CHD

Title:

SYSTEM RELEASE FORM
TELEDATA VERSION 3 RELEASE 5

A/S REGNECENTRALEN

RC SYSTEM LIBRARY: FALKONERALLE 1 DK-2000 COPENHAGEN F

RCSL No: 21-T033
Edition: March 79
Author: Carl Henrik Dreyer

Keywords:

Release, TELEDATA version 3, RC8000, online, data base, invoicing, order processing.

Abstract:

Description of release 5 of the TELEDATA System version 3. Load instruction, contents of system tape, revisions. The system includes application programs for online DB maintenance, invoicing and order processing, utility programs for system creation and maintenance, administration programs for routine online processing, and batch processing.

English edition, 61 pages.

Printed by A/S Regnecentralen, Copenhagen

TABLE OF CONTENTS PAGE

TABLES OF CONTENTS (CONTINUED) PAGE

TABLE OF CONTENTS (CONTINUED) PAGE

Name: TELEDATA VERSION 3, RELEASE 5

Date: March the 15th, 1979

Date of last release: December the 15., 1978.

1. GENERAL DESCRIPTION. 1.

TELEDATA is an online, real time system for the RC8000 computer. It includes a general data base system and supports invoicing and order processing in a multi-user environment. For general introduction, please consult:

RCSL 42-i0706	Introduction to TELEDATA
RCSL 21-T017	A Survey of TELEDATA Terms

Release 5 includes at total system:

- file systems
- tools
- utility programs
- online application skeleton program
- online application model solution
- batch applications
- a complete model for creation and running of the system.

The most significant news are:

- a new online model solution
- many new features in the online skeleton program
- faster online utility programs

This release form contains a detailed description of the revisions in the skeleton program. Beside the purpose of describing the changes, we hope that this will substitute the obvious lack of an updated skeleton program manual.

In the coming updating of all the documentation, this need will be met.

New manuals: Teledata User's handbook (RCSL 21-T028)
Teledata utility program TELESTATAC
(RCSL 21-T030)

Application Programming System (RCSL 21-T027)

2. LOAD INSTRUCTION. 2.

For loading the system tape, the following BOSS2-job will suffice. The user name and number refer to the recomandations in appendix A. Dependent on the format of the tapes delivered, the job may be changed slightly.

```
job scope 17 size 40000 time 15 0,
  perm disc 15000 200,
  stat 1
mode list.yes
tape = set mto <tape name>
load   release.no.tape.1
load   release.no.tape.2
load   release.no.tape.3
load   release.no.tape.4
load   release.no.tape.5
load   release.no.tape.6
finis
```

2.1 New Installations. 2.1

When installing the system from scratch, run the jobs in tape file 5.

2.2 Existing Installations. 2.2

- re-do language translation of "print_data_error" in source file newaktt (or copy your version to the new newaktt)

NB: newaktt dlkt i to filer (NH-skrift)

- implement the new skeleton program by performing the changes mentioned in section 5. This influences the DB and LD descriptions and part of your model solution.

- re-translate all the applications. (In SODALD, make a new vardecl). Save the old descripfile for later use.

- reorganize the data base. Terminal work files must be empty (terminals outside voucher).

- re-translate TELEOP with local newaktt and new vardecl (cfr. job transonline).

- if you do not implement the new skeleton program, you should anyway reorganize the master files after retranslation of DB description (cfr. section 4.2. Only DB description has to be re-translated as a result of this change).

- duetcom should be recompiled, as the source text is changed.

- the logfile blocklength must be changed in DB description (cfr. section 5.2). This implies that release 5 utility programs can not operate on log files created with release 4 utility programs and vice versa.

3. DESCRIPTION OF MASTER TAPE. 3.

The release is contained in one magnetic tape. It is created by the utility program SAVE.

Survey of master tape files:

File 1: CF system, BS system
File 2: Tools
File 3: TD utilities
File 4: TD applications
File 5: System creation
File 6: Miscellaneous
File 7: Empty, end save.

The following pages give a detailed description of the content of every tape file, plus references to documentation. Every tape file includes a text file (cat1, cat2 etc.), containing the names of all the files saved.

The references mentioned can be ordered from the RCSL library. The Teledata Bulletin no. 1 has the RCSL 21-T015.

Contents of tape file 1 (CF-BS-system)

Entries		References
catl		entry catalog
buflengthcf	closecf	
connect	deletechain	
dbrecdecode	deletel	
deletem	extendcf	cfsystem, RCSL 28-D5
gethead	getl	Introduction, RCSL 42-I0766
getm	getnumbl	Extensions, nov. 77:
getparamcf	headl	RC8000 Bulletin no. 1
headm	initchain	
initfilem	initrecm	
insertl	insertm	dbrecdecode: not used here
newreclcf	nextm	
opencf	protectcf	
putcf	readonlycf	
readupdcf	recnocf	
resultcf	setdescrcf	
setjumpscf	setparamcf	
updateallcf		
closebs	logbs	bs-system RCSL 31-D288 (danish)
openbs	resultbs	sortbs and sortprocbs:RCSL 21-V015
shlengthbs	sortbs	
sortprocbs		

Contents of tape file 2 (TOOLS)

Entries		References
cat2		entry catalog
database80	dbtable	RCSL 21-V045
sodald sodatext1 sodatext3 sdtrans	sodatable sodatext2 sodafields	RCSL 21-V019 (+ TD-bulletin no. 1)
duetabler duettext1 duettext3 prim	duettable duettext2 readgeneral tduetcode	RCSL 21-V046
sysdok		RCSL 21-V039
reanalyse reform	reextract reinsert	RCSL 21-V051
duetcomtext	stdcomopt	RCSL 21-T003 (+ TD-bulletin no. 1) stdcomopt: standard version of 'duetoptions'
print80		RCSL 21-V035
sort80	split80	RCSL 21-V038
smart fikstable	fiks	RCSL 21-V041

Contents of tape file 3 (Utilities)

Entries	References
cat3	entry catalog
teletext paramtext statustext newaktt	RCSL 21-T002 Teledata Information Letter no. 1
telelogex	RCSL 21-T008
Listlin writeall movetext longsum copychain initdbinf feltinf individinf registerinf listeinf filinf	Non-standard external algol-procedures required by TD No released documentation
list	Non-standard utility program. Lists text files with BOSS line numbers. Call: list <filename>
duetframe	standard frame. English FP-keywords. Cfr. RCSL 21-T005
teleadm	RCSL 21-T025
teletape telestatus telemove teleclock telereadcf telecleancf teletapr	RCSL 21-T008
teledbex	RCSL 21-T020
telerees	RCSL 21-T019
telestatac	RCSL 21-T030

Contents of tape file 4 (applications)

Entries	References
cat4	entry catalog
sysdokfile	A sysdokfile holding all source texts for applications. Password: td. Survey: Section 102 online_data_base 103 online_applications 103.1 local_data 103.2 skeleton_program RCSL 21-T006 (+ TD-bulletin no. 1) 103.3 model_solution RCSL 21-T028 110 batch_data_base 111 log_extraction RCSL 21-T008 112 db_extraction RCSL 21-T020 148 daily_reports RCSL 21-T021

Contents of tape file 5 (system creation)

Entries	References
cat5	entry catalog
tdbosscat	BOSS user catalog, reflecting the structure shown in appendix A.
transappl	BOSS job translating online applications and batch applications.
transonline	BOSS job translating TELEOP and DUETCOM
autorize	Text file describing how to autorize new versions of programs for routine use.
createdb	BOSS job creating and initializing a new data base.
dbtransact	Text file containing transactions processed in createdb
prepareruns	BOSS job creating all files needed for routine running of TELEADM
offtransact	Text file containing transactions to be used in an offline run or a demo

Contains of tape file 6 (Miscellaneous)

All entries are text files only to be used for 'emergency-maintenance'.

Entries	References
cat6	entry catalog
ttadm tttape ttstatus ttmove ttclock ttreadcf ttcleancf ttlogex tele80 statusfield tcopychain tcopybs ttdbex ttrees tttapr	No released documentation

4. REVISIONS IN TOOLS AND UTILITIES. 4.

4.1 CF-System 4.1

- minor changes, giving a better check of internal tables in list files.

- headl and headm now insert shortclock and content (23.1 and 22.1 respectively). This will appear, when new file heads are created. For information only.

- better alarm printouts from the system. The printouts are primary intended for the support and maintenance people, but will be described in an information letter.

4.2 DATABASE80. 4.2

- an error has been corrected. It concerned the analysis of the key fields in cf-master files, when the fields are collected in an aggregate, which contains more than one half field (which is the case in Teledata). The error resulted in a wrong content of the file heads. Therefore, the cf-master files must be reorganized after loading of the release 5.

4.3 REORG80. 4.3

- the system is extended compared to the manual, RCSL 21-V051. The extensions relevant to Teledata are described below.

- preservation of record chain order in list files. This is possible in the chain holding the primary list. The restrictions are: Primary list is the same before and after reorganization; all records must be connected in the primary list; call reanalyse with fp-parameter order.<list>, where <list> is the number of the primary list;the program reextract must read the primary mother-file.

- reanalyse. Internal table default size is augmented in order to avoid use of size parameter.

- extended ressource demand in reinsert. Needed in list file reorganization. Recommended is: size: 80000; area: 13; temp entries: 20; temp segments: about 2 times the biggest list file.

- new parameters in reinsert. The work space is as default created on the device disc. This may be changed by fp-parameter work. <discname>. The "newformat" file may be cleared just after reading (saves disc space) by adding "clear" to the fp-parameter: newformat. <filename>.clear.

- As previously stated, the system does not support files containing negative key fields. This is not a problem in the recommended use of Teledata.

4.4 SODA.

4.4

- the lookup operation is changed in order to make proper updating of some internal tables. Previously, the sequence get(14)-lookup(11)-put(14) resulted in destroying the record written by put(14).

- the internal handling of sq-files in the SODA DBMS is changed in order to speed up the SODA operations. No change in the use.

- the blocklength of sq-files may be any number of segments (stated in DB description). The sorting operation (algol 1) changed accordingly.

4.5 DUET

- in the DUET interpreter, the vital parts have been slang-coded. Therefore, in algol translations of TELEOP, the command "i tduetcode" is added.

- in printing duet data errors, the printout of variable names will only contain the first 23 characters of the name. Variable names of more than 23 characters has previously resulted in "movefield" from TELEOP.

- an error in the DUET TRACE facility, omitting part of the test output, has been corrected.

- numbers in input are restricted to 15 digits total (the maximum value of the digits being 140 737 488 355 327), and maximum 6 digits after the decimal point. This goes for both DUETABLER and DUET-system.

- testoutprint from "takevalue" and "varadr" (which didn't work in release 4) are now again available.

4.6 GENIUS. 4.6

- several minor errors have been corrected.

- in FIKS and SMART, printfile numbers may be up to 999 (previously max. 25).

- in FIKS, default number of identificators is extended from 300 to 400.

- in FIKS, the parameter "take.all" will check that the GENIUS-descriptions are read in order of increasing GENIUS-numbers.

- in SMART, number of printpages is counted one less.

4.7 TELEOP. 4.7

- the program is restructured in order to improve performance.

- facilities for use of "central output file" are implemented (the facilities ease reestablishing). A new fp-parameter is added in program call: cenout.<setno>. Setno is the set used for writing the file. The parameter must be stated, if the facility is used. Instruction in implementation will be described in a Teledata Information Letter.

- new system commands (from "system 00"):

  Key in: reso.

Gives a survey of restclaims (<segm>,<entries>) on the attatched bs devices.

Key in: tcomm <term.no>'<text string>.

The text string must be max. 40 characters. Writes the text string as an operator message in the reply area attatched to the terminal stated, thus sending a message dedicated to one terminal.

- new fp-parameter duetarea.<words>, stating the size in words of duet work area. Default is 3000. 8000-12000 is recommended.

