


11. DRIVER ASR READER



ENTRY: A6: DWT-ADDR.
A7: ORDER
A8: CSB-ADDR.

12. DRIVER ASR PUNCH

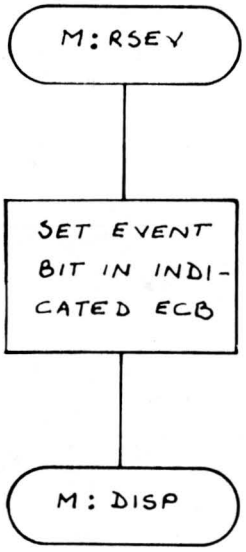


ENTRY: A6: DWT- ADDR.
A7: ORDER
A8: CSB-ADDR.

TAPE-OFF HAS TO BE GIVEN BY USER IN OUTPUT BUFFER

4022 151 04001 bl. 625-58

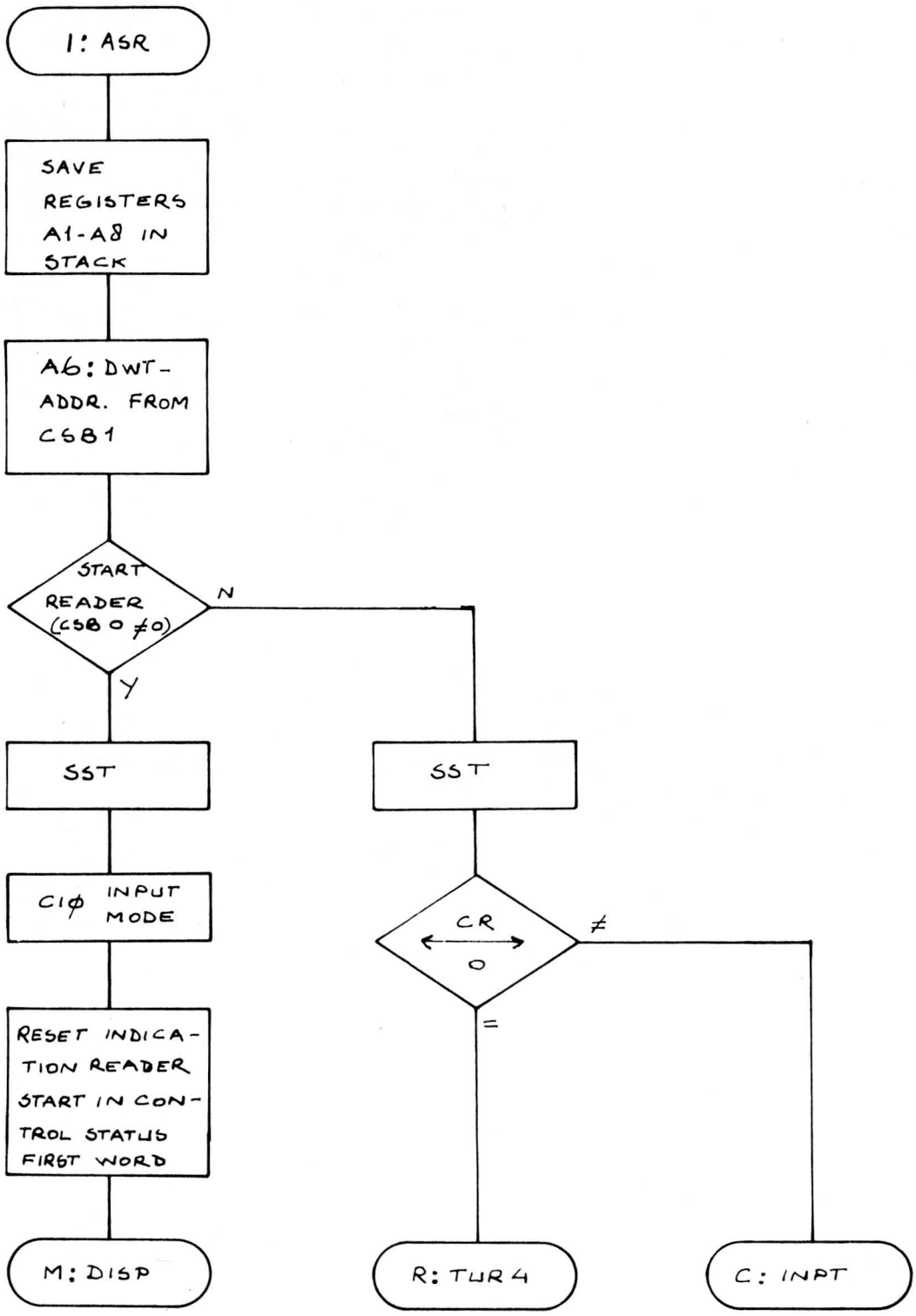
13. RESET EVENT

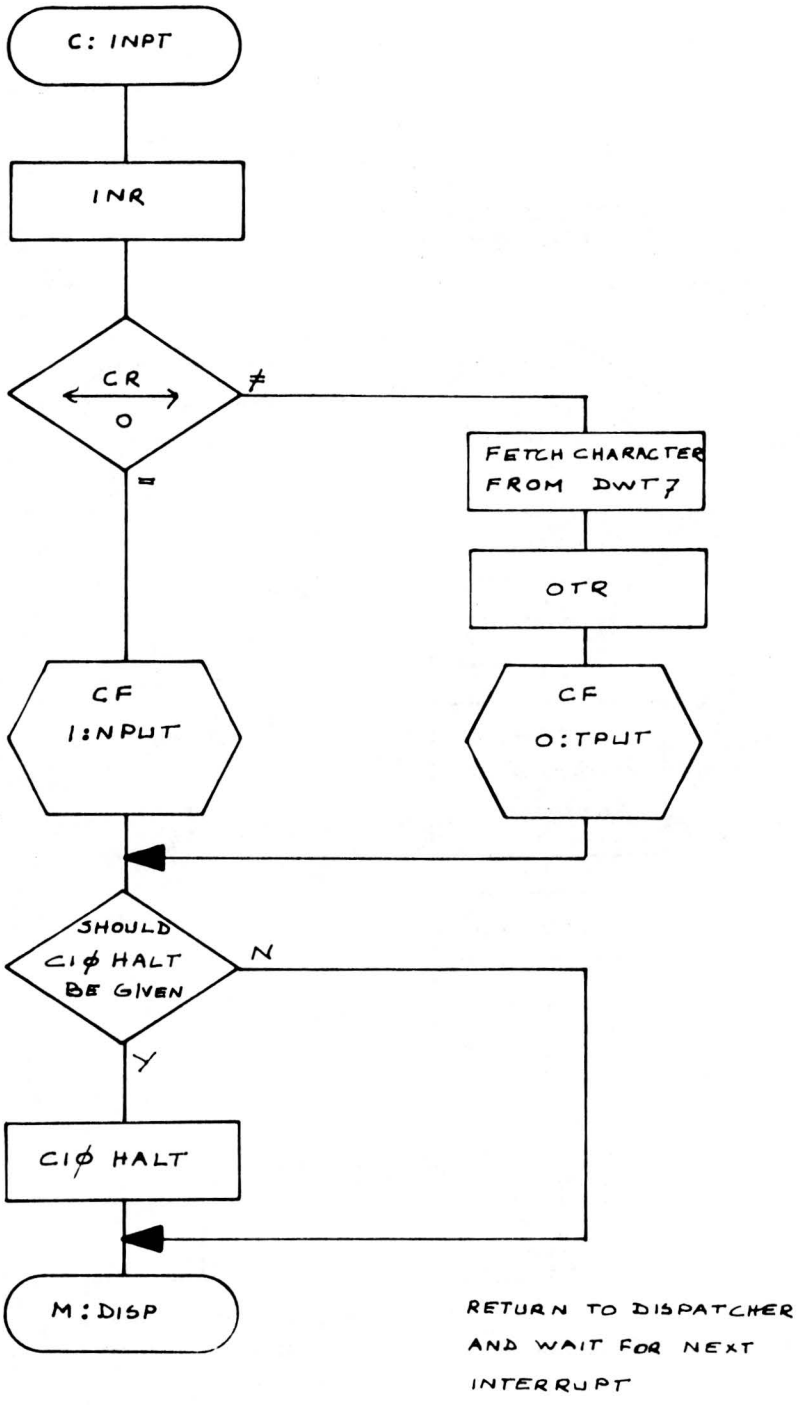


