RCSL No: 30-M 81

Edition: april 1977

Author: Peter Koch Andersson

Title:

D S C 801

MICROPROGRAM

RCSL No 42-i 2164

A/S REGNECENTRALEN
af 1979

Keywords:

(24 printed pages)

Abstract:

Printed by A/S Regnecentralen af 1979, Copenhagen

RCSL No 42-I 2164

CONTENTS PAGE

Drawings

1. MICROPROGRAM DESCRIPTION. 1.

1.1 General. 1.1

The advantages gained by microprogramming the central logic instead of using the conventional hardwired 'random logic' are:

1: The possibility of compressing complex functions into a relatively delimited space, making the device rather intelligent. And

2: Dividing the problem of functional proof into two partial proofs:

   a: that the rather simple hardware executes the specified microcommands properly and

   b: that the microprogram is correct.

It often happens that these partial proofs are easier carried out than the total.

The microprogram in the DSC 801 is implemented as a horisontal (non encoded) microprogram, that is each outputline or group of output lines controle their specific function in the controller.

This approach requires a rather broad micro instruction word (52 bits) but creates the possibility of

1: a high clock frequency and

2: performing more functions in parallel.

One of the characteristics of the DSC 801 microprogram concept is the controle of the microprogram itself. Each micro instruction contains a next microprogram address specifier, i.e. subsequent instructions may be arranged in an arbitrary way in the microprogram store, no explicit jump- or branch-instructions are required. A similar micro instruction sequencing may be studied in the RC4000 CPU.

2. SEQUENCING.

The microprogram store is 512 words by 52 bits wide, hence 9 bits are required for addressing one word. These 9 bits are stored in the microprogram address register connected to the address lines of the store. The contents of the register is derived from the microprogram itself. The 7 most significant bits are specified directly in the instruction word, whereas the two least significant bits reflect specific states in the controller selected by two three bit fields in the instruction. Even these two address bits may be specified directly in the instruction by selecting one or two of the corresponding address outputs.

The deterministic address outputs are named NEXT MP ADDR (0:8) and the conditional branch selector lines: COND SEL 0 (0:2) and COND SEL 1 (0:2). A third condition selector is supplied, specially used for branching on errors. The effect of a selected 'true' on this selector is to force a one into each of the two least significant address bits regardless of the values else specified for these. These selector controles are named ERROR CONDSEL (0:2).

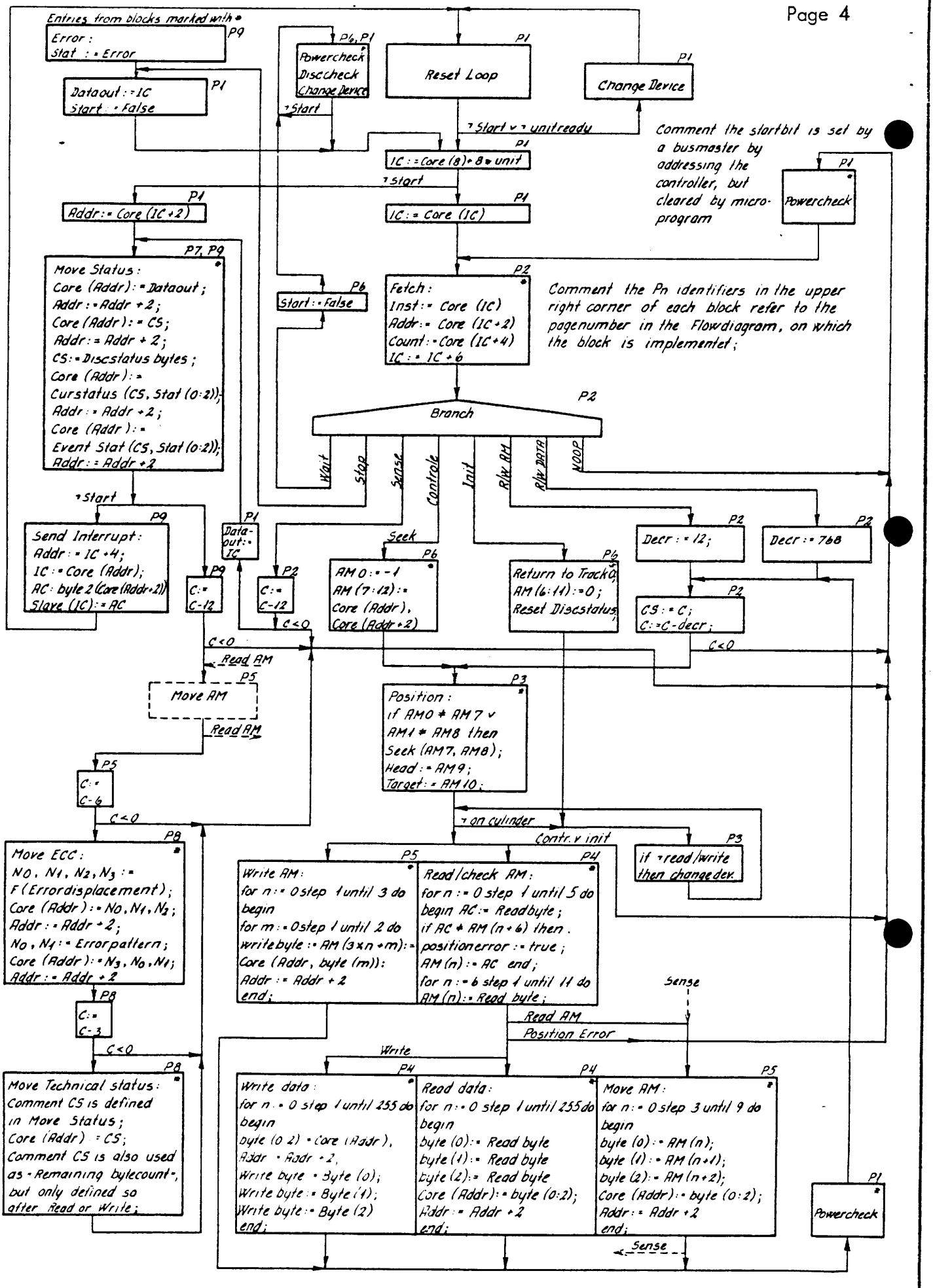
A table over the various branch conditions is supplied in Dwg. No. R21058.