- the program writes information at start and end, concerning date and time, version, elapsed time etc.

- the trap-facility of algol is implemented. This avoids breakdown in most cases of algol alarms. An alarm gives a message on the console, stating the alarm cause from the algol system (ref. ALGOL7), and the program continues.

NOT RELEASED

- when the fp-parameter rerun.yes is stated, the end of file position in the input file is read from statusfile (the fields "replog"). This affects the command file com2rerun (which is changed).

- the standard DUET reading alfabet is changed in two points: the SUB character (value 26) is made illegal and converted to & (value 38) in order to support RC800/20 parity error detection. The EM character (value 25) is in online mode made blind, which is the normal handling of irrelevant characters.

4.8 Algol Operations in TELEOP. 4.8

The algol operations (to call from the DUET program) are placed in the TELEOP source text newaktt, as usual. Remember to re-do the local editings in the procedure print_data_error (or replace the procedure with your local version).

algol 1 - sorting: Revised, in order to sort files with any blocklength.

The following algol operations are added:

Algol operation no. 13 - get input transaction.

Returns a copy of the input transaction

call: algol 13 (<text>)

<text> is a text variable declared for 147 chars

After a call of 'algol 13', <text> will contain an exact copy of the transaction, as it looked when it was read by TELEOP, except for the terminal number in fromt of the keyed-in line (supplied by DUETCOM), i.e. <text> contains the text keyed in at the keyboard.

Algol operation no. 14 - message on operator console.

Makes it possible to write a message on the operator console from the duet program.

Call: algol 14 (<text>,<param1>,<param2>)

text	a text variable declared for 17 chars
param1	word containing an integer value
param2	word containing an integer value

A call of 'algol 14' causes a message on the operator console with format as shown below.

user no.: <actual user no>
term no.: <actual term no>
trans no.: <actual trans no>
- the message in <text> .....
- the value in <<param1> and <param2>

Algol Operation No. 15 - Calender Function

This operation is used to calculate a new date based on a specified time span and a specified date.

call: algol 15 (start_date, time_span, function, due_date)

start_date	(call)	word
time_span	(call)	word
function	(call)	word
due_date	(return)	word

The call values are subject to the following restrictions.

start date - interpreted as being in the form of yymmdd with:

yy >= 77,
1 <= mm <= 12 and
1 <= dd <= last day in mm.

time span - interpreted as a number of days or a number of months, dependent of the value of the function parameter.

due date - the calculated date, represented as yymmdd, or an error indication:

0 - illegal call value of function
1 - illegal call value of start date.

function - controls the function used for the calculation of the due date:

1 - the time span is interpreted as being in days, and is added to the start date.
2 - the time span is interpreted as being in months, and is added to start date.
3 - the due date is calculated as the current month plus the time span, measured in days.

Algol operation no. 25 - move texts of different length. 4.9

The algol operation may be used for truncation of texts - f.ex. "short names".

call: algol 25(adr(to_text), adr(from_text))
Notice the adr-operations!

to_text variable, the text is moved to.
from_text variable, the text is moved from.

The variables must be of the type text and without a subscript. The length of the text string to be moved is evaluated as the length of the shortest of the two variables. An even number of

bytes is transferred and a 0-character is inserted in the last 8 bits of the text.

4.9 Duetcom.

Minor changes are made.

A reported error, saying that the system 00 terminal might be logged in from two different devices (if you go on trying), is a result of the definition of system 00 in options (as the first among the terminals). To avoid this, define a dummy terminal in first line, as shown in stdcomopt.

NB: Duetcom rettet, så dette skulle være overflødigt! (iflg NH)

5. REVISIONS IN SKELETON PROGRAM (SKP). 5.

This section describes in several points the revisions and new features of the SKP and the involved revisions in LD and DB descriptions.

5.1 Blocklength of Terminal Work File Selectable. 5.1

In the file section of the DB description, the blocklength can now be chosen without restrictions (but must be an integral number of segments). Remember to have empty work files, if this parameter is changed.

5.2 Changes in DB Description and SODA LD Giving Better Performance. 5.2

The following changes are recommended and will not disturb the functions:

- change block length of log file to 2048 hw.
- change block length of sq files to 2048 hw.
- remove "log after" and "log before" in set 30 in SODA LD.

Note: as the utility programs Teleop and Telerees in release 5 use a fixed blocklength of 2048 hw for the log file, this change is merely a demand, as the information is used by Telelogex.

5.3 Print Keyed In Transaction Text (Block 1). 5.3

In order to be able to point an error message and its corresponding input transaction, some corrections have been made in block 1 in the SKP and some in the local data description.

corrections in the skeleton program.

Two new duetinstructions d302 and d315 have been introduced.

d302 contains an adaption point, which is a new one. This adaption point should be used in order to assign the communication-variable listtrans:

value 0: - the keyed in transaction should not be printed.

value 1: - the keyed in transaction will be printed in the reply area.

default: if error_number <> 0 then then 1 else 0.

The adaption concerning the new adaption point is located in B(adp_block_1,3).

corrections in the SODA-description.

Two new variables have been declared.

```
v206: listtrans: word          ;communication variable
v207: transtext: text.150      ;after a call of algol-operation 13
                               ;this variable will contain the
                               ;keyed in transaction
```

only the first 120 chars will be printed because of limitations in the line buffer size.

5.4 New Printer Administration (Block 1, 3, 6). 5.4

In the administrations of output files and printers, an alternative set of functions has been added.
The new functions support:

- back sequencing in restart of printing.
- better user information after termination of printing.
- user-oriented printer selection.

The new facilities are implemented in SKP by:

- re-coding block 6 (printer transactions).
- changing block 3 (z-transactions).
- re-structuring the involved administration records.
- new variables and changes in set declarations (SODA LD).

The new model solution is based on these facilities. The facilities can be extracted by using the blocks 23 and 64 and part of block 21, 22, 50, 51. The "old" SKP-facilities can still be activated, controlled by communication variable print_system, assigned in B (adp_block_15, 1).

Introduction to the implementation:

When typing a printout transaction, the selection of printer is done like this (SKP transaction UOR):

- standard (nothing keyed in) is the seperate printer. This is selected in ADP1 by printer_index = 0. The seperate printer (for the actual terminal) is not described in a printer adm record, but the describing fields are placed in the terminal adm record.

- if a printer no. is selected in input (in the range 1-10), one out of a maximum of 10 selectable printers (used by the user) is chosen. The interpretation of this printer index is found in a table in the user adm record, which holds the key values for the printer adm records. The printer adm record hold the name of the printer, as stated in Duetcom options, and this must match the key value (for example, printer key = 47, name = 47p).
- central convert may be selected in the ADP1.
- if the UOR is not the first printout order in an output form type, repeat (the whole file) or resume (from last printed position) is selected in ADP1.

5.4.1 Block 1 (adm of trans, changed): 5.4.1

- transacitons generated by Duetcom as result of a printing are not directed to the installation adaption in B(23,1), but B(6,2) is called.

5.4.2 Block 3 (z-transactions, changed): 5.4.2

- create printer adm (ZPO) changed to:

  D51: read_set_no:= 11, adm_type:= 805

  ADP1: B(adp_block_2,3), read printer no

  ADP2: B(21,5), read user no and lookup user adm record (s15)

  get s15, user adm

  ADP3: B(21, 46), check contents of printer table, assign keys for printer adm.

  create s11, i805, printer adm record

  ADP4: B(21, 6), assign printer information

```
          put s11
- modify printer adm (ZPR) changed to:
  D61:    read_set_no:= 11, adm_type:= 805
          ADP1: B(adp_block_2,3), read printer no
          ADP2: B(21,5), read user no and lookup user adm (s15)
          get s15, user adm
          ADP3: B(21,46), check contents of printer table, assign
                keys for printer.
          get s11, printer adm
          ADP4: B(21,19) read information
          put s11, printer adm

- delete printer adm (ZPN) changed to:
  d71:    read_set_no:= 11, adm_type:= 805
          ADP1: B(adp_block_2,3), read printer no
          ADP2: B(21,5) read user no and lookup user adm (s15)
          get s15, user adm
          ADP3: B(21,46), check contents of printer table, assign
                keys for printer
          get s11, printer adm
          ADP4: B(21,27), clear printer entry in user adm: prin-
                ter_table (printerno)
          delete s11, printer adm
          put s15, update user adm

- inquiry, printer adm (ZP?) changed to:
  d101:   read_set_no:= 11, adm_type:= 805
          ADP1: B(adp_block_2,3), read printer no
          ADP2: B(21,5), read user no and lookup user adm (s15)
          get s15, user adm
          ADP3: B(21,46), assign keys for printer
          get s11, printer adm
          ADP4: B(21,42), print information
```

5.4.3 Block 6 (Print Administration, Changed):

The block has been re-coded, and now the SKP supports 2 user transactions: Start printing (UOR) and accept printing (UAC), and the Duetcom-generated transactions.

- sketch of start printing (UOR):

```
d220:   initialize variables
        ADP1: B(adp_block_15,1), read input line
              assign communication variables:
              get_outp, print_function
              get_print, centr_conv
        if get_outp = 1, then d240
        if get_print = 1, then d300
        if centr_conv = 1, then d370 else d380
        put s11 output form adm, if it was read
        put s24 printer adm, if it was read
        return
d240:   assign key variables
        get s11, output form adm
        if textarea2_state = 0, then d254 else d256
d254:   exchange textarea1 and 2
        compute form_inf
d256:   if print_function = 5, then
        B(22,48), print_error = 13, error message
d300:   assign key variables for user adm
        get s15, user adm
        assign key for printeradm
        get s24, printer adm
        if error then B(22,48), print_error = 4
```

```
        if "printer busy", then B(22,48), print_error = 9
        if print_function <3, then d320
d320:   check legality. Errors:
        B(22,48) print_error = 1, 3, 4, 5
        assign start position (in text file)
d370:   algol 8, ask for central convert
        Errors: B(22, 48), print_error = 6, 7, 8
        algol 2, create new text area
d380:   check legality if seperate printer is used,
        Error: B(22,48), print_error = 9
        algol 7, convert on printer
        (seperate or selectable).
        Errors: B(22,48), print error = 13, 14
```

- sketch of accept printing (UAC):

```
d520:   ADP2: B(adp_block_15,3), read inputline
              assign adm_ident2
        assign key variables
        get sl1, output form
        check state (must be 3).
        Error B(22,48), print_error = 10, 11, 12
        reset textarea2: write EM in file
        put sl1, output form
```

- sketch of Duetcom generated trans:

```
d610: initialize variables
        if printerno >0, then d630 else d650
        ADP3: B(23,1), write printing results
        update and put sl1, output form, if it was read
        put sl4, term adm, if it was read
        (and update reply area position)
```

```
        update and put s24, printer adm, if it was read
d630:   (end of printing on selectable printer)
        get s24, printer adm
        get s14, term adm
        (open reply area for writing)
        if "output form was printed" then d640 else d645
d640: assign key variables
        get s11, output form type
d645: assign textarea2_name from term adm
d650: (end of printing on seperate printer)
        if "output form was printed", then d660 else d665
d660: assign key variables
        get s11, output form type
d665: assign textarea_name from term adm
```

5.4.4 Implementing the New Printer Administration. 5.4.4

I. Add the following fields in DB descriptions:

i801 : user

f1027 adm_last_printer	half
f1028 adm_printer_table	word (10)

i802 : term admin

f2031 adm_sep_outp_form	word
f2032 adm_text_file	text t11

i804 : outp form type

```
f4042 adm_txt_a2_start        word
f4043 adm_txt_a2_rollb        word
```

i805 : printer admin

(remove the group output_pool)

```
f5001 adm_pr_name             word
f5002 adm_pr_user_no          word
f5003 adm_pr_term_no          word
f5004 adm_pr_outp_form        word
f5005 adm_pr_table_idx        word
f5006 adm_pr_text_file        text t11
```