ENTRY: A8: ADDR.-ECB

RETURN TO ACTIVE PROGRAM ON HIGHEST SOFTWARE LEVEL

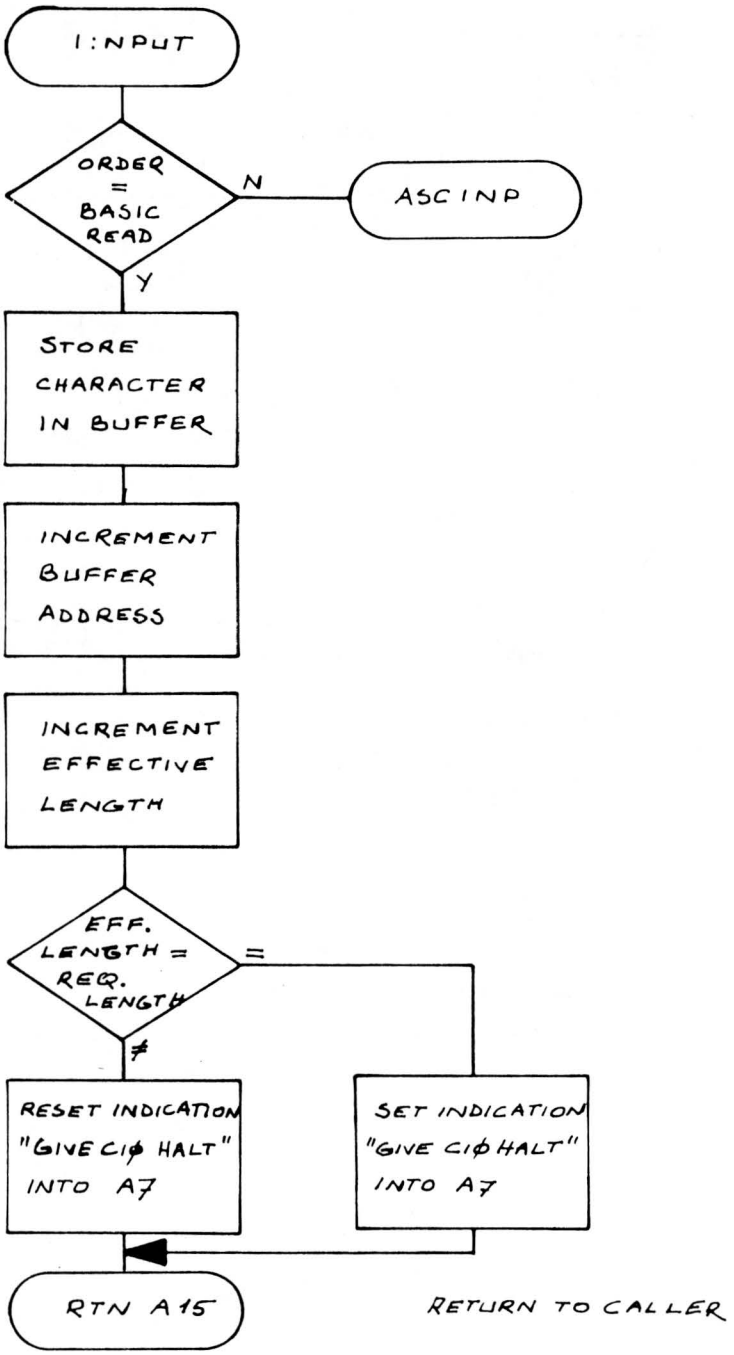
14. INTERRUPT HANDLING ASR



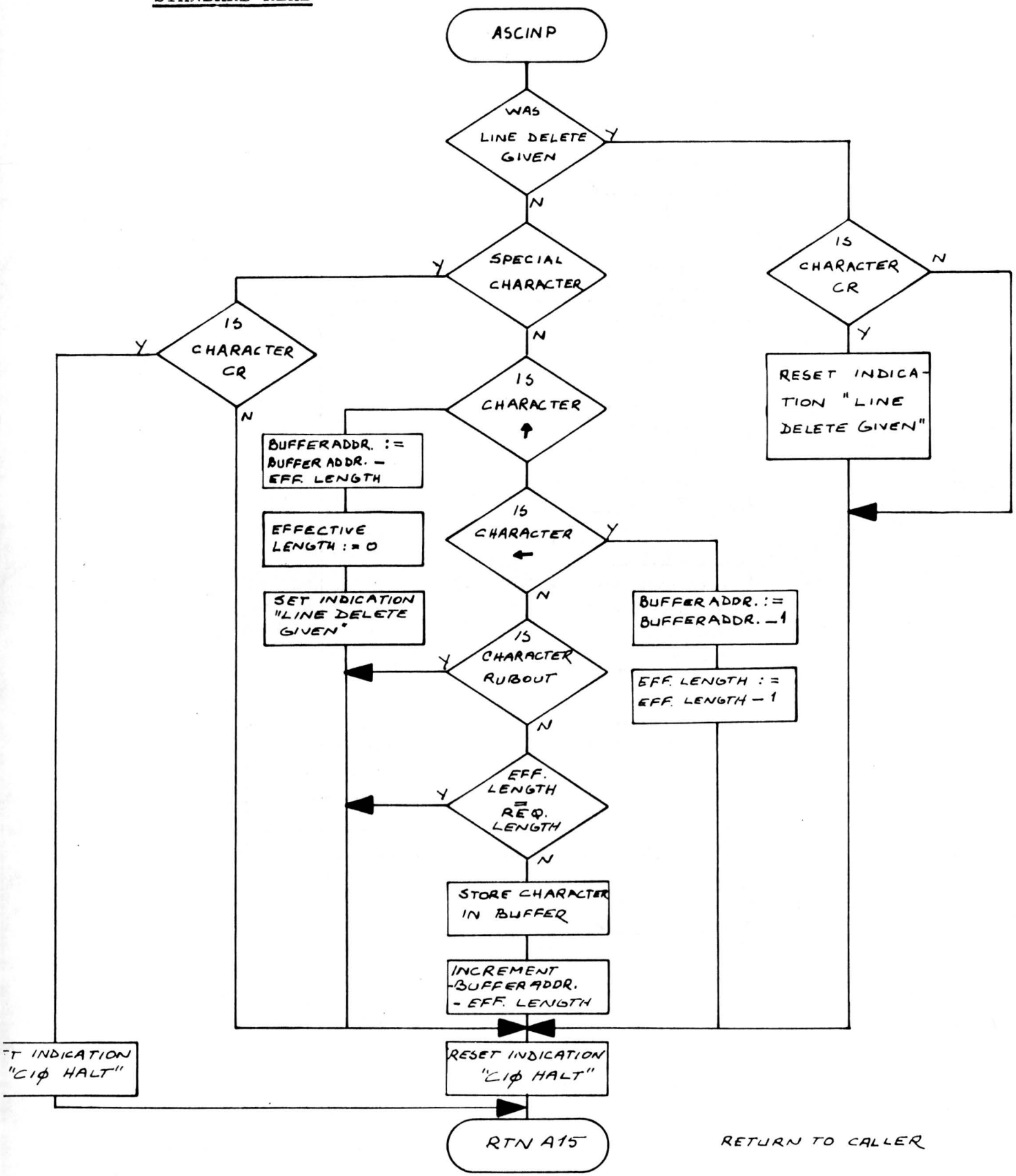

C: INPT
INR
CR
0
≠
=
FETCH CHARACTER FROM DWT7
OTR
CF
I:NPUT
CF
O:TPUT
SHOULD CIØ HALT BE GIVEN
N
Y
CIØ HALT
M:DISP
RETURN TO DISPATCHER AND WAIT FOR NEXT INTERRUPT