3. MICROPROGRAM BLOCK DIAGRAM. 3.

In the block diagram a survey is given over the major functions and control flows in the microprogram.

The notation used is a selfexplaining mixture of graphic flow-diagram and ALGOL.

Each functional block has references to the pages in the micro-program flowdiagram where the program code is found.



GNECENTRALEN A/S

Designed by 770428 PKR
Drawn by 770531 AOB
Dwg.-C. Check
Design Check
Replaces Dwg. No.
due to ECN
Replaced by Dwg. No.

Unit DSC 801
Dwg. No. R 11966

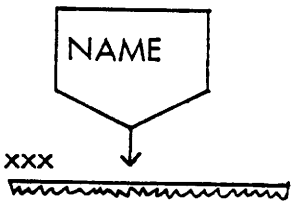
MICROPROGRAM BLOCKDIAGRAM

RC doc VP

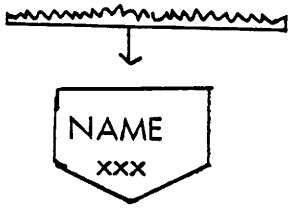
4. MICROPROGRAM FLOWDIAGRAM. 4.

The flowdiagram is a complete and exact description of the DSC 801 microprogram, and it is usefull for errortracing during running and trouble shooting, as well as it gives an understanding of the DSC 801 functional structure.

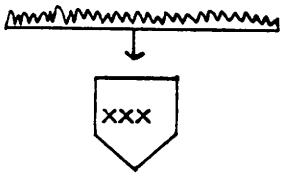
Labels and Connectors:



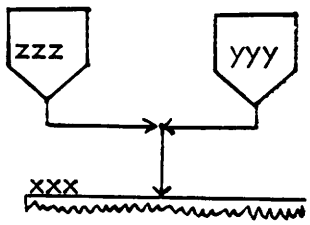
Entry to a program (NAME) with first instruction in word xxx (Octal).



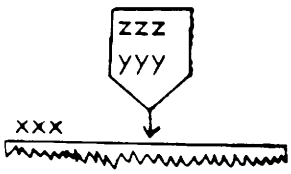
Off page connector indicating a jump to program (NAME) found on page Y



On page connector indicating a jump to word xxx found on same page.

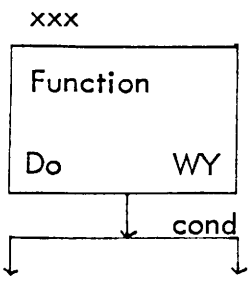


On page connector indicating entry from words zzz or yyy on the same page. Equivalent to the symbol



Statements:

Programword:



xxx: Is the octal address of the program word

Function: Is one or more assignments:

dest.A, dest.B: = AC op Source;

or an addressing:

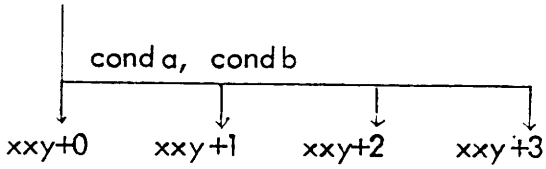
AC op Source;

used for tests resulting in a conditional branch

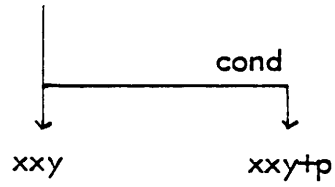
Do: Is a number of controles active during the instruction time.

WY: Y indicates a wait condition which must be satisfied before the instruction can be completed. If no wait condition is indicated, the instruction is completed in 200 nSec.

Branches:



This is a full fork. Y has one of the values 0 or 4, and next instruction is found in address: xxy+2 xxy+2 x cond a + cond b, where conda and condb are the values of the selected conditions.