II Add the following declarations in SODA LD

Variables (the variable numbers can be changed):

```
v49   : centr_conv            : word
v181  : get_term              : word
v192  : get_print             : word
v183  : get_user              : word
v184  : get_outp              : word
v186  : print_error           : word
v190* : startposition         : word
v182  : saved_outp_form       : word
v185  : saved_user_no         : word
v187  : print_system          : word
v188  : print_function        : word
v189  : printer_index         : word

v594  : adm_last_printer      : word v13 n2
v595  : adm_printer_table     : word v12 n2      user (i801)
```

```
v592  : adm_sep_outp_form   : word     n2      term (i802)
v593  : adm_text_file       : text.11  n1

v1011 : adm_txt_a2_start    : word     n2      outp (i804)
v1008 : adm_txt_a2_rollb    : word     n2

v1186 : adm_pr_name         : word     n2
v1187 : adm_pr_user_no      : word     n2
v1188 : adm_pr_term_no      : word     n2    printer (i805)
v1189 : adm_pr_outp_form    : word     n2
v1190 : adm_pr_table_idx    : word     n2
v1191 : adm_pr_text_file    : text.11  n1
```

Record sets:

set 11 (used for create, modify all - plus update outp_form):

```
v594*<=>  f1027     user (i801)
v595*<=>  f1028

v592*<=>  f2031     term (i802)
v503*<=>  f2032

v1011*<=> f4042     outp (i804)
v1008*<=> f4043

v1186*<=> f5001
v1187*<=> f5002
v1188*<=> f5003     printer (i805)
v1189*<=> f5004
v1190*<=> f5005
v1191*<=> f5006
```

set 14 (used for updating fields in term.adm)

v592 < > f2031
v593 < > f2032

set 15 (used for get used_adm)

v594 < f1027
v595 < f1028

set 24 (used for updating printer adm)

v1186< f5001
v1187< f5002
v1188< > f5003
v1189< > f5004
v1190< f5005
v1191< > f5006
v47 < f5001

III. Load the new SKP

IV. Copy the following parts of the model

Block 21, entry points 5, 6, 19, 27, 42, 46
Block 22, entry point 48
Block 23, all
Block 50 (adp_block_1): change conversion of line code to skp_line_code (d75 and d115).
Block 51 (adp_block_2), entry point 3, read keys for printer admin (d500, d505)
Block 64 (adp_block_15), all.

5.5 Read Hotnews - New SKP Transaction (Block 3). 5.5

SKP line code: ZM2.
Sketch of SKP and ADP:

```
d180:   ADP1: B(21,51) read hotnews to a variable.
        return
```

5.6 Correct User DB Record - New SKP Transaction (Block 3). 5.6

The transaction is primarily intended for selective correction of fields in user DB records. The fields concerned are for example the counter fields, which by some application malfunction might get an errornous value. SKP line code: ZM1.
Sketch of SKP and ADP's:

```
d160:   read_set_no:= 1, adm_type:= 801
        ADP1: B(adp_block_2,3), read user ident to variable
              adm_ident1
        lookup s11 (user)
        if error: B(22,46), and exit p1
        adm_user_no:= adm_ident1
        read record type and assign types, according to:
        input          assigned to      read_set_no
        501 or 502     dc_part_dc_type      8
        601 or 602     it_type              3
        701 or 702     od_type             21
        else: B(22,47) and exit p1
        ADP2: B(adp_block_2,3) read record ident to relevant
              variable.
        put_record:= 1
        get s(read_set_no), get user record
```

```
        ADP3: B(21,50) modify record
        if put_record = 1, then d168
        return
d168:   put s(read_set_no), put user record
```

5.7 Check of Secondary Master in Structure List Creation (Block 4). 5.7

The change affects both the parts list and the DC-structure list. In a new ADP, the secondary master is available for check of relevance and consistity.

A new communication variable,

v169: structure_ok: word

is introduced (in SODA LD and skeleton program).

A new adaption point is introduced (in skeleton program)

B(adp_block_3,29); check secondary master

If you want to use the new skeleton program, you have to declare this new variable and insert the new entry point in adaption block 3 (in the present model solution block 51).

The old duet instruction d180 in block 4 is replaced by two new instructions d180 and d181:

```
d180:   get s(skp_set3)      ; fetch secondary master
    >   modify structure_ok:= 1
                put_record:= 0
                s
```

```
    >   execute B(adp_block_3,29)      ;check sec. master
                s
    >   if structure_ok:= 1 then d181
d181:   put s(skp_set1)      ; put list record
    >   if error <> 0, then d189
    >   modify no_of_parents:+ 1       ; update record counter
                no_of_parts:+ 1
                s
    >   put s(skp_set3)      ; put secondary master
```

5.8 Secondary Master available in inquiries on Structure Lists (Block 5). 5.8

It is now possible to have the secondary master available in any case (not just, when alfanumeric ident is used). The change affects both parts list and DC-structure list.

The variable

v648 : admin_alphanum possible : word

is removed (from SODA LD and skeleton program).

A new communication variable

v 648 : sec_master_wanted : word

is introduced (in SODA LD and skeleton program).

In block 5 (Inquiry) duet instruction d111, a new statement is added just after "> get s(skp set2)":
modify sec_master_wanted:= 0

In d115 just after "> if scan_result = 2 then d117":
if sec_master_wanted = 1, then d30

"d122: if admin_alphanum_possible = 1, then d30"
is corrected to:

d122: if sec_master_wanted = 1, then d30

5.9 Creation of Rebate-Records in Order-Voucher-end (Block 7). 5.9

The communication variable order wanted may be assigned = 2 in ADP B(adp block 17,14), and a new rebate record is created by the SKP, without using the normal order-line-creation procedures.

5.10 Refinement of Order Invoicing (Block 7 and 9). 5.10

In order to destinguish between rebate records originating in different orders, the SKP is changed in block 9. A new communication variable is intruduced:

order_no long

It should be assigned to order number in ADP B(adp_block_17,14), when a creation of a rebate record is wanted (see above).

5.11 Errors in Counter Field Updating (Blocks 7, 8, 9, 10). 5.11

All known errors have been corrected.

5.12 New ADP in Calculation and Display Voucher (Block 7). 5.12

A new adaption point:

B(adp_block_17,26), called from B7.

The ADP is called after the scan of the voucher lines. Intended for printing of calculated voucher totals. Called unconditionally.

5.13 Fetch of parent dc record at Voucher Start (Block 8 and 9). 5.13

Introduced in create order (block 8) and invoice voucher start (block 9). A new communication variable is intruduced:

parent_dc_wanted.

A new set is defined (s34), working on the dc_structure_file. Two new adaption points are introduced to process the parent record.

- in create order:

A new communication variable has been defined

parent_dc_wanted.

When assigned to 1 in the adaption point (ADP2 or ADP3), the dc structure list will be accessed and the "parent" dc record fetched. This is done by a new set for the list, s34 - the parent dc record is accessed by s33. Default value for the communication variable is zero.

Besides, a new adaption point has been added to process the "parent" dc record:

ADP4: B(adp_block_6,3).

The entry point has to be added in the relevant block(s), e.g. block 55. ADP4 is activated after put s16 and after fetch of the parent dc record (s33). No next ADP. Use of ADP4: transfer of in-

formation from the parent dc record to the voucher start and/or order master record.

- In invoice voucher start:

A new communication variable has been introduced

parent_dc_wanted

with the same use as in create order (see above). A new adaption point has been added:

ADP4: B(adp_block_9,3).

The new entry point must be added to the relevant block(s), e.g. block 58. The use of ADP4 is exactly as in create order.

5.14 Use of Parts Lists in Order Invoicing (Block 9). 5.14

In SKP block 9 a new ADP is added. In the ADP B(adp_block_12,2), the communication variable part_order_lines may be assigned = 1, activating a scan through the parts list and reading all part items before invoicing the parent item. After reading each part item, the new ADP is called: B(adp_block_12,5) - check part item. There the available stock quantities should be checked.

5.15 Single Transaction for Order Invoicing, Select Voucher Number Series (Block 9). 5.15

The SKP block 9 is changed, so that the variable admin_vouch_ident_idx (selecting the proper voucher number series) is assigned a standard index (dependent of voucher type), as for the corresponding invoicing types (and at the same place).

5.16 Single Transaction for Order Invoicing, update invoice start record (Block 9). 5.16

After creation of the invoice start record, this record is now available for updating with order master information. A new ADP is called just after the order master is read and before the scan of order lines.

A new line is added in instruction d286. 2 new Duet instructions d384 and d385 are added.

In the new Duet-instruction d384 the (invoice) voucher start is fetched and a new adaption point - (adp_block_12,10) ; update voucher start - is executed.

The new adaption point is ment for updating of the voucher start record and possible assign of different adm- and wf-variables with information from the order master record.

NOTE: The new entry point adp_block_12,10 must be added in the beginning of adaption block 12 (block 61).

5.17 Use of Parts Lists in Invoicing (Block 9). 5.17

After the scan through the parts list and creation of part item line records, a new optional scan of the part item line records (and fetch of the corresponding part item records) is added. In this scan, the item lines and the items may be updated with for example new stock quantities, in case of stock shortage in some part item.

The duet-instruction d220 has been amended.

Two new duet-instructions d221 and d222 have been added.

The adaption point 6 (in d220) is now placed in adaption block 11, entry point 1:
execute B(adp_block_10,5) is changed to
execute B(adp_block_11,1),

The communication variable parts_list_wanted must be assigned in adaption point 6.
value: 0 - scan of part invoice lines not wanted (standard value).
1 - scan of part invoice lines wanted.

A new adaption point (7) is executed from the new duet-instruction d222. The adaption point is ment for updating of the information in the part invoice lines and in the part item records. The adaption point 7 is called
B(adp_block_11,2).

Two new skeleton variables has been declared in SODA:

v222 : save_it_type
v223 : save_it_ident

They are used internally in SKP.

An important addition in SODA LD must be inserted in Set 23:

v306 < f301
v307 < f302

5.18 Deletion of Voucher Lines after Part List Execution in Create Item Line (Block 9). 5.18

The restrictions on voucher line scan before deletion are now properly assigned by calling the existing ADP in B(adp_block_16,6).

5.19 Assignment of Serial Number to Account Entry Records (Block 9). 5.19

An error has been corrected.

5.20 Special Account Entry Line for Batch System (Block 10). 5.20

A new SKP transaction is implemented: KX. The purpose is to write an account record on the log file without updating the account entry list file. This is done by writing a record in the terminal work file.

Sketch of the transaction KX:

```
d165:   create s23 (i7571), entry line
        posting_wanted:= 1
        ADP1: B(adp_block_18,3) read input.
              assign field variables
        if posting_wanted = 1 then put s23.
```

5.21 Extensions in Account Entry Functions (Block 8, 9, 10, 12). 5.21

The extensions are described below. New transactions are implemented and the existing ones modified. The old functions are still present, using the old ADP's.

New communication variables:

v174 : gen_before_scan
v175 : erase_scan
v176 : new_record_wanted
v177 : printout_wanted
v936 : scan_type
v935 : recordpos
v42 : acc_entry_line_pos
v40 : update_counter

New ADP's in balancing if account:

	block, entry	
ap2	adp_block_18, 20	read inputline
ap3	adp_block_18, 21	
ap4	adp_block_18, 22	terminal scan
ap5	adp_block_18, 23	

New ADP's in connecting of acount:

	block, entry
ap2	adp_block_18, 25
ap3	adp_block_18, 26
ap4	adp_block_18, 27
ap5	adp_block_18, 28
ap6	adp_block_18, 29
ap7	adp_block_18, 30

New ADP's in cancellation:

```
adp_block_16, 27                dummy
```

- a new entry point in block 22 has been added.