15. COMMON SUBROUTINE HANDLE INPUT

ENTRY: A6: DWT-ADDR. EXIT: A7: = 0 GIVE NO CIϕ HALT
A2: INPUT CHAR. ≠ 0 GIVE CIϕ HALT

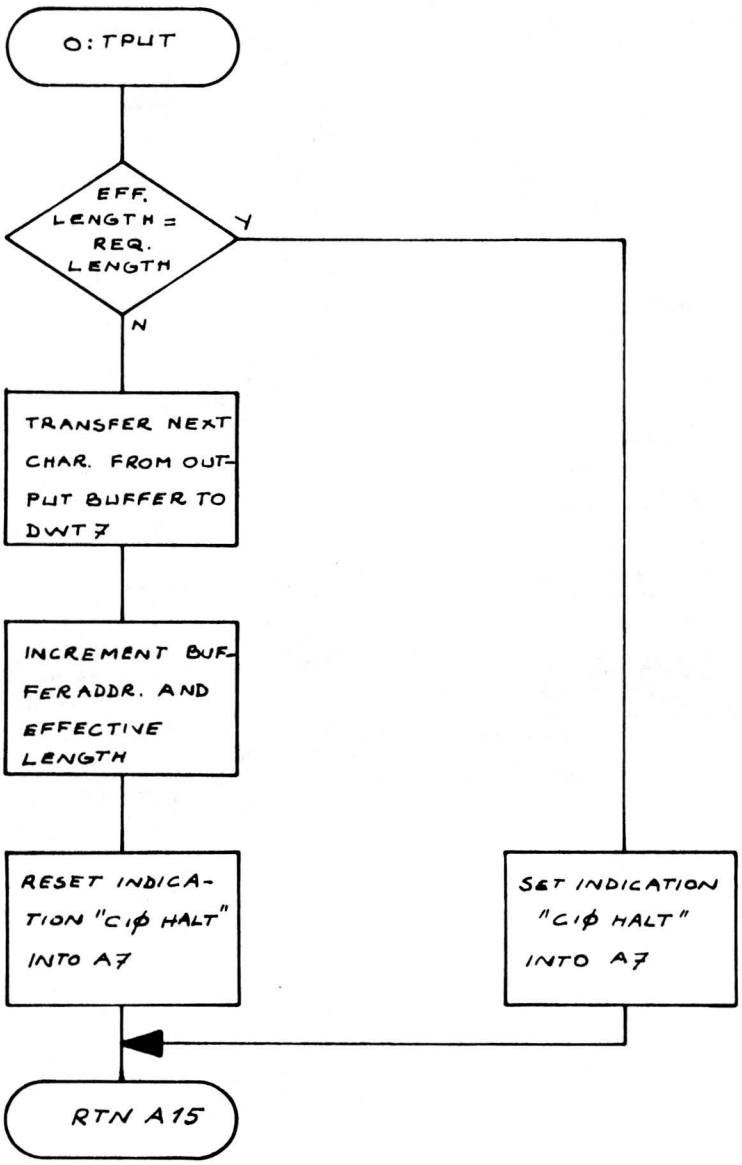


COMMON SUBROUTINE
HANDLE INPUT (CON.)
STANDARD READ



16 COMMON SUBROUTINE
HANDLE OUTPUT

ENTRY : A6 : DWT-ADDR
EXIT : A7 : = 0 GIVE NO CIØ HALT
≠ 0 GIVE CIØ HALT

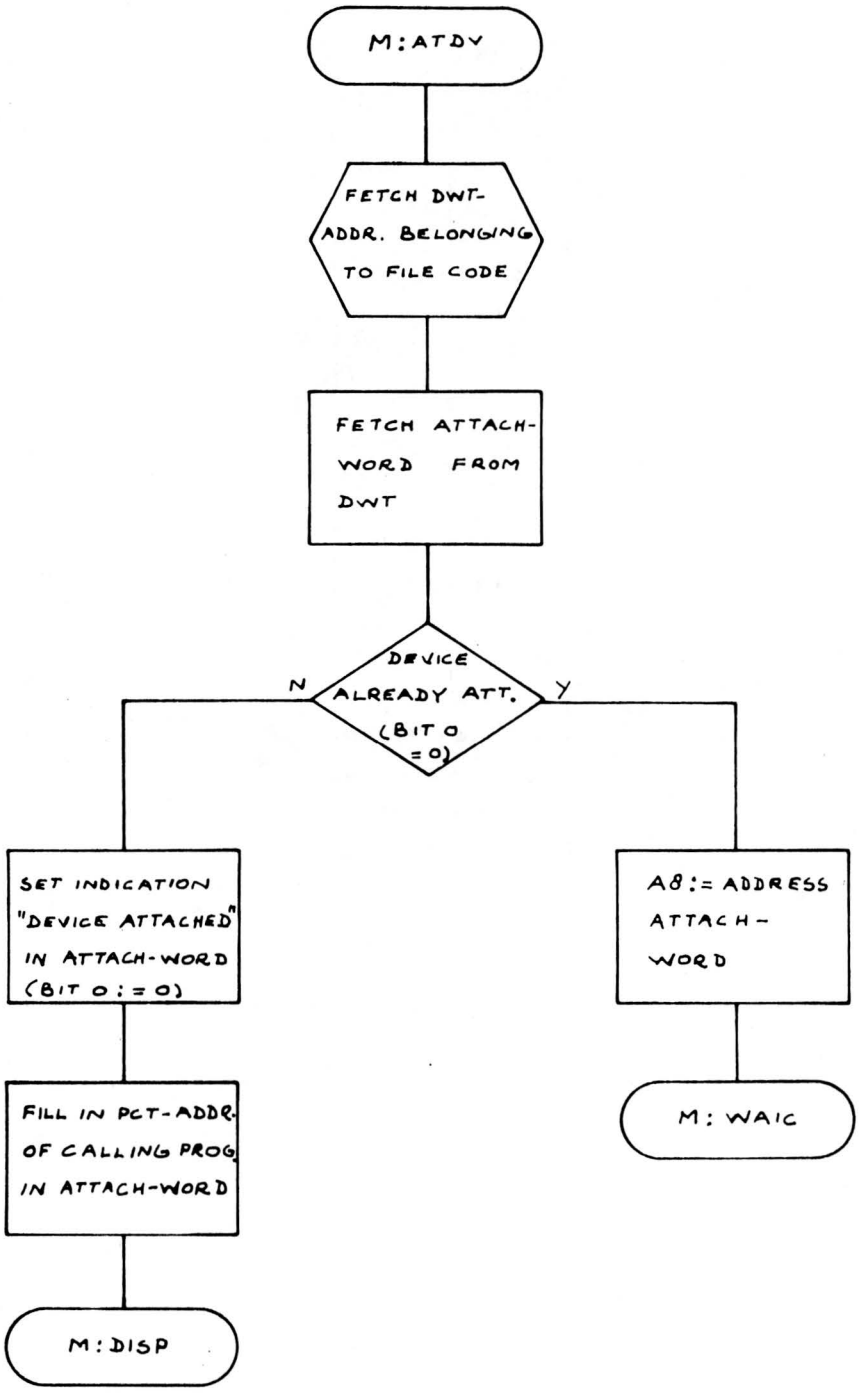


17. END OF INPUT/OUTPUT HANDLING

ENTRY : A6 : DWT-ADDR.

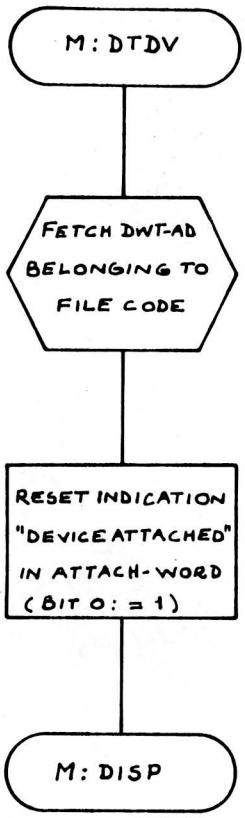
18. ATTACH MONITOR REQUEST

ENTRY : A8 : ADDRESS FILE CODE
A5 : PCT-ADDR. CALLING PROGRAM



19. DETACH MONITOR REQUEST

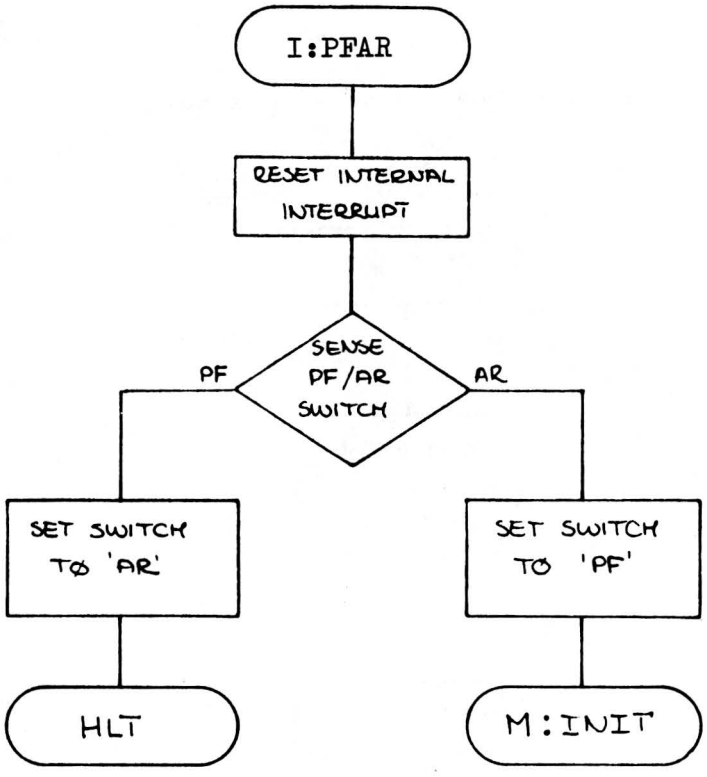
ENTRY: A8 : ADDRESS FILE CODE



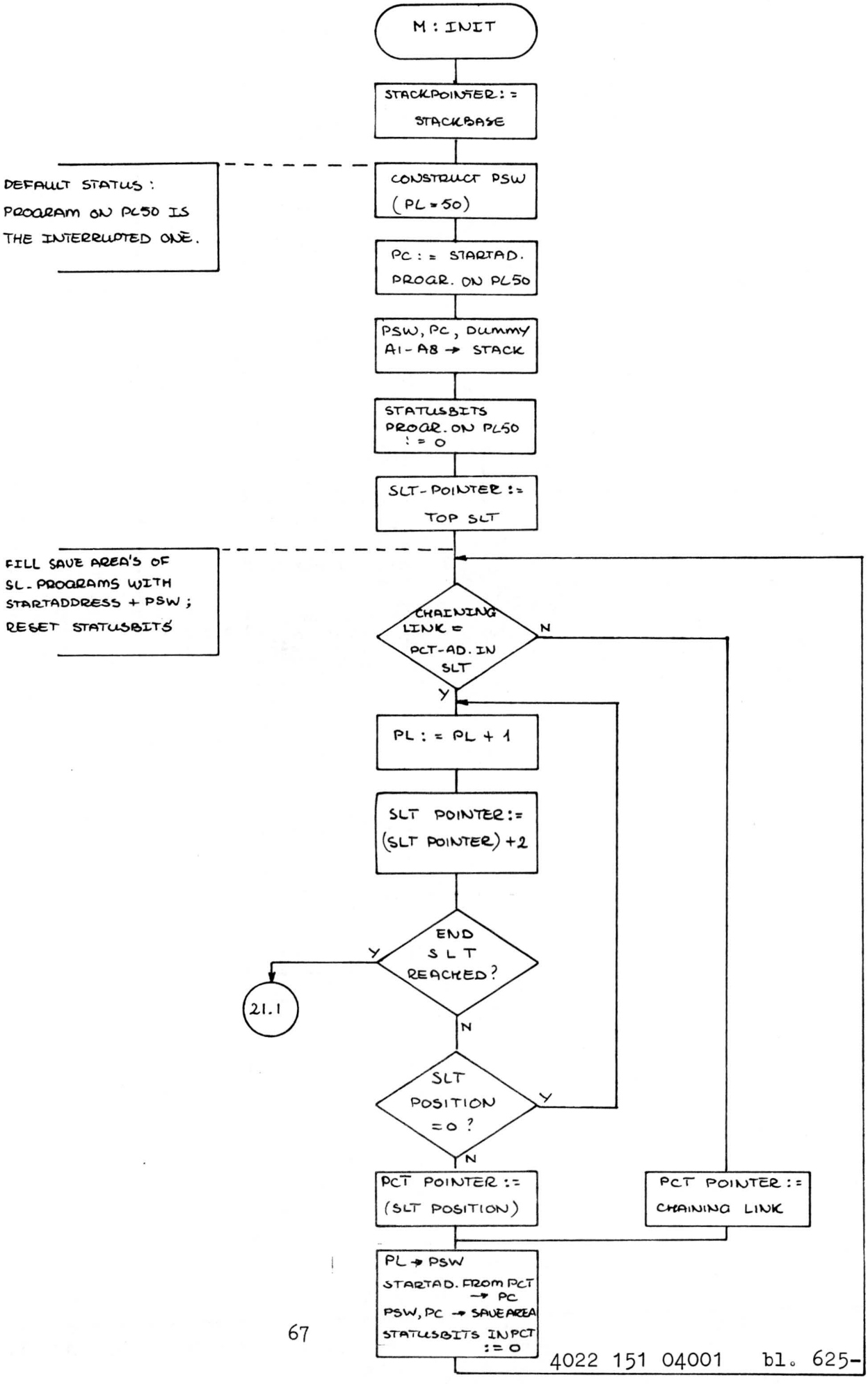
RETURN TO ACTIVE PROGRAM
ON HIGHEST SOFTWARE LEVEL