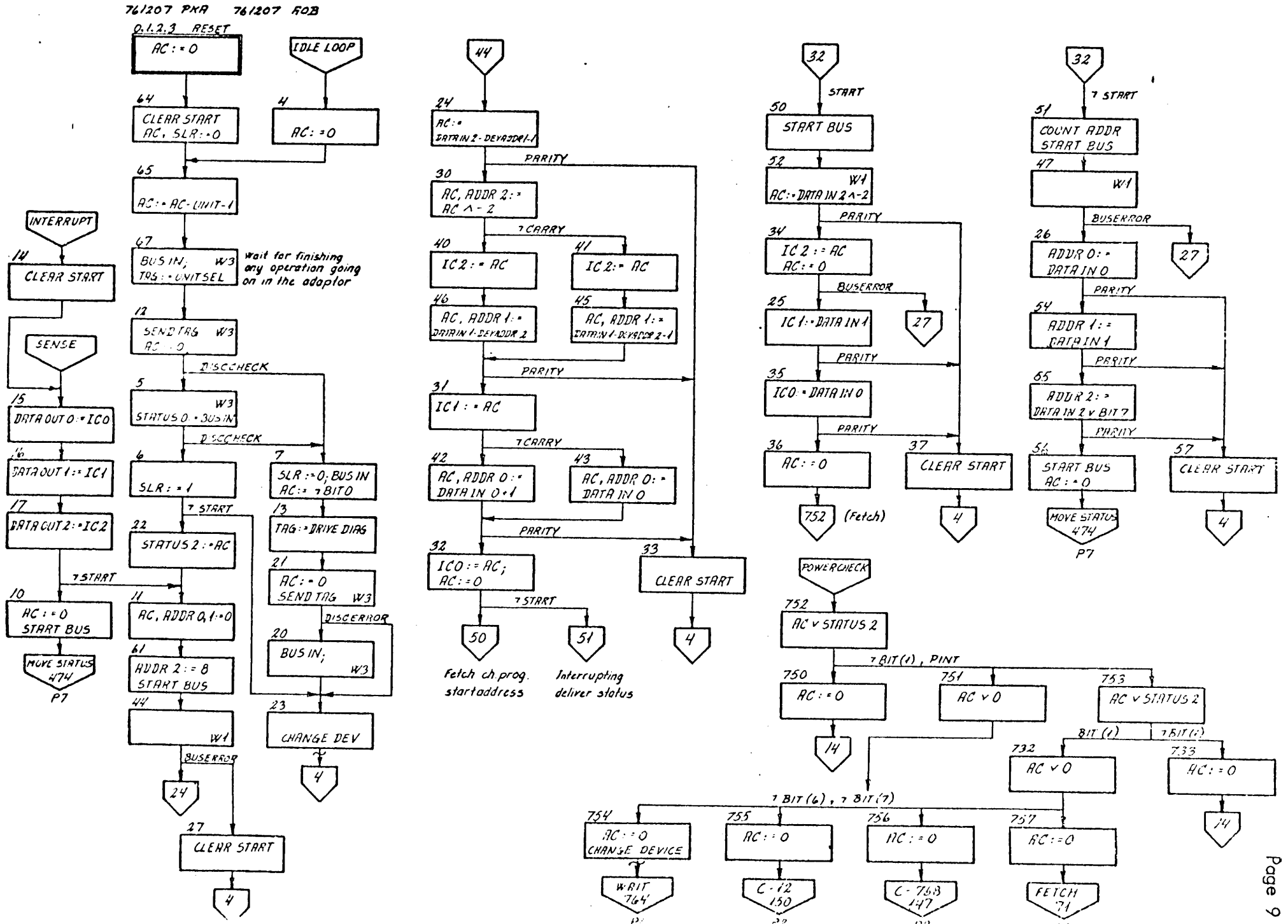
This is a partial (two) forks. The value of the condition directs the program flow towards right or left. Generally a true condition opens the branch on which it is written.

In full forks where the selected conditions are mutual exclusive the impossible branch is omitted (The corresponding program word is normally used elsewhere in the program).

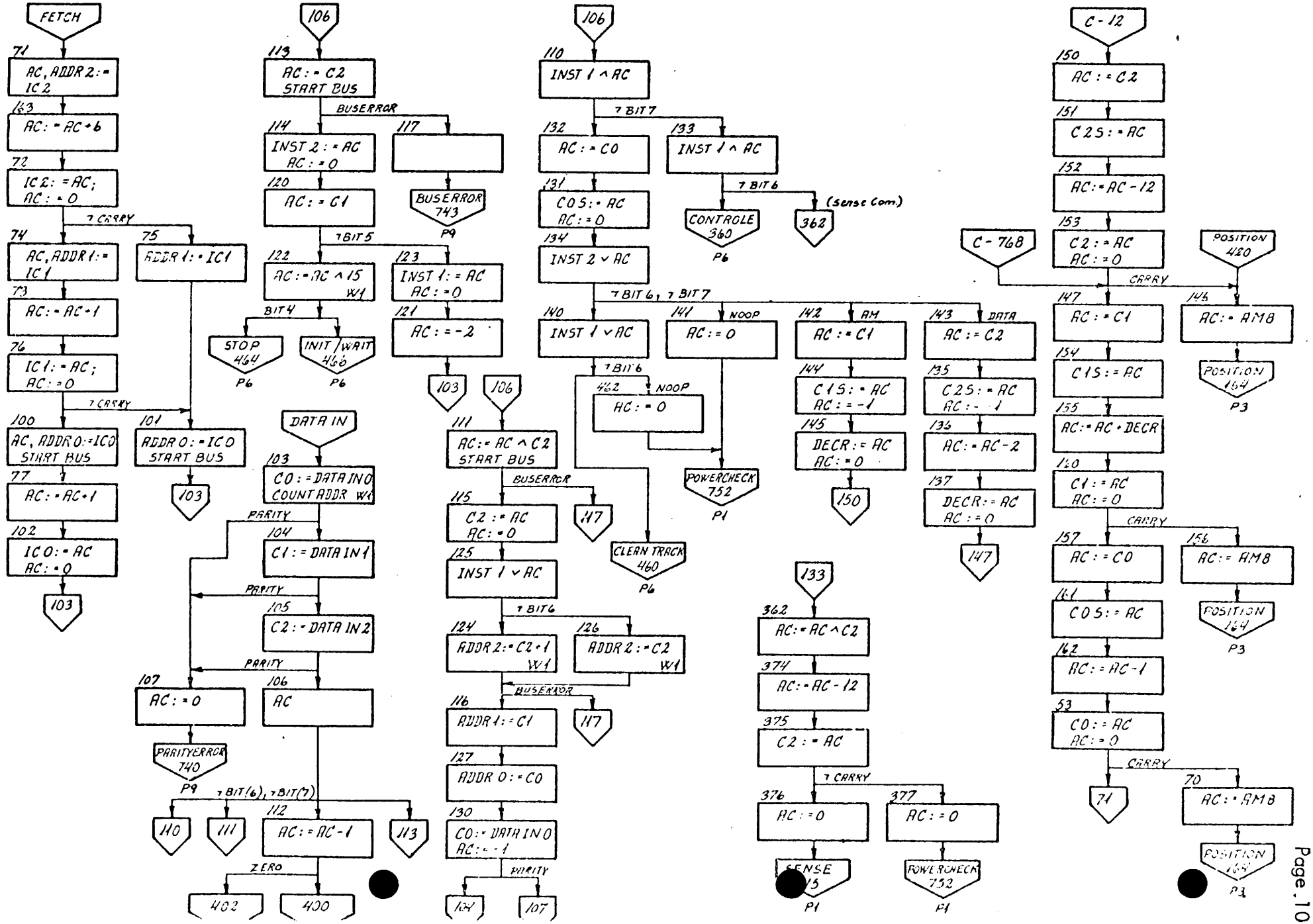
A fork may be expanded with an errorcondition. If the errorcondition is true, the branch pointing at the highest next address is selected disregarded the value of other branch selectors.