  This entry point is used in balancing and correcting functions within account entry processing.

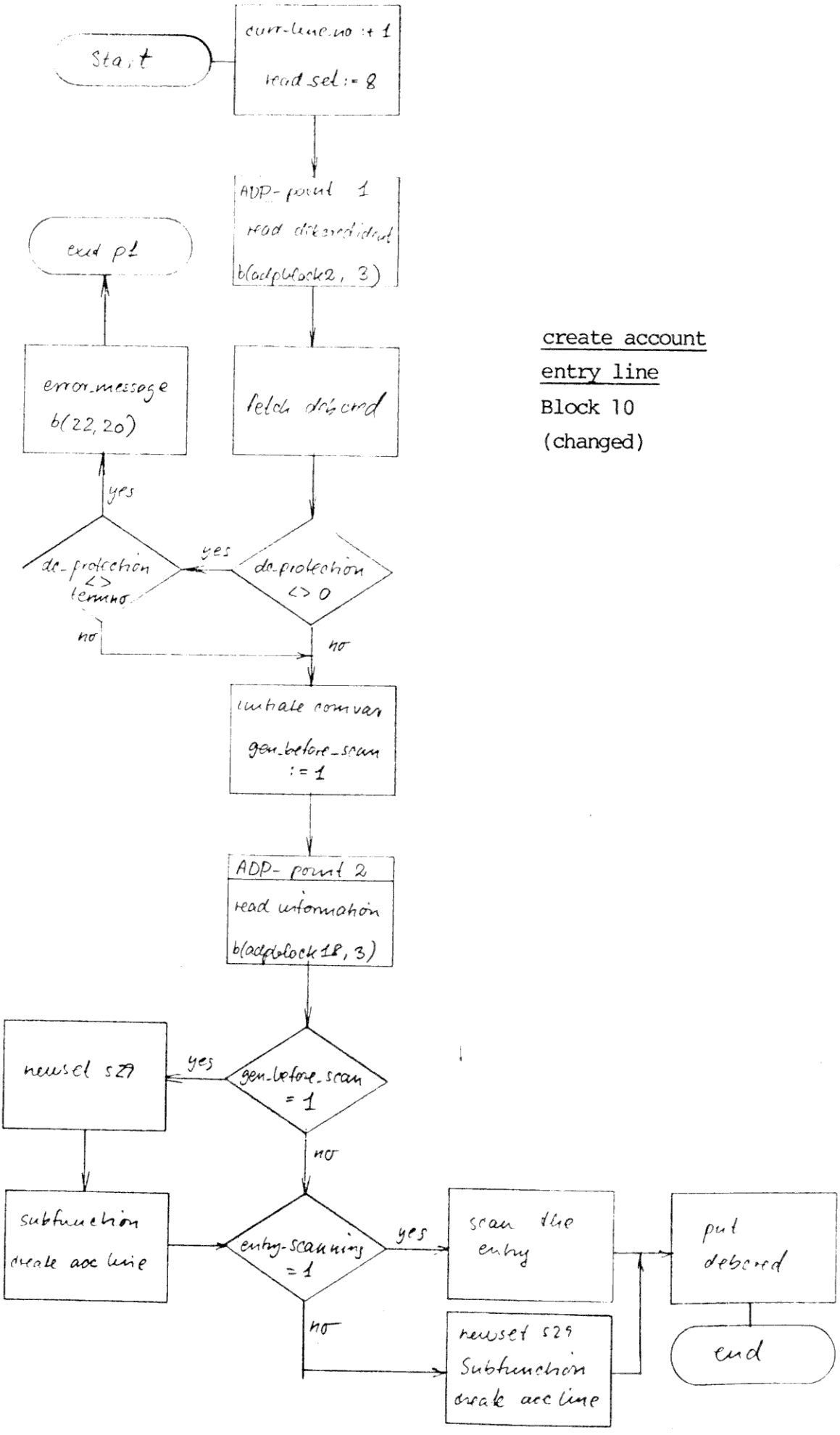
```
d278:   execute   b(22,29)              ; error message
```

- because of the size of skpblock8, all code concerning cancellation has been moved to skpblock12.

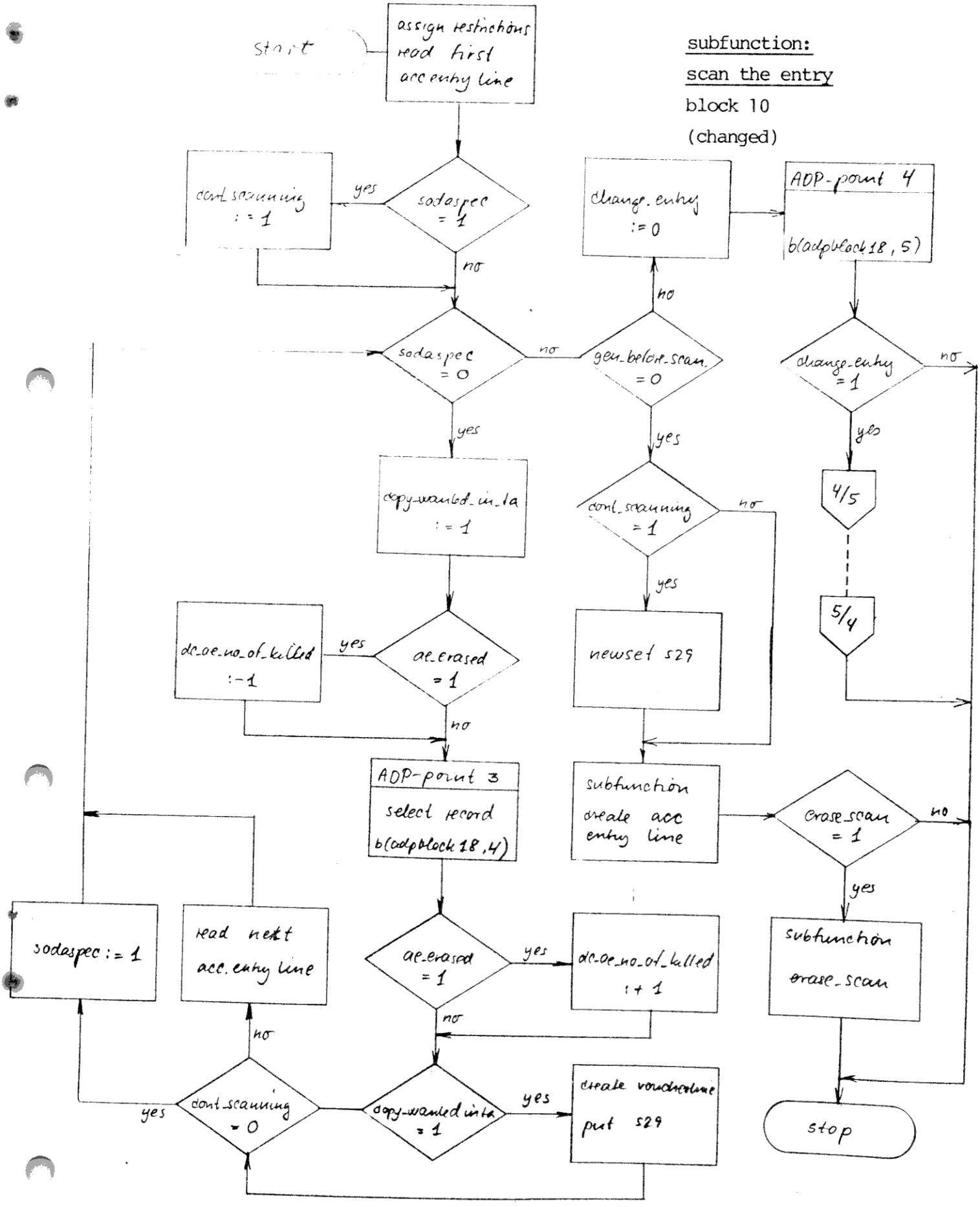
- an adaption point has been added in duet-instruction d962, block 12

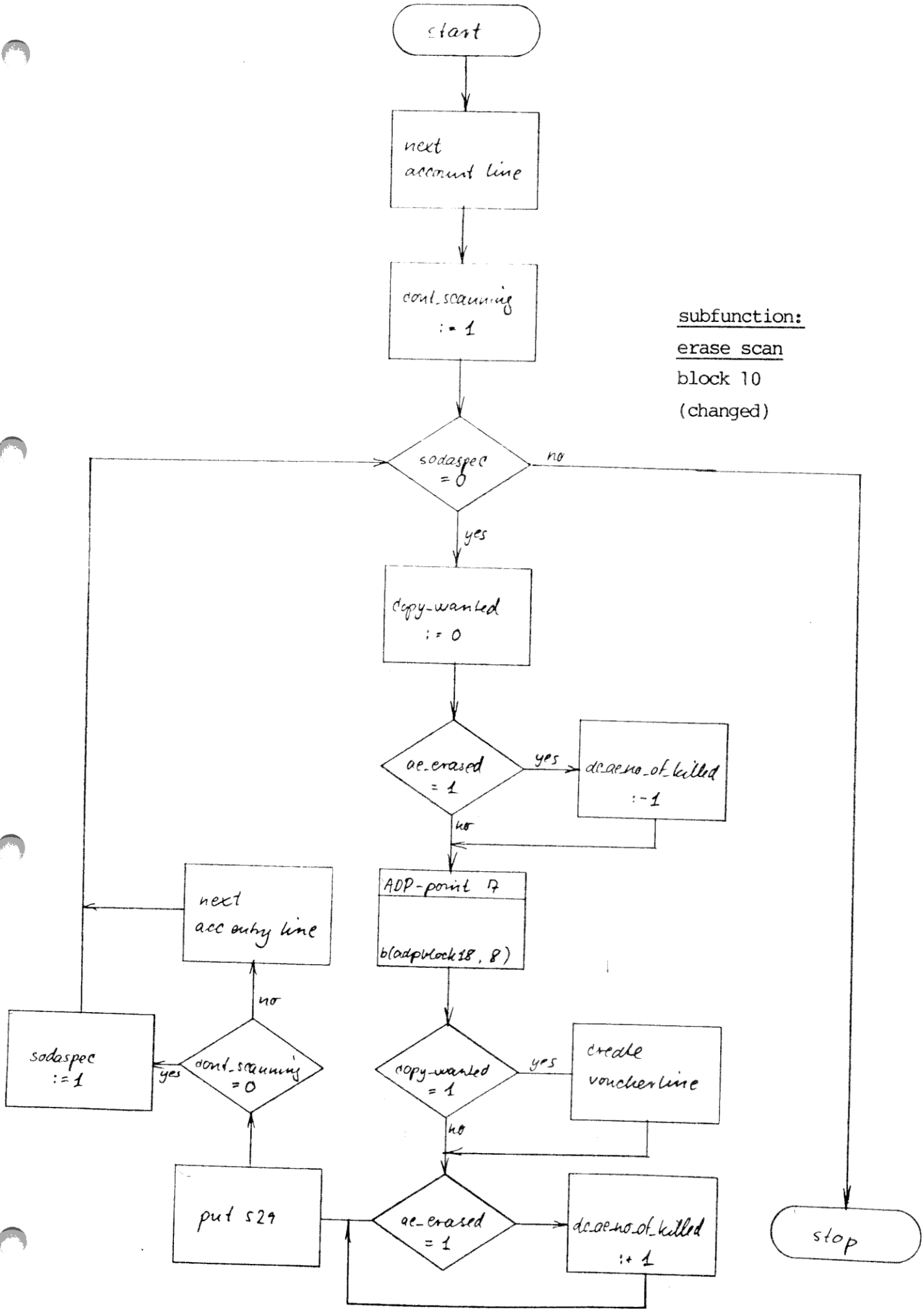
```
d962:   modify gen_before_scan:= 1
        s
      > execute b(adp_block_16, 27)
        s
      > if gen_before_scan = 1, then d965 else d980
```