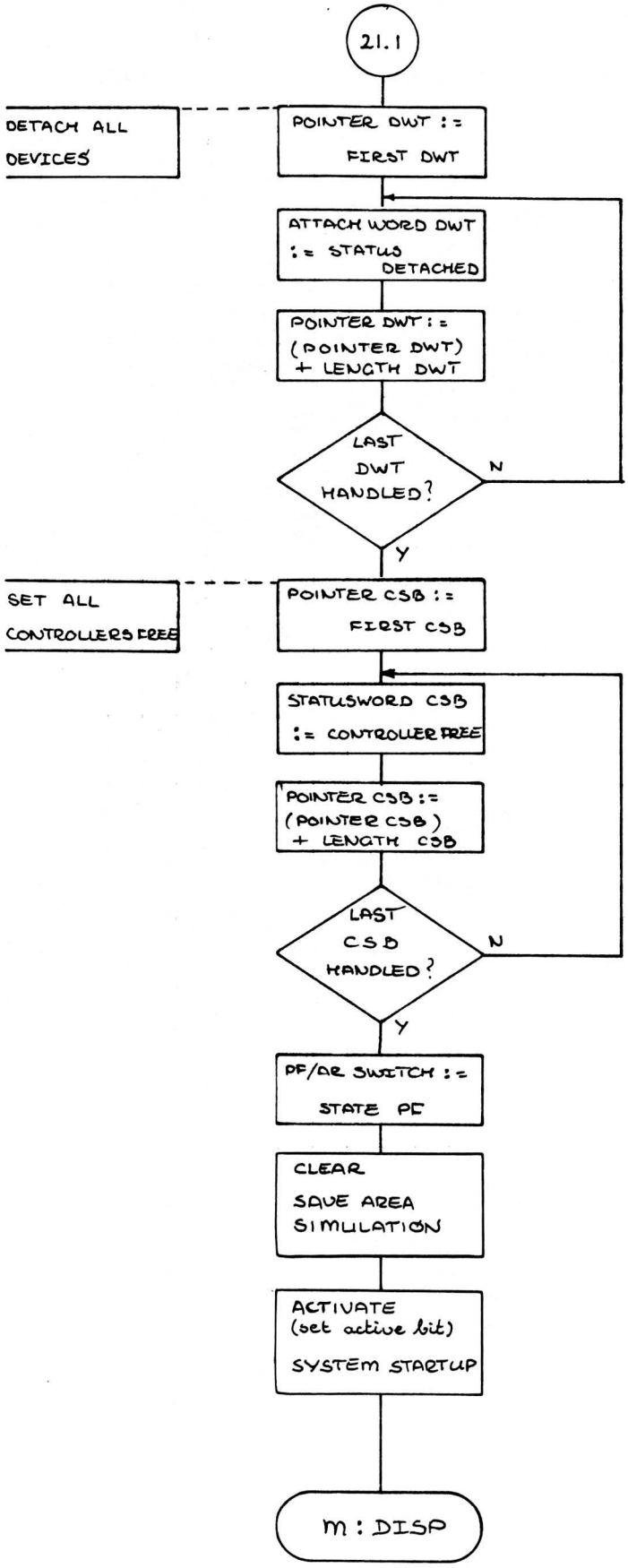
4022 151 04001 bl. 625-67

20. POWER FAILURE / AUTOMATIC RESTART



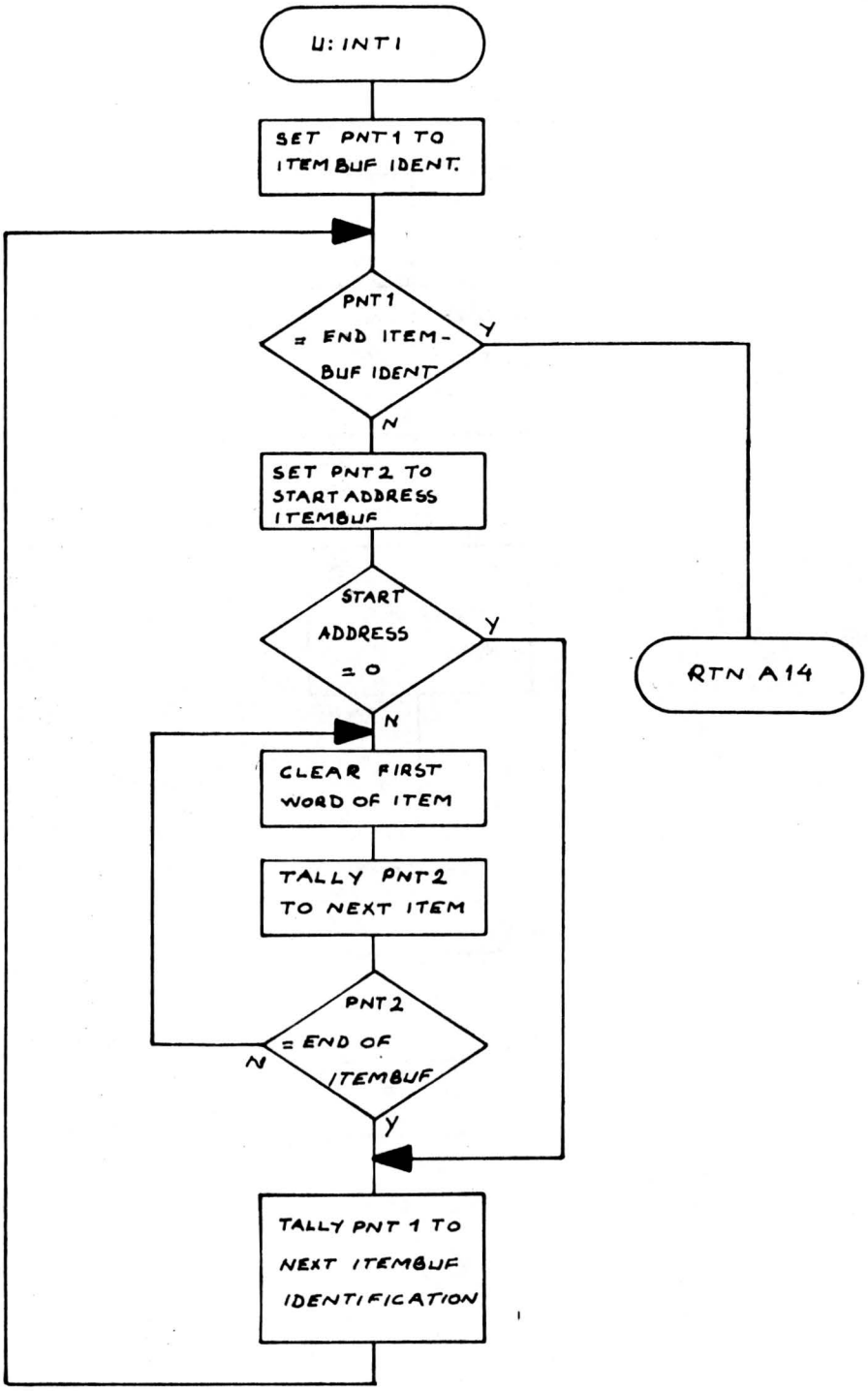
21. MONITOR INITIALISATION




21.1
DETACH ALL DEVICES
POINTER DWT := FIRST DWT
ATTACH WORD DWT := STATUS DETACHED
POINTER DWT := (POINTER DWT) + LENGTH DWT
LAST DWT HANDLED?
N
Y
SET ALL CONTROLLERS FREE
POINTER CSB := FIRST CSB
STATUSWORD CSB := CONTROLLER FREE
POINTER CSB := (POINTER CSB) + LENGTH CSB
LAST CSB HANDLED?
N
Y
PF/DR SWITCH := STATE PF
CLEAR SAVE AREA SIMULATION
ACTIVATE (set active bit) SYSTEM STARTUP
M : DISP

TIMER MODULE (M:TIMR)