MICROPROGRAM RESET, INTERRUPT, START, POWERCHECK

761207 PXA 761207 AOB

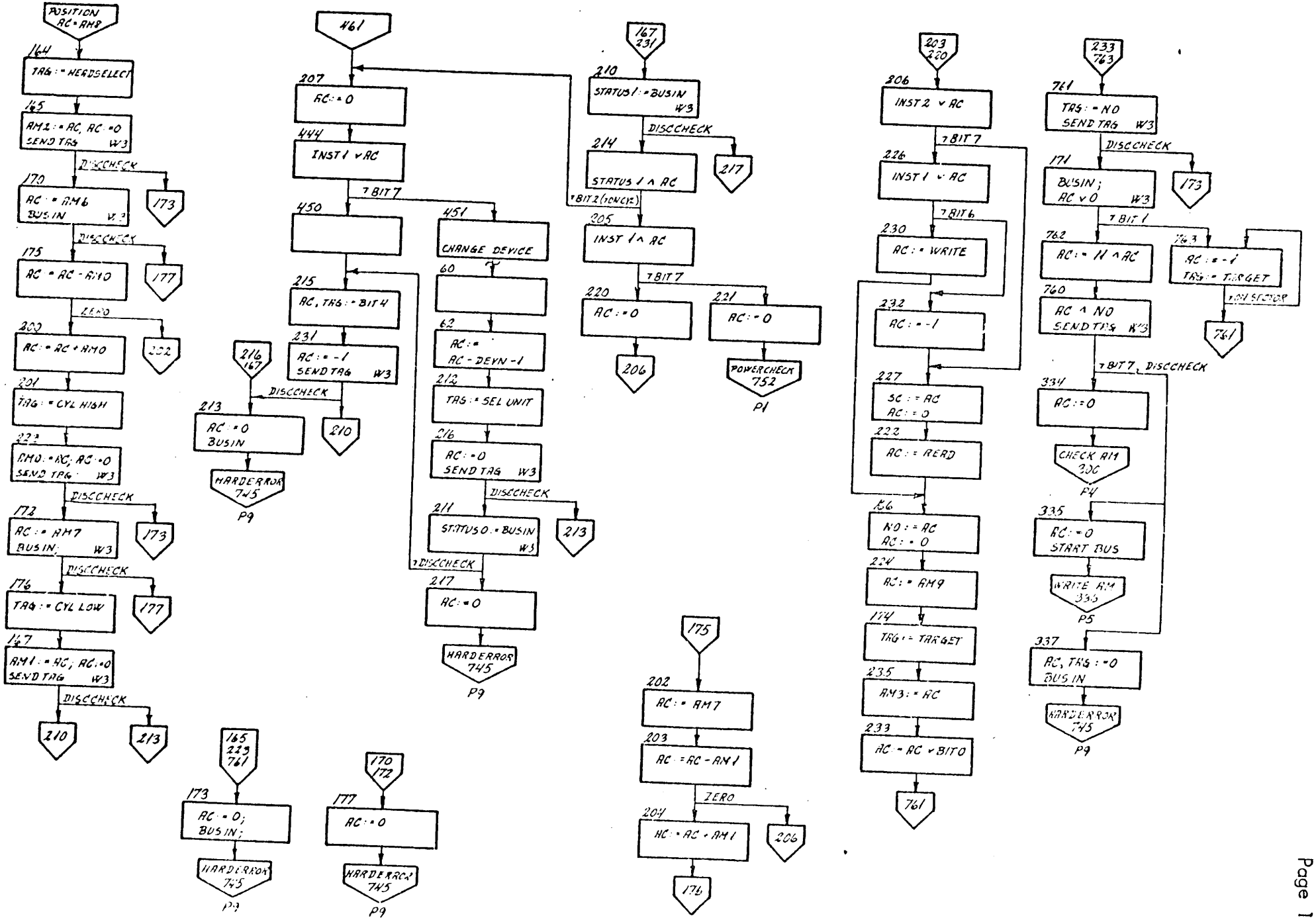


761207 PKR 761207 ROB

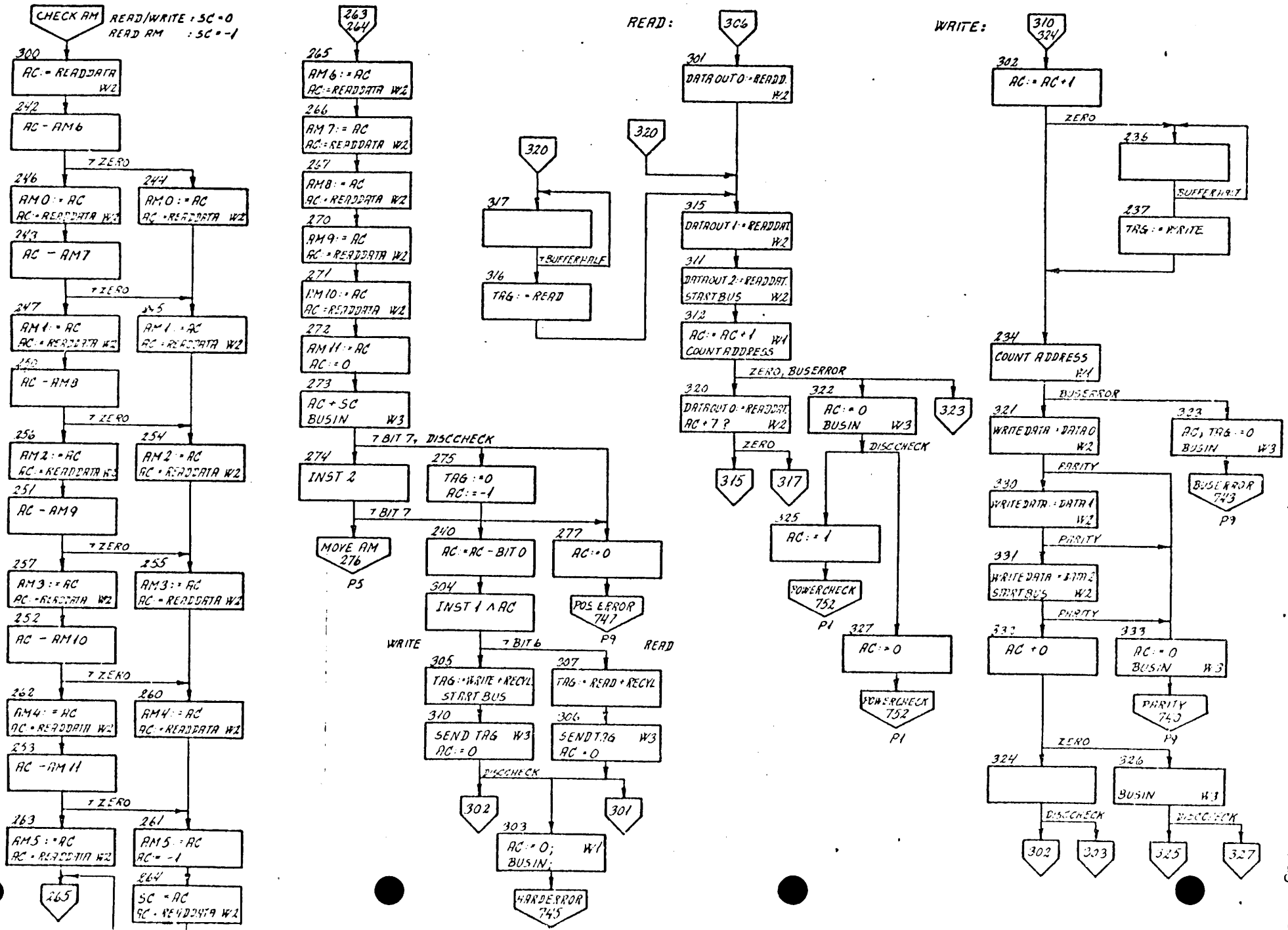


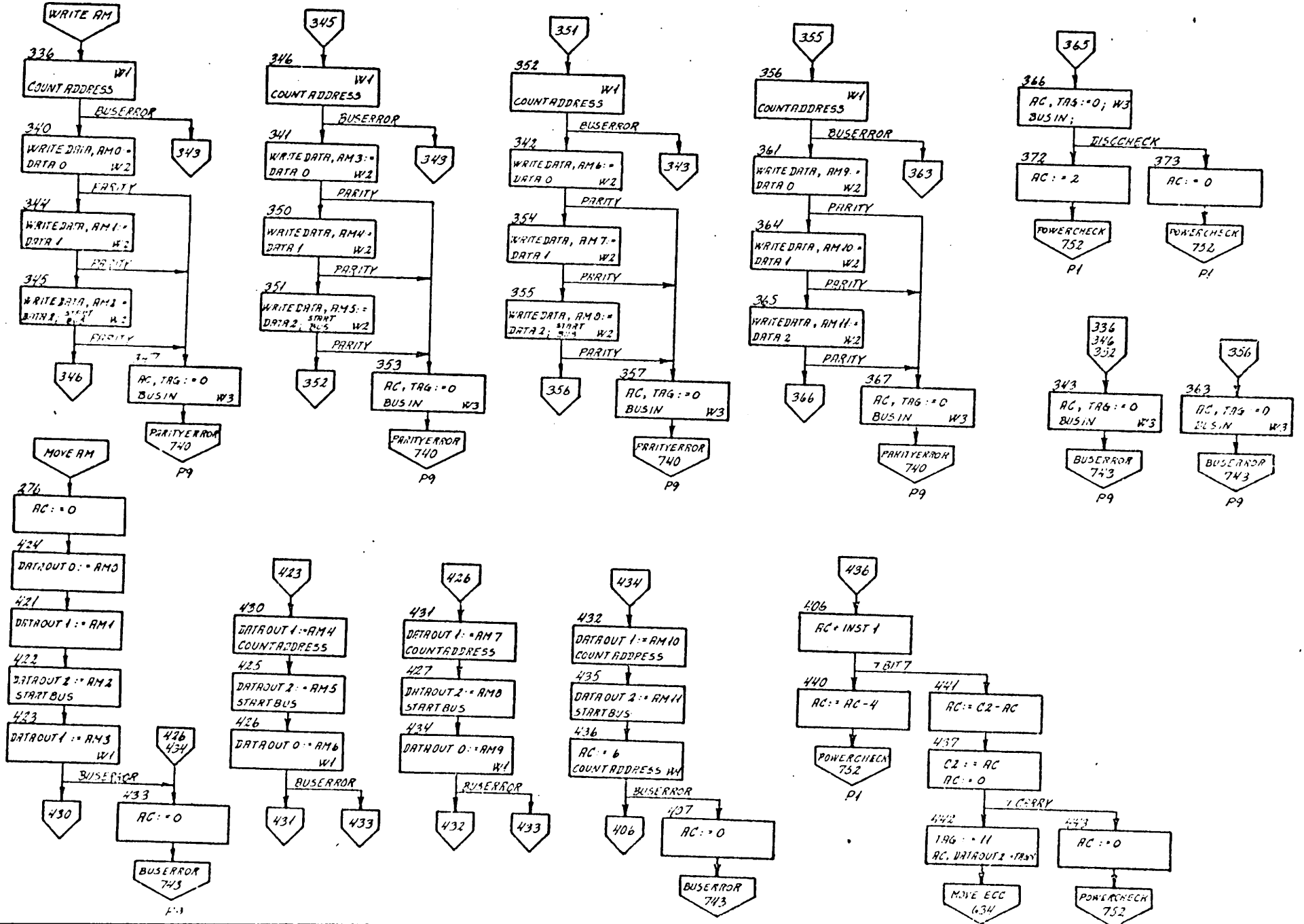
MICROPROGRAM FETCH, BRANCH

MICROPROGRAM
POSITION

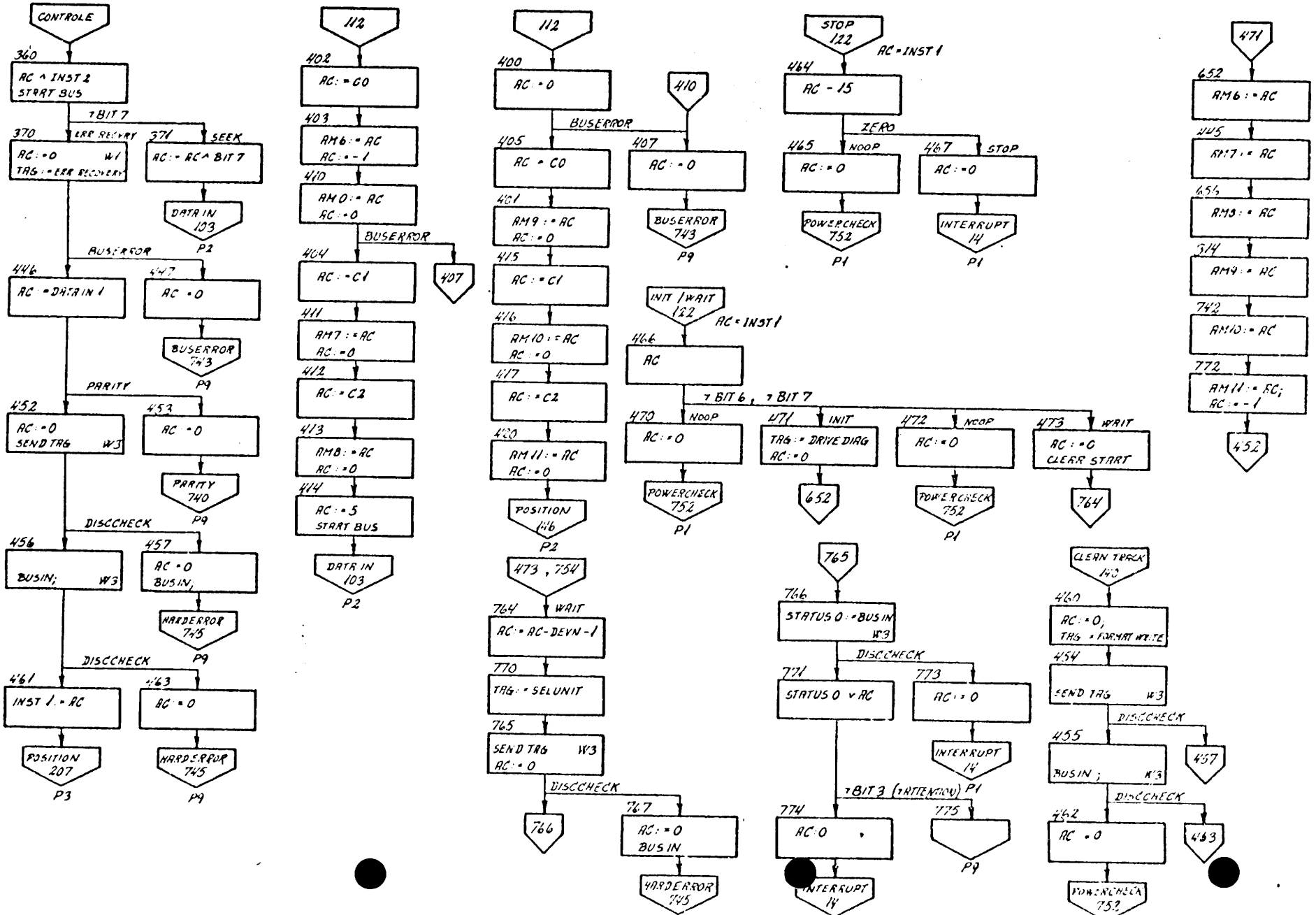


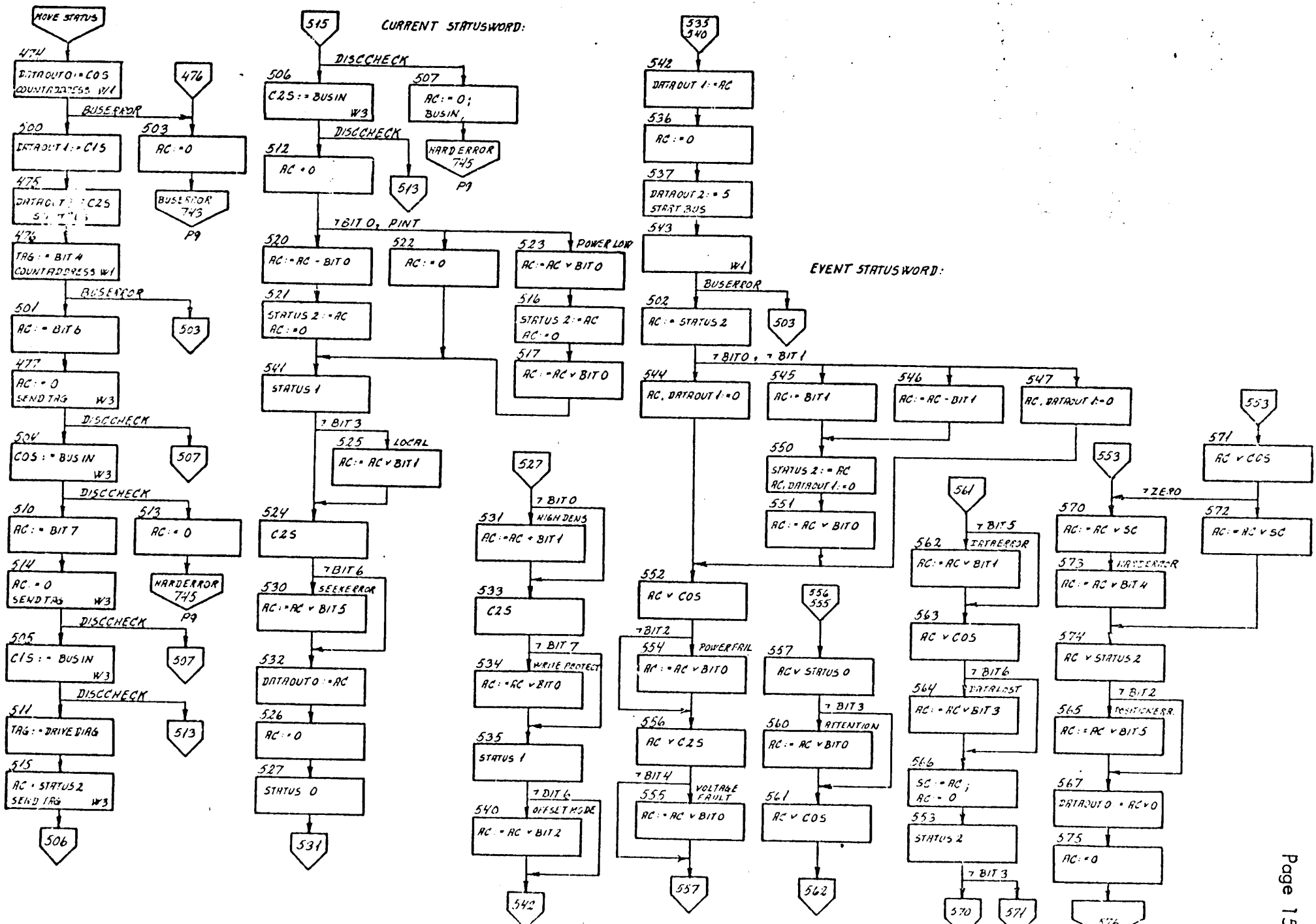
761207 PHA 761207 AOB




WRITE AM
336
W1
COUNT ADDRESS
BUSERROR
340
WRITE DATA, AM0 := DATA 0 W2