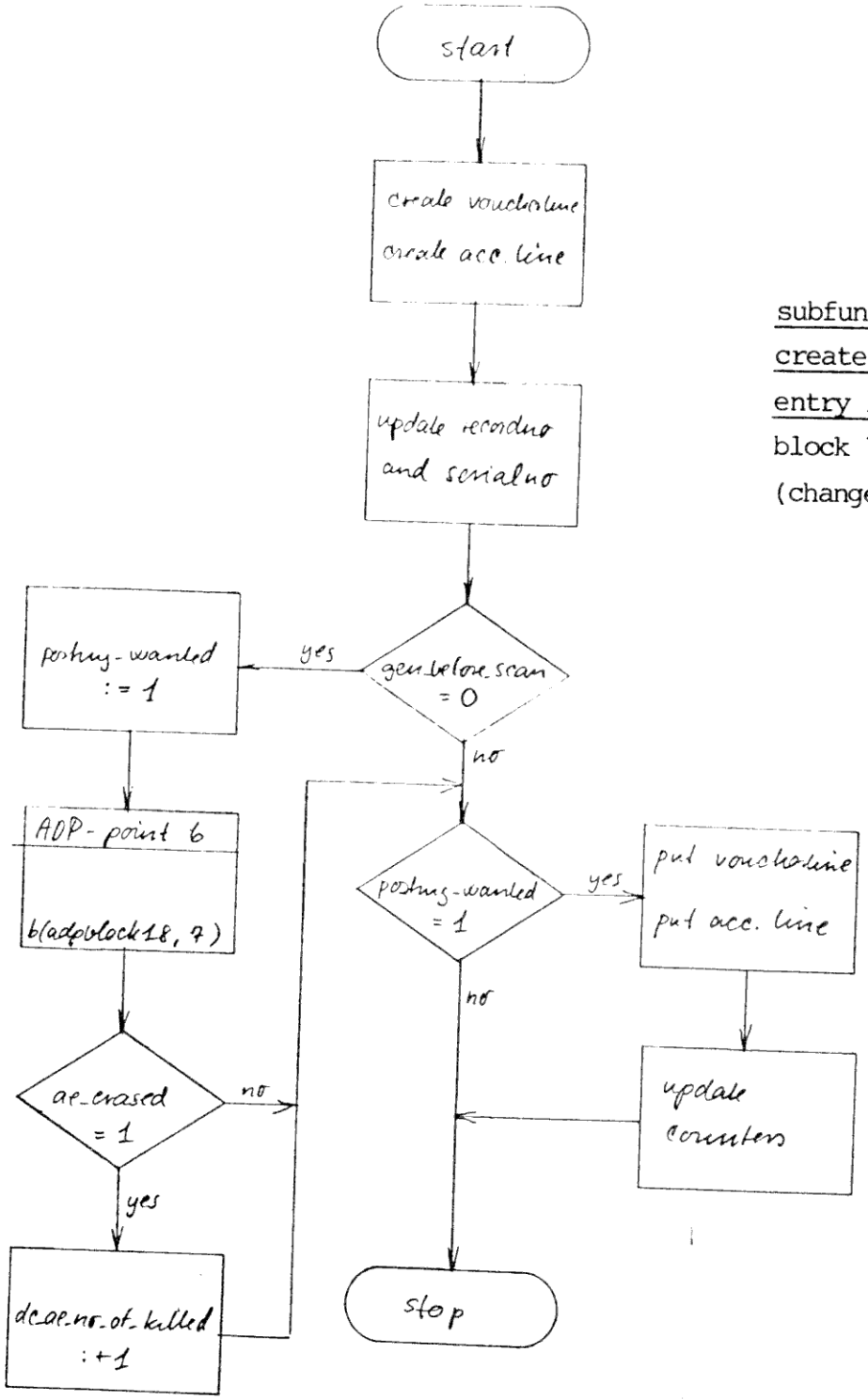
In this adaption point you have to decide, whether you use the old or the new strategy for inserting the account entries, i.e. old or new version of account entry functions.



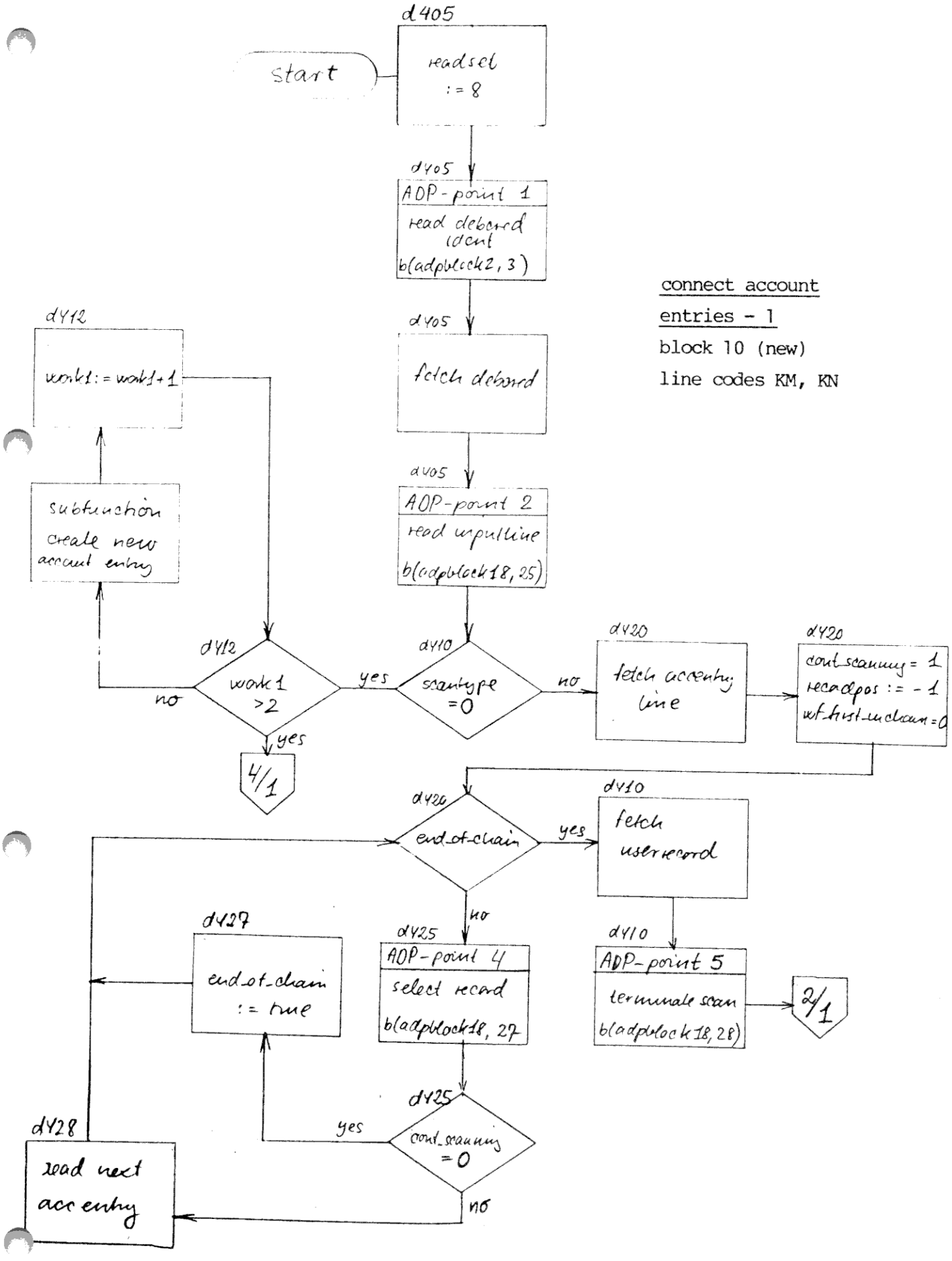
create account
entry line
Block 10
(changed)


start
assign restrictions
read first
acc entry line
sodaspec
= 1
yes
dont_scanning
:= 1
no
sodaspec
= 0
no
yes
copy_wanted_in_ta
:= 1
dc_ae_no_of_killed
:-1
yes
ae_erased
= 1
no
ADP-point 3
select record
b(adpblock 18,4)
sodaspec := 1
read next
acc. entry line
no
dont_scanning
= 0
yes
ae_erased
= 1
yes
dc_ae_no_of_killed
:+ 1
no
copy_wanted_in_ta
= 1
yes
create vorubergehend
put 529
change_entry
:= 0
no
gen_befor_scan
= 0
yes
dont_scanning
= 1
no
yes
newset 529
subfunction
create acc
entry line
erase_scan
= 1
no
yes
subfunction
erase_scan
stop
ADP-point 4
b(adpblock 18, 5)
change_entry
= 1
no
yes
4/5
5/4
subfunction:
scan the entry
block 10
(changed)


start
next account line
dont_scanning := 1
subfunction:
erase scan
block 10
(changed)
sodaspec = 0
no
yes
copy-wanted := 0
ae_erased = 1
yes
dcaeno_of_killed := -1
no
ADP-point 7
bladpblock18, 8)
next acc entry line
copy-wanted = 1
yes
create voucherline
no
sodaspec := 1
yes
dont_scanning = 0
no
put s29
ae_erased = 1
dcaeno_of_killed := 1
stop



subfunction:
create account
entry line
block 10
(changed)