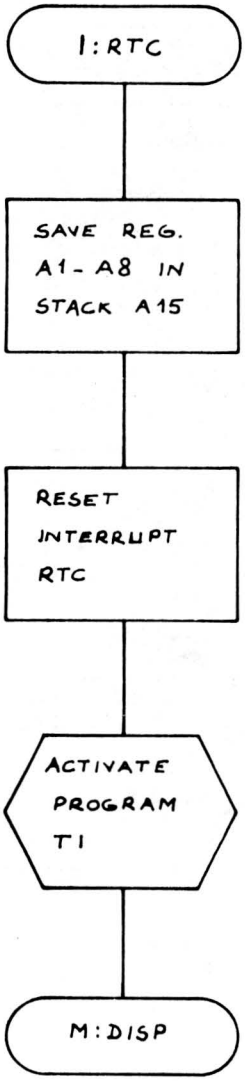
Initialisation



4022 151 04001 bl. 625-71

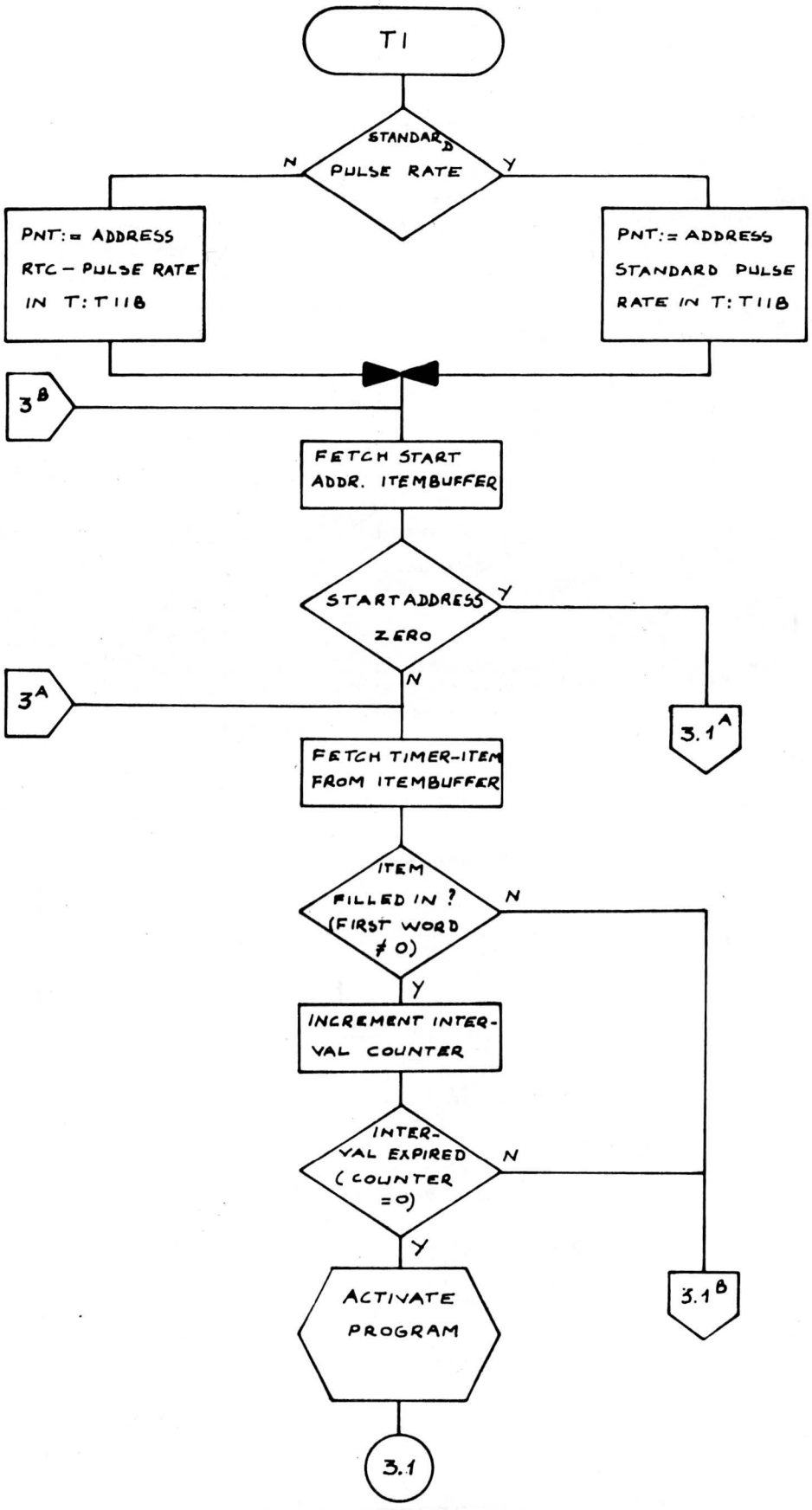
TIMER MODULE (M:TIMR)

Interrupt Mode



TIMER FUNCTION (M:TIMR)

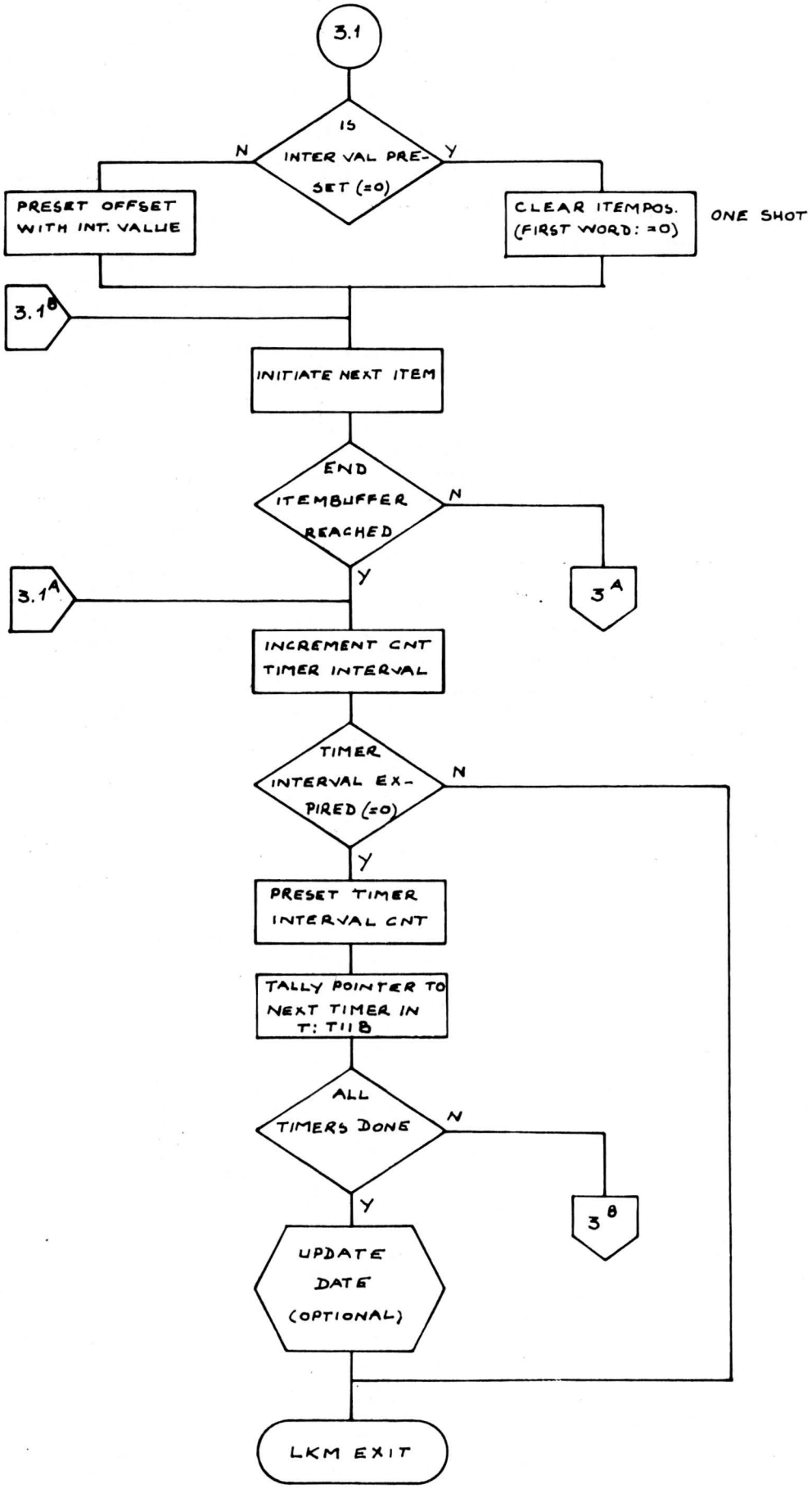
Non-Interrupt Mode



4022 151 04001 bl. 625-73

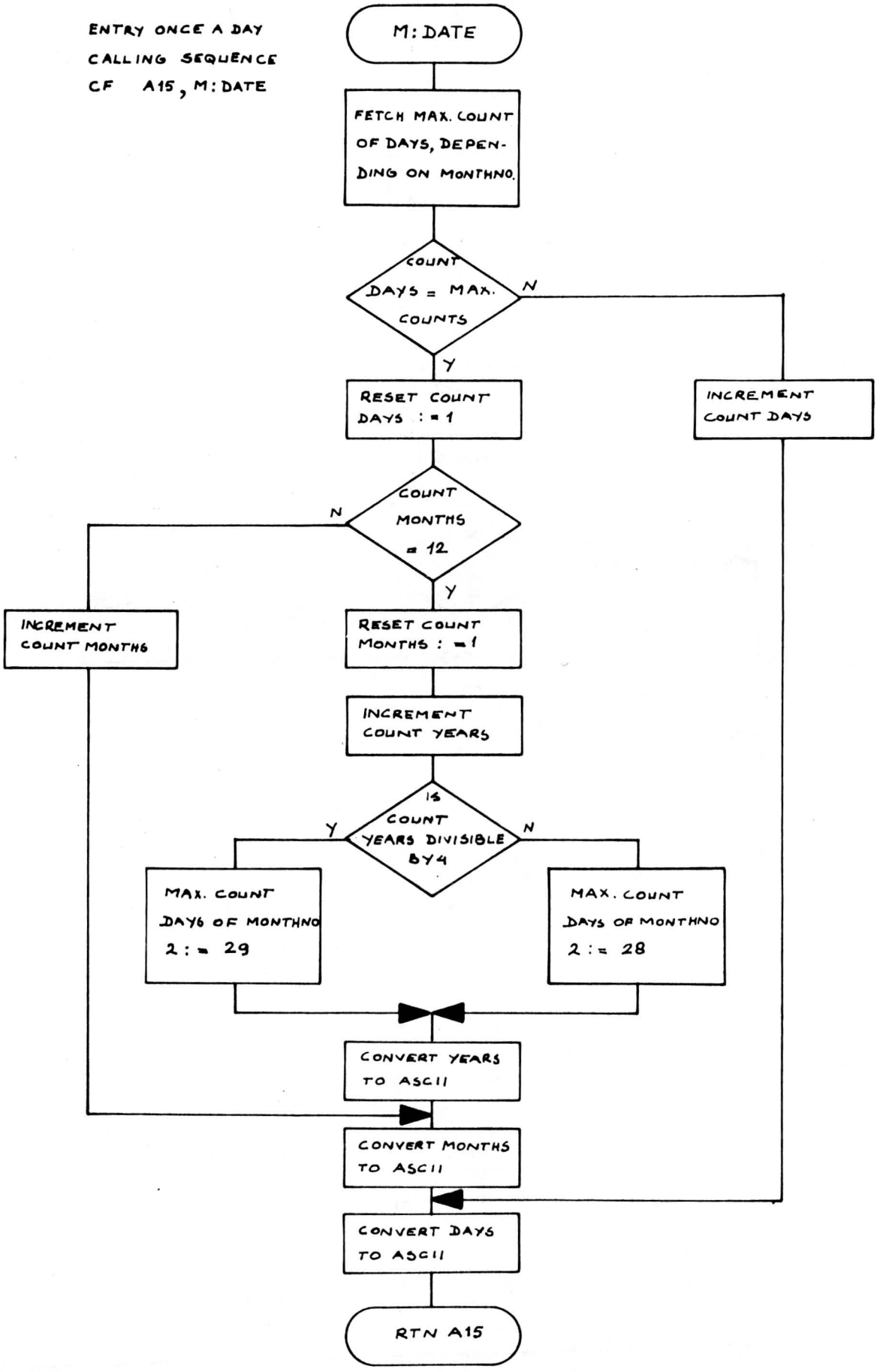
TIMER FUNCTION (M:TIMR)

Non-Interrupt Mode (cont.)