343
PARITY
344
WRITE DATA, AM1 := DATA 1 W2
PARITY
345
WRITE DATA, AM2 := DATA 2; START BUS W2
PARITY
346
347
AC, TAG := 0 BUS IN W3
PARITYERROR 740
P9
345
346
W1
COUNT ADDRESS
BUSERROR
341
WRITE DATA, AM3 := DATA 0 W2
343
PARITY
350
WRITE DATA, AM4 := DATA 1 W2
PARITY
351
WRITE DATA, AM5 := DATA 2; START BUS W2
PARITY
352
353
AC, TAG := 0 BUS IN W3
PARITYERROR 740
P9
351
352
W1
COUNT ADDRESS
BUSERROR
342
WRITE DATA, AM6 := DATA 0 W2
343
PARITY
354
WRITE DATA, AM7 := DATA 1 W2
PARITY
355
WRITE DATA, AM8 := DATA 2; START BUS W2
PARITY
356
357
AC, TAG := 0 BUS IN W3
PARITYERROR 740
P9
355
356
W1
COUNT ADDRESS
BUSERROR
361
WRITE DATA, AM9 := DATA 0 W2
363
PARITY
364
WRITE DATA, AM10 := DATA 1 W2
PARITY
365
WRITE DATA, AM11 := DATA 2 W2
PARITY
366
367
AC, TAG := 0 BUS IN W3
PARITYERROR 740
P9
365
366
AC, TAG := 0; W3 BUS IN;
DISCCHECK
372
AC := 2
373
AC := 0
POWERCHECK 752