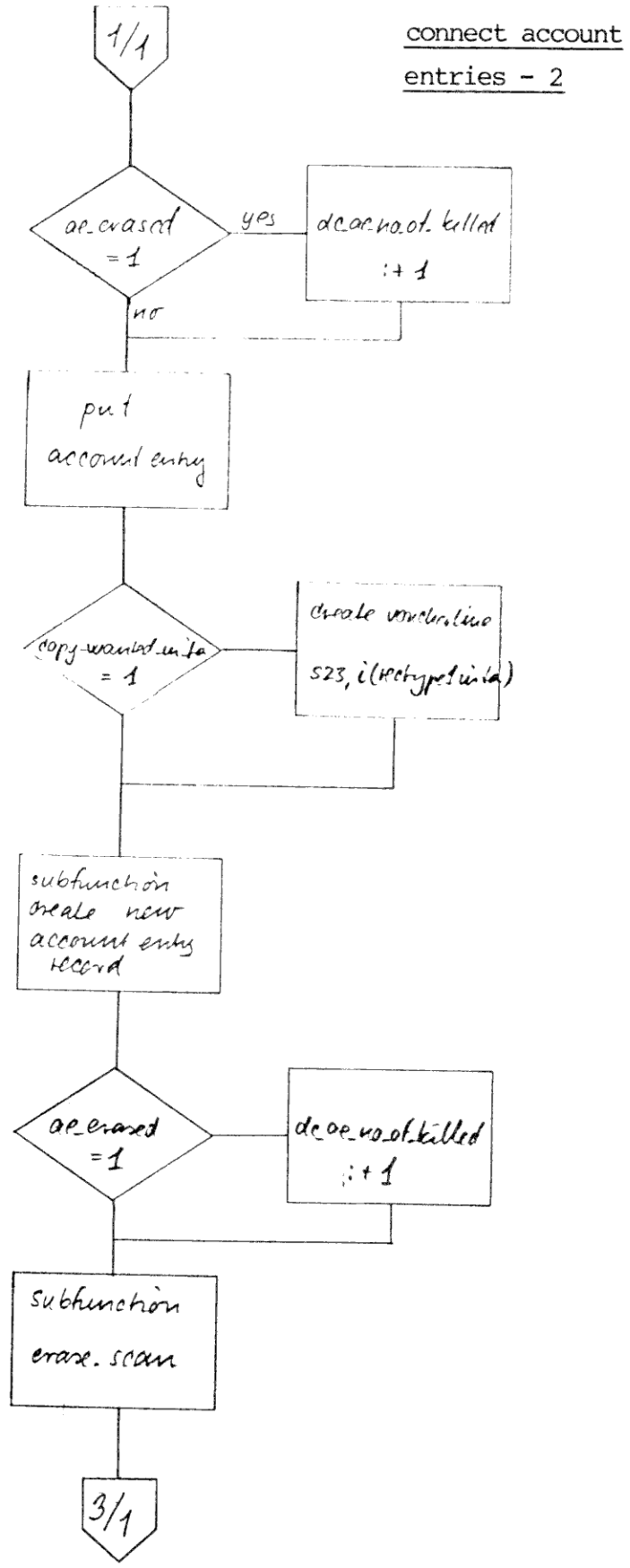


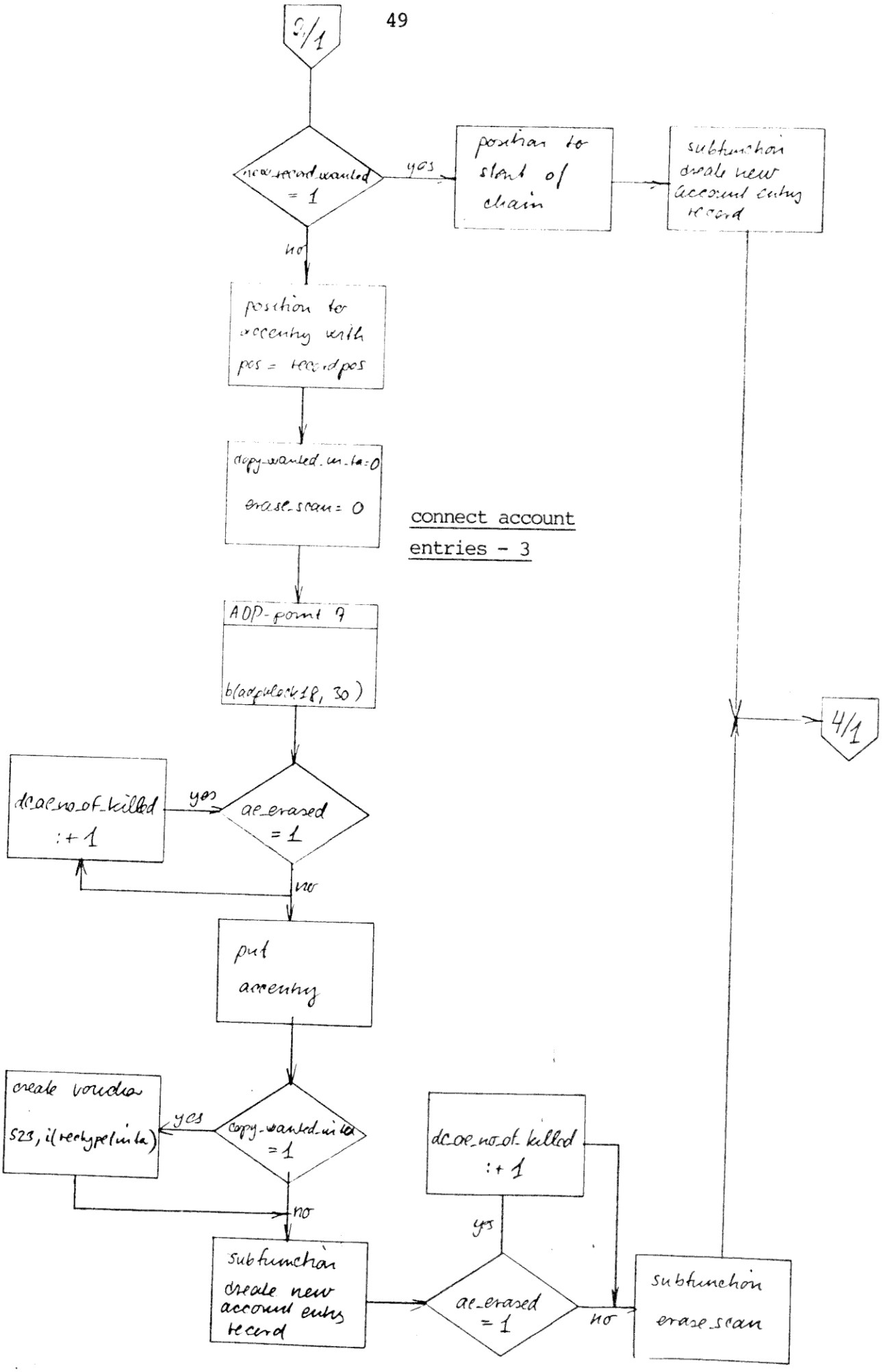
connect account

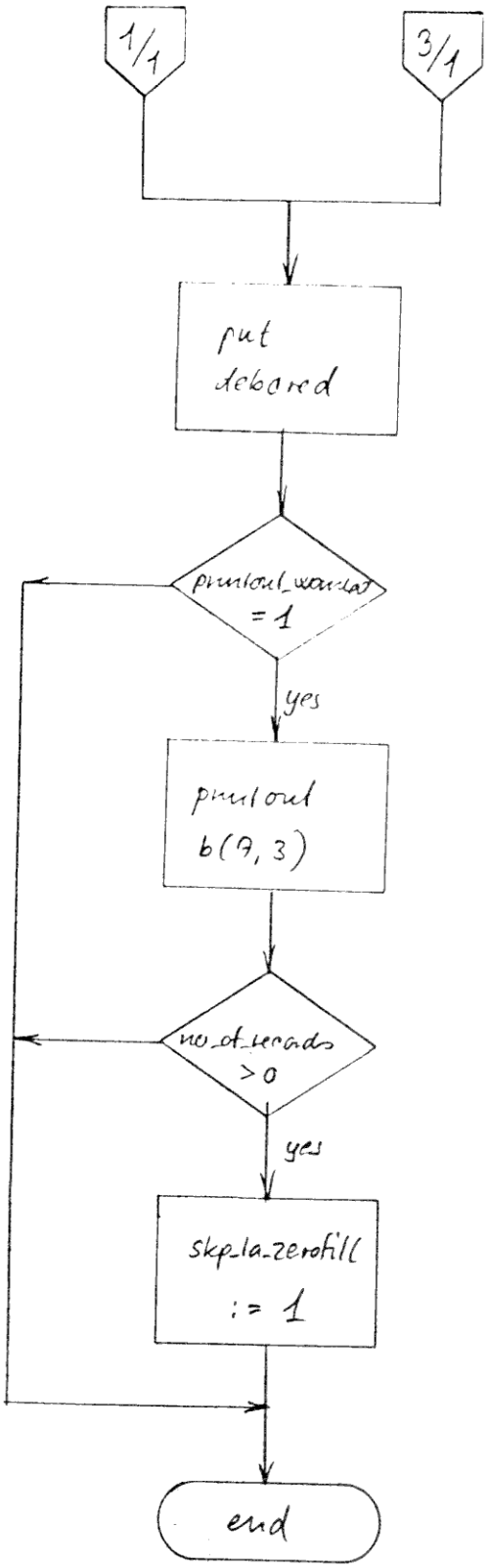
entries - 1

block 10 (new)

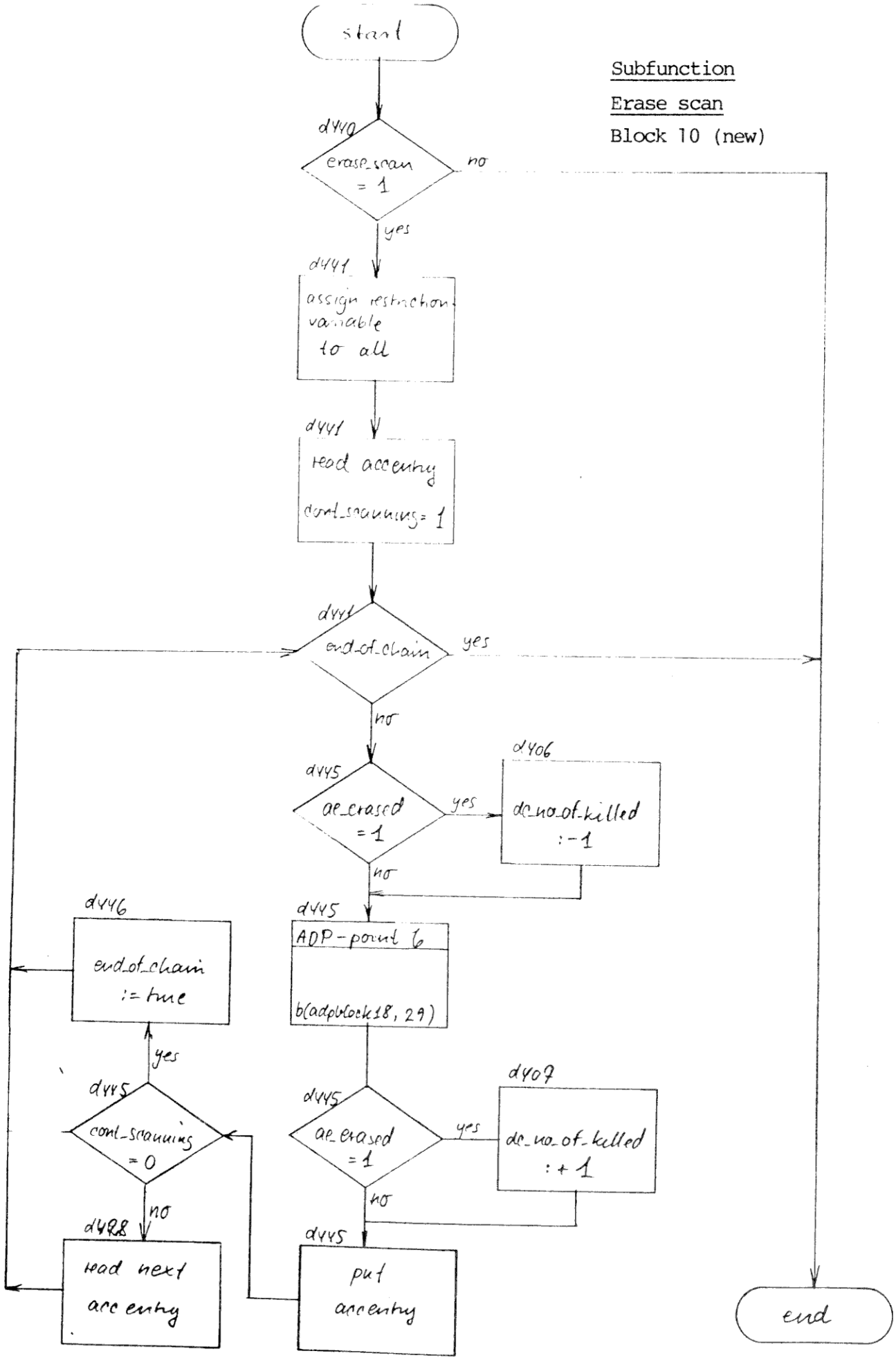
line codes KM, KN


connect account
entries - 2
1/1
ae_erased
= 1
yes
dc_ac_no_of_killed
:+ 1
no
put
account entry
copy_wanted_info
= 1
create connection
523, i(rectype1_info)
subfunction
create new
account entry
record
ae_erased
= 1
dc_ae_no_of_killed
:+ 1
subfunction
erase_scan
3/1


3/1
new_record_wanted = 1
yes
position to start of chain
subfunction create new account entry record
no
position to account with pos = record_pos
copy_wanted_in_ta = 0
erase_scan = 0
connect account entries - 3
ADP-point 9
block(ta_id, 30)
4/1
dc_ae_no_of_killed :+ 1
yes
ae_erased = 1
no
put account
create voucher 523, if(rec_type in ta)
yes
copy_wanted_in_ta = 1
dc_ae_no_of_killed :+ 1
no
yes
subfunction create new account entry record
ae_erased = 1
no
subfunction erase_scan