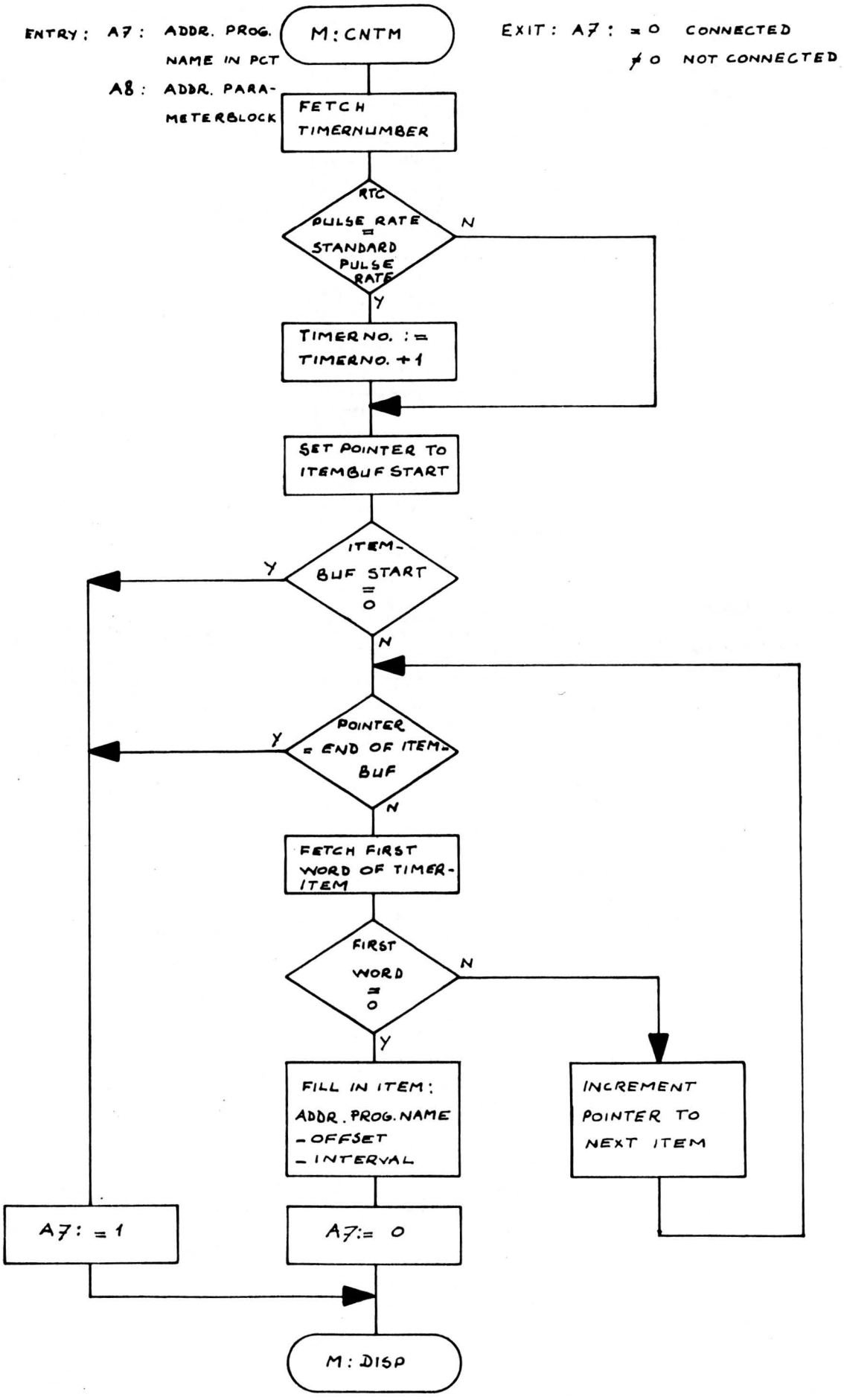
ROUTINE DATE UPDATE

ENTRY ONCE A DAY
CALLING SEQUENCE
CF A15, M:DATE

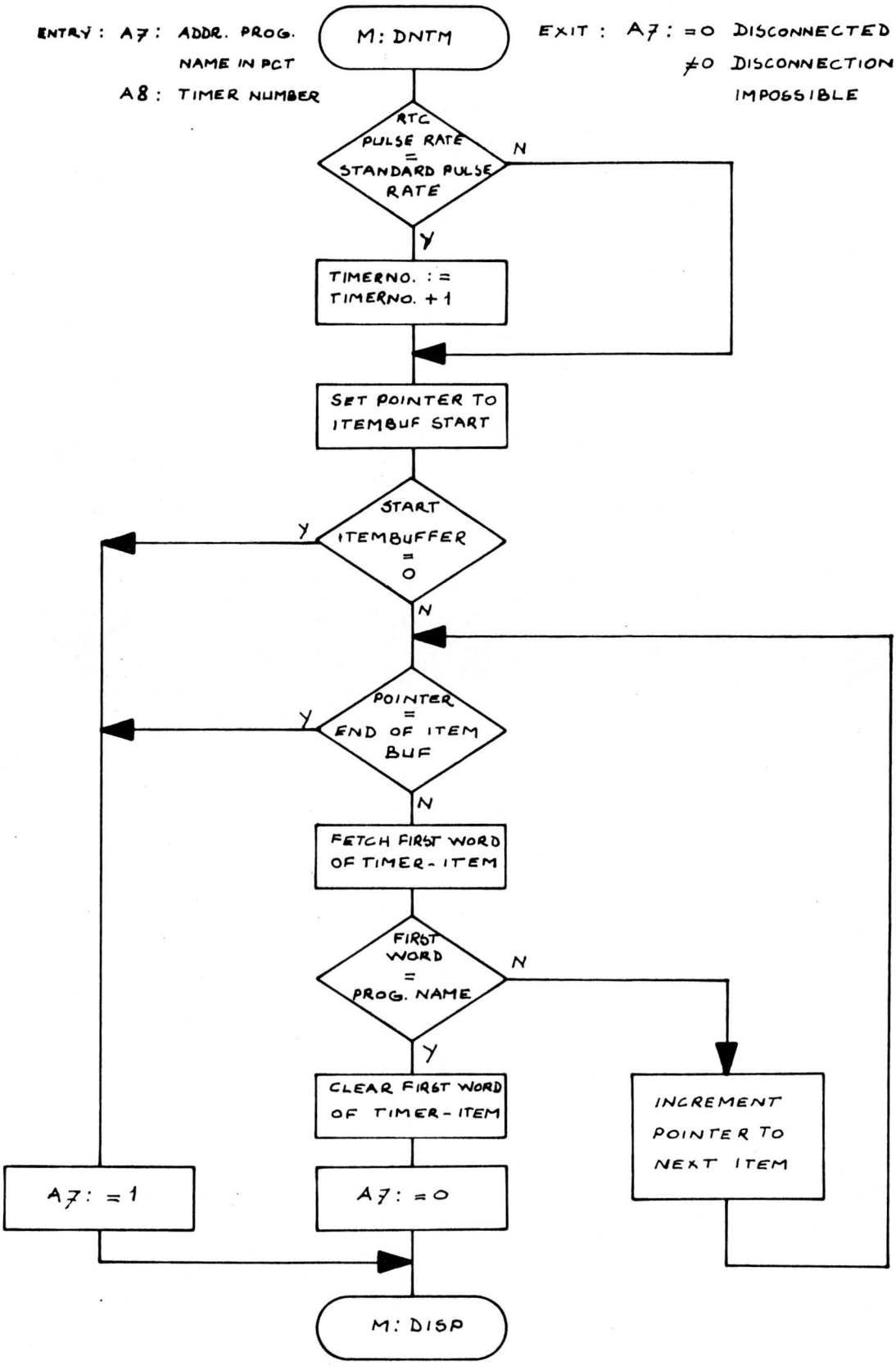


4022 151 04001 bl. 625-75

CONNECT TIMER MONITOR REQUEST



DISCONNECT TIMER MONITOR REQUEST



12. Examples

12.1 Example of user program using the SRTM without timerfunctions.

4022 151 04001 bl. 625-78

```
00000                   IDENT    TSTPRG                SRTM52   750103                ASM 05
00001     *
00002     *
00003     *
00004     *         CATEGORY   :  STANDARD APELDOORN SOFTWARE
00005     *
00006     *
00007     *
00008     *         AUTHOR     :  N.V. PHILIPS GLOEILAMPENFABRIEKEN P.I.T.-I.D.S.
00009     *
00010     *
00011     *
00012     *         PROGRAMMER :  D.M.G. BOS
00013     *
00014     *
00015     *
00016     *         COMP.CONF. :  P852
00017     *
00018     *
00019     *
00020     *         SUMMARY    :  EXAMPLE OF SYSTEM USING SRTM
00021     *
00022     *
00023     *
```


4022 151 04001 bl. 625-79

```
00024                           EJECT
00025                   *
00026                   * THIS PROGRAM ECHOES MESSAGES FINISHED BY CR.
00027                   * POSSIBLE INPUT DEVICES: ASR-KEYBOARD (05 - D:WAS1)
00028                   *                         ASR-READER   (06 - D:WAS2)
00029                   *                         PT-READER    (08 - D:WPTR)
00030                   * POSSIBLE OUTPUT DEVICES:ASR-KEYBOARD (05 - D:WAS1)
00031                   *                         ASR-PUNCH    (07 - D:WAS3)
00032                   *                         PT-PUNCH     (09 - D:WPTP)
00033                   *
00034                   * CORRECTION SIGNS "VERTICAL ARROW" AND "HORIZONTAL ARROW" ARE TAKEN
00035                   * INTO ACCOUNT, CHARACTERS RANGING FROM 0-1F ARE DELETED (STANDARD READ)
00036                   *
00037                   * START THE INPUT-OUTPUT CYCLE BY PRESSING THE INTERRUPT BUTTON
00038                   * ON THE CONTROL PANEL.
00039                   *
00040                   * INPUT- AND OUTPUT DEVICES CAN BE ALTERED BY MODIFYING THEIR
00041                   * SPECIFICATIONS LISTED BELOW.
00042                   *
00043                   ************************************************************************
00044                   *
00045                   * ENTRY POINTS
00046                   *
00047                           ENTRY   STRTST
00048                           ENTRY   STRTTI
00049                           ENTRY   STRTTS
00050                           ENTRY   I:CP
00051                   *
00052                   * EXTERNAL DECLARED LABELS
00053                   *
00054                           EXTRN   PCTTS
00055                           EXTRN   NAMTS
00056                           EXTRN   M:DISP
00057                   *
00058                   * INPUT DEVICE SPECIFICATION
00059                   *
00060                   * ECB INPUT DEVICE
00061  0000  0006       ECB1    DATA    06                      FILE CODE
00062  0002  0010  R            DATA    BUFFER
00063  0004  0048               DATA    72
00064  0006                     RES     1
00065                   *
00066                   * OUTPUT DEVICE SPECIFICATION
00067                   *
00068                   * ECB OUTPUT DEVICE
00069  0008  0005       ECB2    DATA    05                      FILE CODE
00070  000A  0010  R            DATA    BUFFER
00071  000C  0048               DATA    72
00072  000E                     RES     1
```


4022 151 04001 bl. 625-80

```
00073                          *
00074                          * BUFFERS
00075                          *
00076  0010                    BUFFER   RES      36                  IN- AND OUTPUTBUFFER
00077  0058                    DUMBLK   RES      2                   DUMMY PARAMETERBLOCK FOR ACTIVAT
00078                          *********DUMMY TIMERFUNCTION*********
00079  005C                             RES      16                  SAVE AREA
00080  007C  2C7F              STRTTI   HLT                          ENTRY FROM DISPATCHER
00081                          *********SYSTEM START-UP*********
00082  007E                             RES      16                  SAVE AREA
00083  009E  2804              STRTST   LKM                          NOTHING TO DO
00084  00A0  0003                       DATA     3                   EXIT
00085                          *********TESTPROGRAM*********
00086  00A2                             RES      16                  SAVE AREA
00087                          STRTTS   EQU      *                   ENTRY FROM DISPATCHER
00088  00C2  BC20                       MLK      8                   CHECK TRAP-ACTION AND SIMULATION
00089  00C4  0000                       DATA     0                   A1
00090  00C6  0000                       DATA     0                   A2
00091  00C8  0000                       DATA     0                   A3
00092  00CA  0000                       DATA     0                   A4
00093  00CC  0000                       DATA     0                   A5
00094  00CE  0000                       DATA     0                   A6
00095  00D0  0001                       DATA     1                   A7, ATTACH INPUT DEVICE
00096  00D2  0000  R                    DATA     ECB1                A8, AD FILE CODE
00097  00D4  2804                       LKM
00098  00D6  000E                       DATA     14
00099  00D8  0248                       LDK      A2,72               SET REQ.LENGTH IN
00100  00DA  8241                       ST       A2,ECB1+4           TO ECB INPUTDEVICE
       00DC  00D4  R
00101  00DE  0702                       LDK      A7,2                STANDARD READ
00102  00E0  80A0                       LDKL     A8,ECB1
       00E2  0000  R
00103  00E4  2804                       LKM
00104  00E6  0001                       DATA     1
00105  00E8  80A0                       LDKL     A8,ECB1             WAIT ON INPUT FINISHED
       00EA  0000  R
00106  00EC  2804                       LKM
00107  00EE  0002                       DATA     2
00108  00F0  8240                       LD       A2,ECB1+6           TRANSFER EFF.LENGTH OF INPUT
       00F2  0006  R
00109  00F4  8241                       ST       A2,ECB2+4           TO REQ. LENGTH OUTPUT
       00F6  00DC  R
00110  00F8  80A0                       LDKL     A8,ECB1             DETACH INPUT DEVICE
       00FA  0000  R
00111  00FC  2804                       LKM
00112  00FE  000F                       DATA     15
00113  0100  0701                       LDK      A7,1                ATTACH OUTPUT DEVICE
00114  0102  80A0                       LDKL     A8,ECB2             ADDRESS FILE CODE
       0104  00D8  R
00115  0106  2804                       LKM
```


4022 151 04001 bl. 625-81

```
00116  0108  000E          DATA    14
00117  010A  0705          LDK     A7,5                    BASIC WRITE
00118  010C  80A0          LDKL    A8,ECB2
       010E  0008  R
00119  0110  2804          LKM
00120  0112  0001          DATA    1
00121  0114  80A0          LDKL    A8,ECB2                 WAIT ON OUTPUT FINISHED
       0116  0008  R
00122  0118  2804          LKM
00123  011A  0002          DATA    2
00124  011C  80A0          LDKL    A8,ECB2                 ADDRESS FILE CODE
       011E  0008  R
00125  0120  2804          LKM
00126  0122  000F          DATA    15                      DETACH OUTPUT DEVICE
00127  0124  2804          LKM                             EXIT TESTPROGRAM
00128  0126  0003          DATA    3
00129                  *
00130                  *********INTERRUPT FROM CONTROL PANEL*********
00131                  *
00132  0128  813F    I:CP  STR     A1,A15                  SAVE A1-A8 IN STACK
00133  012A  823F          STR     A2,A15
00134  012C  833F          STR     A3,A15
00135  012E  843F          STR     A4,A15
00136  0130  853F          STR     A5,A15
00137  0132  863F          STR     A6,A15
00138  0134  873F          STR     A7,A15
00139  0136  883F          STR     A8,A15
00140  0138  28DF          RIT     /0F                     RESET INTERRUPT LINE
00141  013A  2840          ENB
00142  013C  8720          LDKL    A7,NAMTS                A7:= ADDR. NAME TESTPROGR.
       013E  0000  X
00143  0140  80A0          LDKL    A8,DUMBLK               A8:= ADDR. DUMMY PAR. BLOCK (COMP
       0142  0058  R
00144  0144  2804          LKM
00145  0146  000C          DATA    12                      ACTIVATE TESTPROGRAM
00146  0148  8120          ABL     M:DISP                  GO TO DISPATCHER
       014A  0000  X
00147                  * END OF TSTPRG
00148                  END
```


4022 151 04001 bl. 625-82

SYMBOL TABLE

BUFFER	001C	R	DUMBLK	0058	R	ECB1	0000	R	ECB2	0008	R
I:CP	012C	R	M:DISP		X	NAMTS		X	PCTTS		X
STRTST	009E	R	STRTTI	007C	R	STRTTS	00C2	R			

ASS.ERR. 00000

OF

PROG ELAPSED TIME: 00H-01M-01S-180MS-

KEY /O

LABEL = IDS00xx DATE = 74/11/13 PACK NBR = 008 SRTM52

ASM TSTTI

DATE 75 /01 /06 TIME 09H-20M-22S-

LABEL = IDS00xx DATE = 74/11/13 PACK NBR = 008 SRTM52

4022 151 04001 bl. 625-83

12.2 **Example** of user program using the SRTM with timerfunctions.

4022 151 04001 bl. 625-84

```
00000                    IDENT    TSTTI                SRTM52   750103                ASM 05
00001                    *
00002                    *
00003                    *
00004                    *        CATEGORY   :  STANDARD APELDOORN SOFTWARE
00005                    *
00006                    *
00007                    *
00008                    *        AUTHOR     :  N.V. PHILIPS GLOEILAMPENFABRIEKEN P.I.T.-I.D.S.
00009                    *
00010                    *
00011                    *
00012                    *        PROGRAMMER :  D.M.G. BOS
00013                    *
00014                    *
00015                    *
00016                    *        COMP.CONF. :  P852
00017                    *
00018                    *
00019                    *
00020                    *        SUMMARY    :  EXAMPLE TIMER-FUNCTION USE
00021                    *
00022                    *
00023                    *
```


4022 151 04001 bl. 625-85

```
                        EJECT
                *
                * THIS IS A TESTPROGRAM FOR THE TIMER FUNCTION OF SRTM.
                * ONCE STARTED BY CONTROL PANEL INTERRUPT, TIME AND DATE
                * ARE PRINTED EACH 10 SECONDS ON ASR.
                * THIS CYCLE IS ENDED AFTER ANOTHER CP-INTERRUPT.
                *
                ****************************************************************
                *
                * ENTRY POINTS
                *
                        ENTRY   STRTST
                        ENTRY   STRTTS
                        ENTRY   I:CP
                *
                *  EXTERNAL DECLARED LABELS
                *
                        EXTRN   PCTTS
                        EXTRN   NAMTS
                        EXTRN   M:DISP
                        EXTRN   U:INTI
                        EXTRN   BIN2AS
                        EXTRN   H:TIME
                        EXTRN   M:TIME
                        EXTRN   S:TIME
                        EXTRN   W:DAY
                        EXTRN   W:MONT
                        EXTRN   W:YEAR
                *
                * OUTPUT DEVICE SPECIFICATION
                *
0000  0005      ECBOUT  DATA    05
0002  0008  R           DATA    BUFFER
0004  0010              DATA    16
0006  0000              DATA    0
                * OUTPUT BUFFER
0008  0A0D      BUFFER  DATA    /0D0A                   CR, LF
000A            TIME    RES     3
0010  2020              DATA    /2020                   TWO SPACES
0012            DATE    RES     3
                * VARIABLE FIELD
0018  0A00      PARBLK  DATA    /0A00                   TIMERNO.= 2; INTERVAL = 10
001A  0000              DATA    0                       OFFSET = 0
001C            CONSW   RES     1                       CONNECT/DISCONNECT TIMER SWITCH
                *********SYSTEM START-UP*********
001L                    RES     1
0020            A14STK  RES     1                       A14-STACK
0022                    RES     16                      SAVE AREA
0042  86A0      STRTST  LDKL    A14,A14STK
```