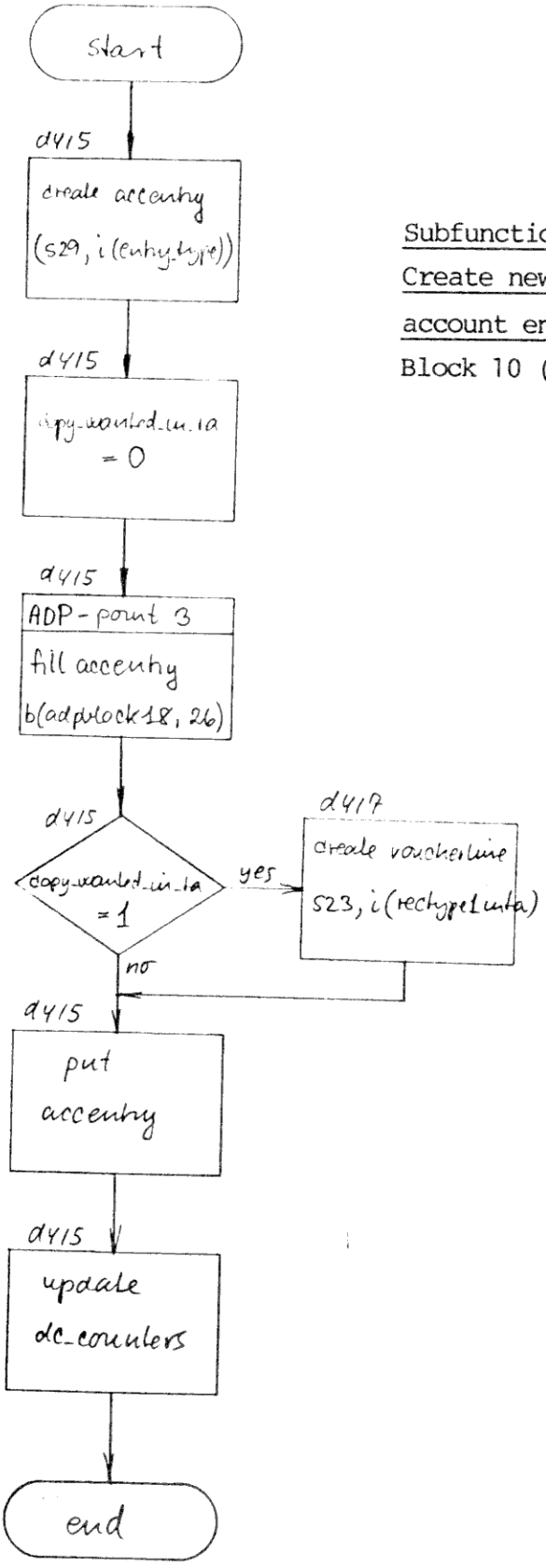
connect account
entries - 4



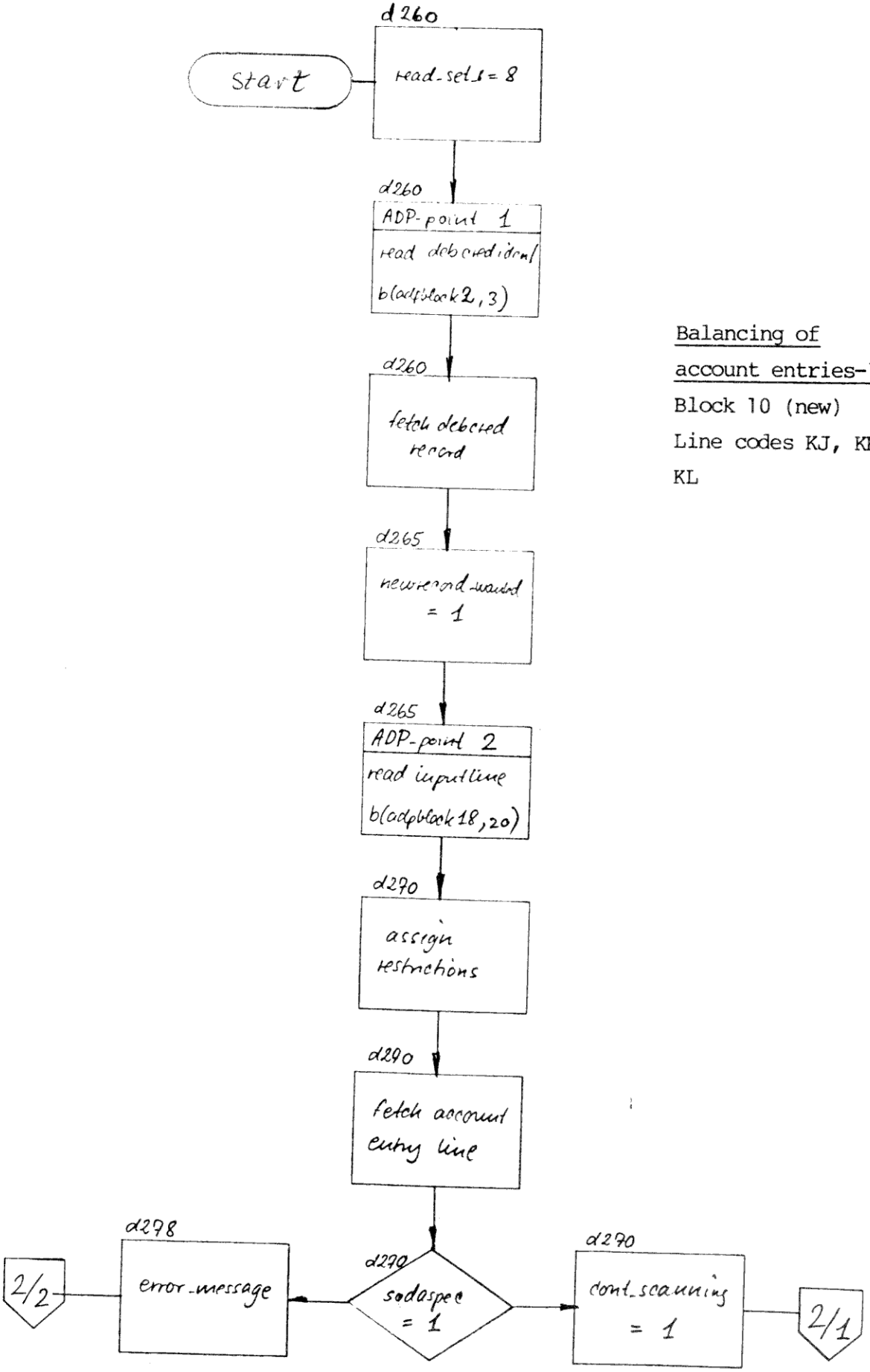
Subfunction

Erase scan

Block 10 (new)



Subfunction:
Create new
account entry
Block 10 (new)