4022 151 04001 bl. 625-86

```
        0044  0020  R
00073   0046  F6A1        CF      A14,U:INTS          INITIALIZE TESTPROGRAM
        0048  0052  R
00074   004A  F6A1        CF      A14,U:INTI          INITIALIZE TIMER
        004C  0000  X
00075   004E  2F04        LKM
00076   0050  0003        DATA    3                   EXIT SYSTEM START-UP
00077                   *********TESTPROGRAM*********
00078                   *
00079                   * INITIALISATION TESTPROGRAM
00080                   *
00081   0052  0101  U:INTS LDK     A1,1                SET CONNECT SWITCH
00082   0054  8141        ST      A1,CONSW
        0056  001C  R
00083   0058  F63A        RTN     A14                 RETURN TO SYSTEM START-UP
00084                   *
00085                   * SAVE AREA TESTPROGRAM
00086                   *
00087   005A              RES     16
00088                   * THIS IS THE ENTRY FROM DISPATCHER
00089   007A  8140  STRTTS LD      A1,H:TIME           CONVERT HOURS TO ASCII
        007C  0000  X
00090   007E  1118        ADK     A1,24
00091   0080  F7A1        CF      A15,BIN2AS
        0082  0000  X
00092   0084  8241        ST      A2,TIME
        0086  000A  R
00093   0088  8140        LD      A1,M:TIME           CONVERT MINUTES TO ASCII
        008A  0000  X
00094   008C  113C        ADK     A1,60
00095   008E  F7A1        CF      A15,BIN2AS
        0090  0000  X
00096   0092  8241        ST      A2,TIME+2
        0094  000C  R
00097   0096  8140        LD      A1,S:TIME           CONVERT SECONDS TO ASCII
        0098  0000  X
00098   009A  113C        ADK     A1,60
00099   009C  F7A1        CF      A15,BIN2AS
        009E  0000  X
00100   00A0  8241        ST      A2,TIME+4
        00A2  000E  R
00101   00A4  8140        LD      A1,w:YEAR           STORE DATE IN OUTPUT BUFFER
        00A6  0000  X
00102   00A8  8141        ST      A1,DATE
        00AA  0012  R
00103   00AC  8140        LD      A1,w:MONT
        00AE  0000  X
00104   00B0  8141        ST      A1,DATE+2
        00B2  0014  R
00105   00B4  8140        LD      A1,w:DAY
```


4022 151 04001 bl. 625-87

```
        00B6  0000  X
00106   00B8  8141        ST      A1,DATE+4
        00BA  0016  R
00107   00BC  0705        LDK     A7,5            BASIC WRITE
00108   00BE  80A0        LDKL    A8,ECBOUT
        00C0  0000  R
00109   00C2  2E04        LKM
00110   00C4  0001        DATA    1
00111   00C6  80A0        LDKL    A8,ECBOUT       WAIT ON OUTPUT FINISHED
        00C8  0000  R
00112   00CA  2E04        LKM
00113   00CC  0002        DATA    2
00114   00CE  2E04        LKM
00115   00D0  0003        DATA    3               EXIT TESTPROGRAM
```


4022 151 04001 bl. 625-88

```
00116                          EJECT
00117                  *
00118                  *********INTERRUPT FROM CONTROL PANEL************
00119                  *
00120                  I:CP    EQU     *
00121  00D2  813F              STR     A1,A15          SAVE A1-A8
00122  00D4  823F              STR     A2,A15
00123  00D6  833F              STR     A3,A15
00124  00D8  843F              STR     A4,A15
00125  00DA  853F              STR     A5,A15
00126  00DC  863F              STR     A6,A15
00127  00DE  873F              STR     A7,A15
00128  00E0  883F              STR     A8,A15
00129  00E2  2F0F              RIT     /0F             RESET INTERRUPT LINE
00130  00E4  2E40              ENB
00131  00E6  8140              LD      A1,CONSW        CHECK IF CONNECT TIMER TO BE DONE
       00E8  001C  R
00132  00EA  5C14              RF(0)   DISCON          NO
00133  00EC  0100              LDK     A1,0            YES
00134  00EE  8141              ST      A1,CONSW        SET SWITCH TO DISCONNNECT
       00F0  001C  R
00135  00F2  8720              LDKL    A7,NAMTS        A7:= ADDR PROGRAMNAME IN PCT
       00F4  0000  X
00136  00F6  88A0              LDKL    A8,PARBLK       A8:= ADDR. PAR.BLOCK TIMING
       00F8  0018  R
00137  00FA  2E04              LKM
00138  00FC  000A              DATA    10              CONNECT TIMER MONITOR REQUEST
00139  00FE  5714              RF      EXIT
00140  0100  8140      DISCON  LD      A1,PARBLK       ELIMINATE TIMERNO. FROM PAR.BLK
       0102  0018  R
00141  0104  396C              SRL     A1,12
00142  0106  8084              LDR     A8,A1           A8:=  TIMERNO.
00143  0108  8720              LDKL    A7,NAMTS        A7:= ADDR PROGRAMNAME IN PCT
       010A  0000  X
00144  010C  8741              ST      A7,CONSW        SET SWITCH TO CONNECT
       010E  001C  R
00145  0110  2E04              LKM
00146  0112  000B              DATA    11              DISCONNECT TIMER MONITOR REQUEST
00147  0114  6F20      EXIT    ABL     M:DISP          GO TO DISPATCHER
       0116  0000  X
00148                  * END OF TSTTI
00149                          END
```

SYMBOL TABLE

A14STK	0020	R	BIN2AS		X	BUFFER	0008	R	CONSW	001C	R
DATE	0012	R	DISCON	0100	R	ECBOUT	0000	R	EXIT	0114	R
H:TIME		X	I:CP	00D2	R	M:DISP		X	M:TIME		X
NAMTS		X	PARBLK	0018	R	PCTTS		X	S:TIME		X
STRTST	0042	R	STRTTS	007A	R	TIME	000A	R	U:INTI		X
U:INTS	0052	R	W:DAY		X	W:MONT		X	W:YEAR		X

ASS.LRP. 00000

OF

PROG ELAPSED TIME: 00H-01M-05S-100MS-

KMT /U

LABEL = IDS00L8 DATE = 74/11/13 PACK NBR = 008 SRTM52

ASG /E0,1Y10

BYE

DATE 75 /01 /06 TIME 09H-24M-22S-

Comment Sheet

P800M Programmer's Guide 1 - Volume V: SRTM 5122 991 27351

Name _____

Company _____

Department _____

Address _____

_____

Telephone Number _____ ext. _____

Comments or Suggestions:

PHILIPS DATA SYSTEMS B.V.

MARKETING GROUP SMALL COMPUTERS
P.O. Box 245, Apeldoorn, The Netherlands
Phone: 055-230123; telex: 49142

For further details contact the above address or:

EUROPE

Sweden
Svenska AB Philips
Data Systems
Minidatorer
Rissneleden 16
Fack
172 07 Sundbyberg
Tel. 08 830300

Denmark
Philips Data Systems A/S
Prags Boulevard 80
2300 København S
Tel. 0127 2222

Norway
Norsk A/S Philips
Data Systems Division
Nils Hansens vei 2
P.O. Box 5040
Oslo 6
Tel. 02 679380

Finland
OY Philips AB
Department Data Systems
Kaivokatu 8
P.O. Box 10255
Helsinki 10
Tel. 90 17271

Belgium
Philips Data Systems SA
Marketing Group
Small Computers
Anspachlaan 1
1000 Brussel
Tel. 02 2193900

France
Philips Data Systems
Département Mini-ordinateurs
5 Square Max-Hymans
75015 Paris 15
Tel. 01 734 7759

Western Germany
Philips GmbH-Eiserfeld
Bereich Prozessrechner
Münsterstrasse 330
4 Düsseldorf 30
Tel. 0211 631064

Höhenstrasse 17
7012 Fellbach bei Stuttgart
Tel. 0711 523081

Austria
Österreichische Philips GmbH
Industrie Elektronik
Breitenfurterstrasse 219
1230 Wien
Tel. 0222 831501

Italy
Philips s.p.a.
Sezione P.I.T.
Via Elvezia 2
20052 Monza
Tel. 04 361441

Switzerland
Philips AG Data Systems
Binzstrasse 18
8027 Zürich
Tel. 01 442211

Great Britain
Philips Data Systems
Elektra House
2 Bergholt Road
Colchester
C04-5AA Essex
Tel. 206 5115

The Netherlands
Philips Nederland B.V.
Professionele
Produkten en Systemen
Boschdijk 525
Eindhoven
Tel. 040 782953
788325

Spain
Philips Ibérica S.A.E.
Grupo Instrumentación
Martinez Villergas 2
Madrid 27
Tel. 091 4042200

FAR EAST

Japan
Nihon Philips Corporation
P.O. Box 13
World Trade Centre
Hamamatsu-cho, Minato-ku
Tokyo 105
Tel. 03 435 5211

NORTH AMERICA

U.S.A.
North American Philips Corp.
Dept. 007
100 East 42nd Street
New York N.Y. 10017
Tel. 212 697 3600

Canada
Philips Electronics
Industries Ltd.
Telecommunication Division
1001, Ellesmere Road
Scarborough 706
Ontario
Tel. 416 7521980