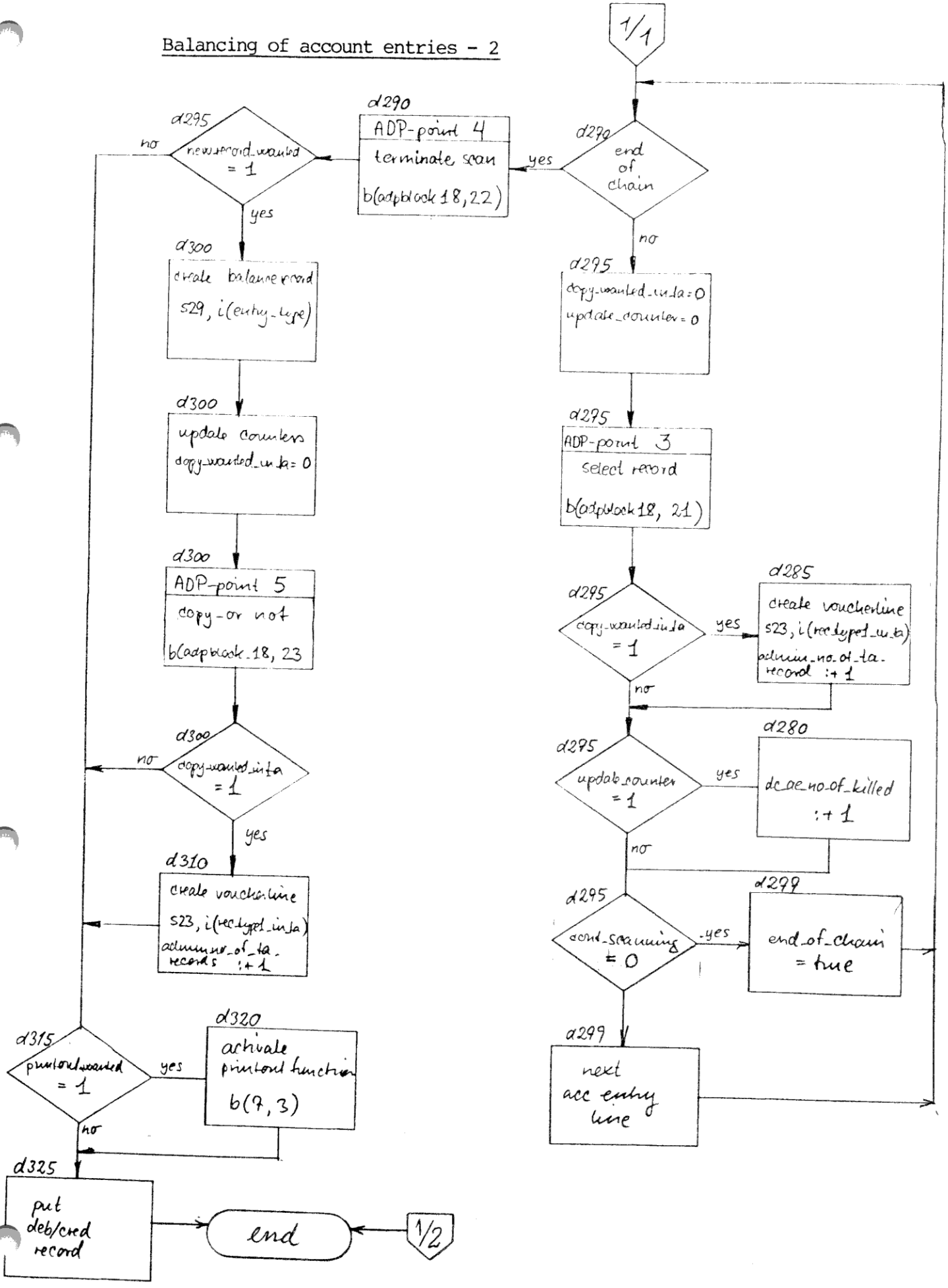


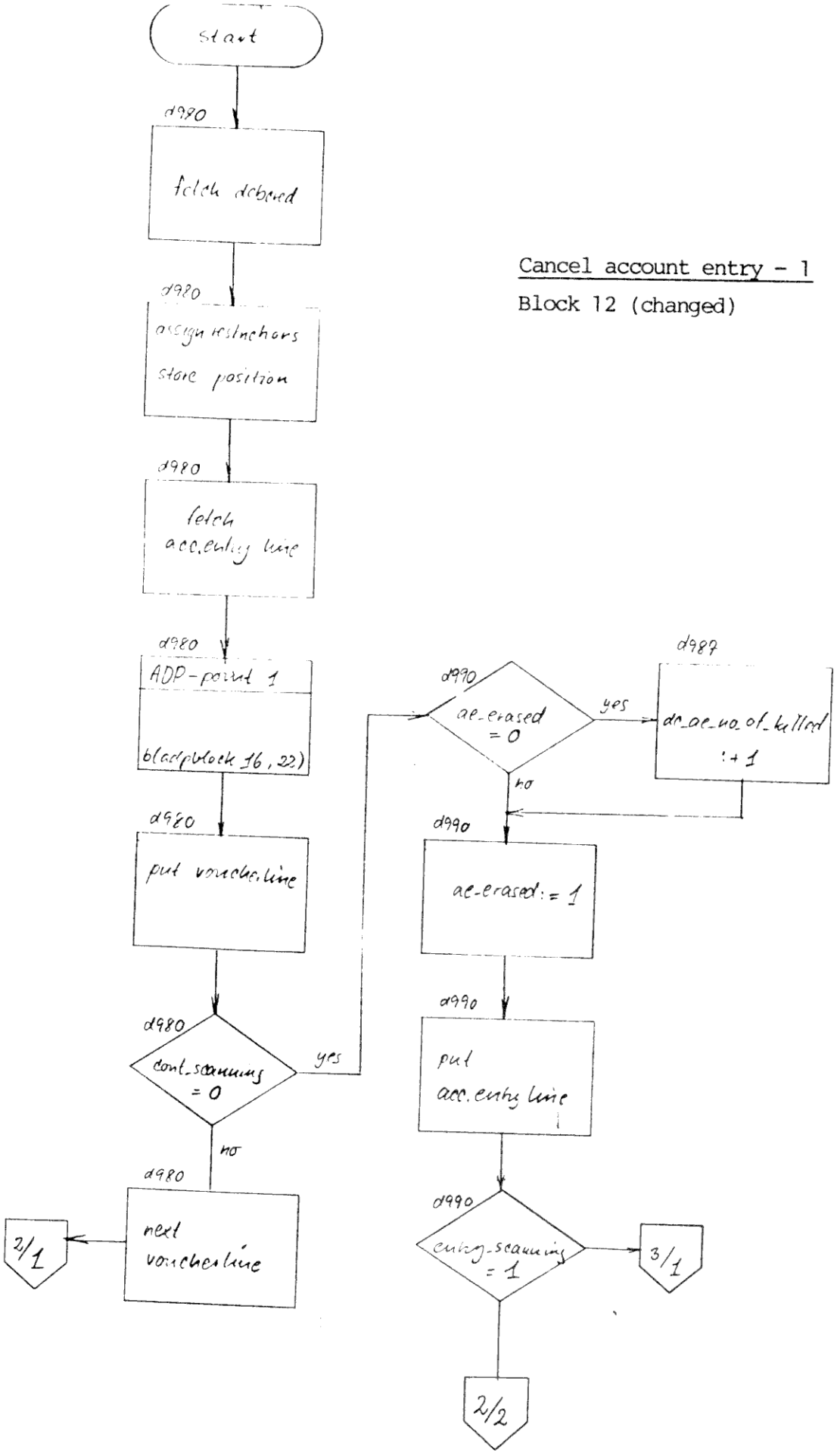
Balancing of account entries-1

Block 10 (new)

Line codes KJ, KK, KL

Balancing of account entries - 2

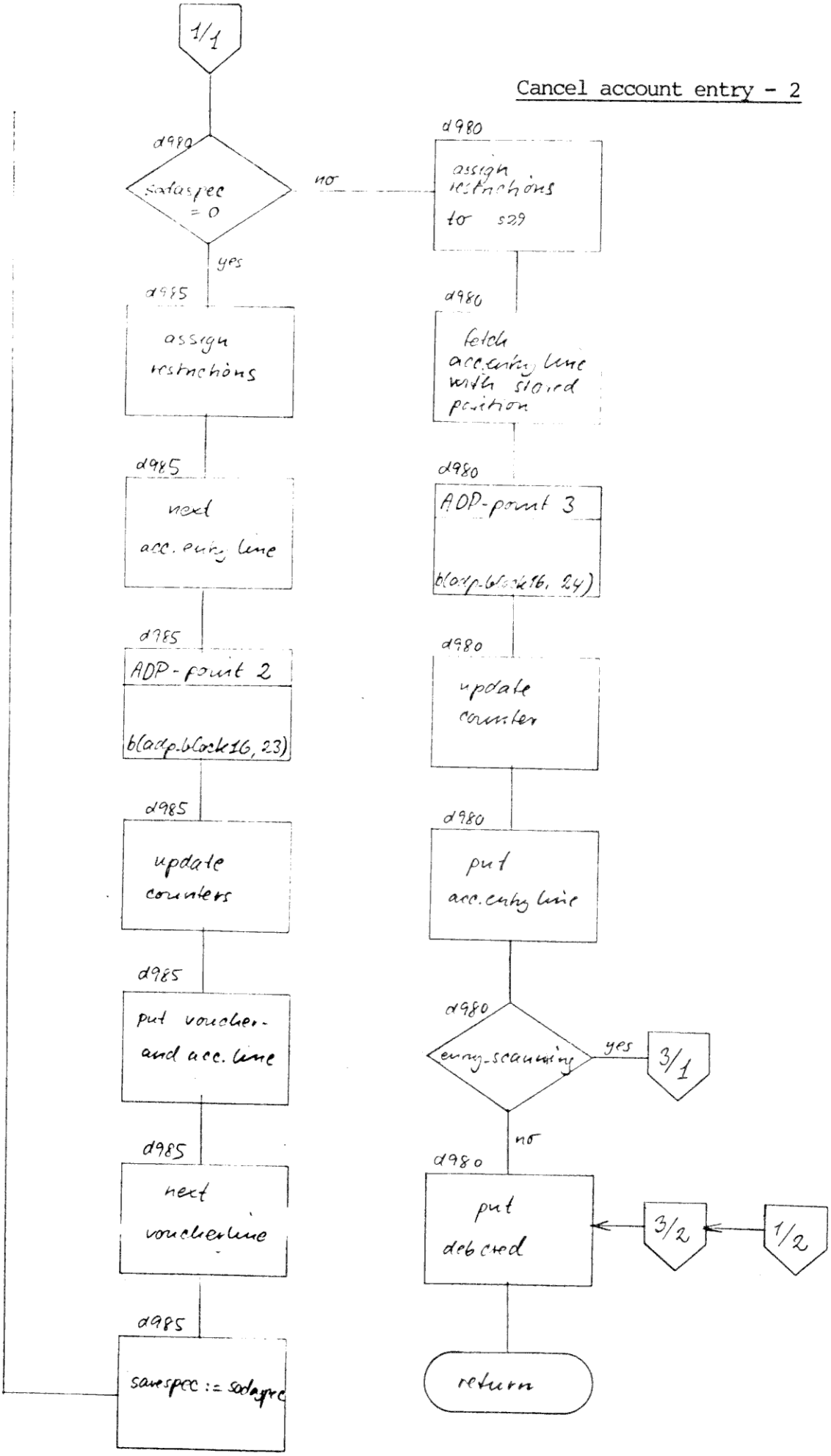


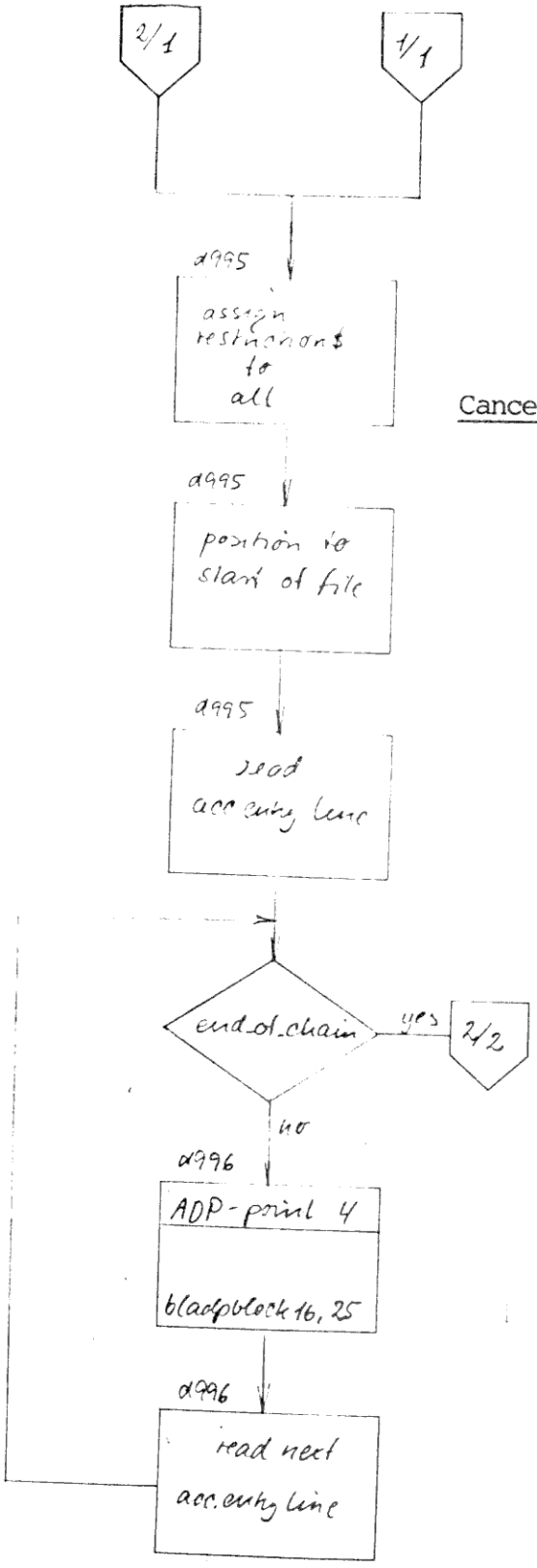


Cancel account entry - 1

Block 12 (changed)

Cancel account entry - 2





Cancel account entry - 3

6. MODEL SOLUTION. 6.

The model solution has been extended at several points. The functions are described in RCSL 21-T028: Teledata User's handbook. The extensions are:

- credit notes for both deb and cred.
- dynamic standard values in creation of item and dc records.
- dc-structure list active in invoicing.
- improved comment output after transaction processing.
- addition amount may be keyed in at voucher lines.
- discount calculation extended with fixed pct. at both item line level (pct. from item record) and voucher level (pct. from dc record).
- stock updating in single transaction.
- extended facilities for control of back order creation.
- short name in dc and item displays.
- automatic order creation possible in delivery costumer vouchers, and single transaction accounting of the order.
- advanced account entry facilities.
- error numbers printed in transaction list report.
- account entry restricted to dedicated terminals.
- hot news facility.
- foreign currency in all vouchers.
- alphanumeric keys in dc-records.
- single transactions for order invoicing, delivery and accounting on any order type.
- orders are calculated exactly as invoices.
- stock profile transaction.
- parts list processing fully implemented in all functions of invoicing and order entry.
- new extended printout facilities.
- single and doubble buffer input mode.

- keyed in transaction displayed at errors.

It should be noted that the system creation files are modified, and some of the changes are of interest beyond the new model solution:

- the command file com2rerum is changed (cfr. section 4.7, rerun.yes)
- the command file com2op is changed, so that current output is on the console. There are 2 good reasons for this: ensuring, the output is printed, even if the process is stopped (here people usually forget to print admlog), and better supervision of the system, ~~which is important after implementing the trap in TELEOP (cfr. section 4.7).~~ NOT RELEASED

7. KNOWN ERRORS IN TELEDATA (ERROR REPORTS). 7.

- (TD 211) missing space in printing a word containing 2 characters, occupying 3 positions (both DUET and GENIUS). Detailed inf. missing.

- (TD 223) CF alarm "ch.head 18" from TELEREADCF. Caused by illegal mother-references in a list file record. Remedy: reestablish. Cause: unknown, maybe hard error on disc. Note that the CF alarms are now more informative.

- Several error reports are received on alarms in PRINT80. The program is now being revised, but unfortunately it is not ready for release.

Ny version ud alligavel

Appendix A: Recommended user structure