P1
POWERCHECK 752
P1
336
346
352
343
AC, TAG := 0 BUS IN W3
BUSERROR 743
P9
356
363
AC, TAG := 0 BUS IN W3
BUSERROR 743
P9
MOVE AM
276
AC := 0
424
DATAOUT 0 := AM0
421
DATAOUT 1 := AM1
422
DATAOUT 2 := AM2 START BUS
423
DATAOUT 3 := AM3 W1
BUSERROR
426
434
430
433
AC := 0
BUSERROR 743
P9
423
430
DATAOUT 1 := AM4 COUNT ADDRESS
425
DATAOUT 2 := AM5 START BUS
426
DATAOUT 0 := AM6 W1
BUSERROR
431
433
426
431
DATAOUT 1 := AM7 COUNT ADDRESS
427
DATAOUT 2 := AM8 START BUS
434
DATAOUT 0 := AM9 W1
BUSERROR
432
433
434
432
DATAOUT 1 := AM10 COUNT ADDRESS
435
DATAOUT 2 := AM11 START BUS
436
AC := 6 COUNT ADDRESS W1
BUSERROR
406
437
AC := 0
BUSERROR 743
436
406
AC := INST 1
7 BIT 7
440
AC := AC - 4
441
AC := C2 - AC
POWERCHECK 752
P1
437
C2 := AC
AC := 0
1 CARRY
442
LOG := 11 AC, DATAOUT 2 := PARITY
443
AC := 0
MOVE ECC 634
POWERCHECK 752

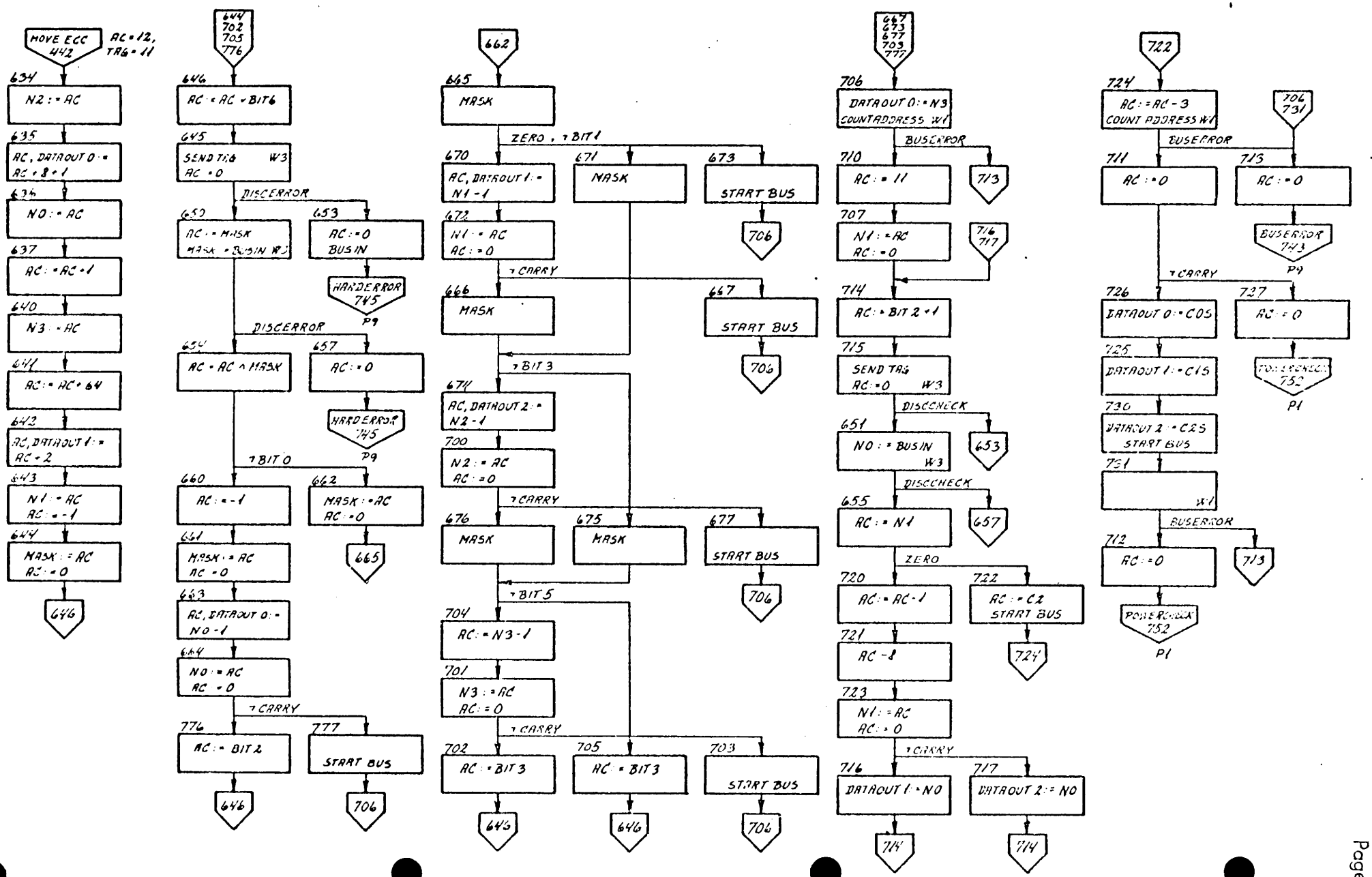
SEEK:

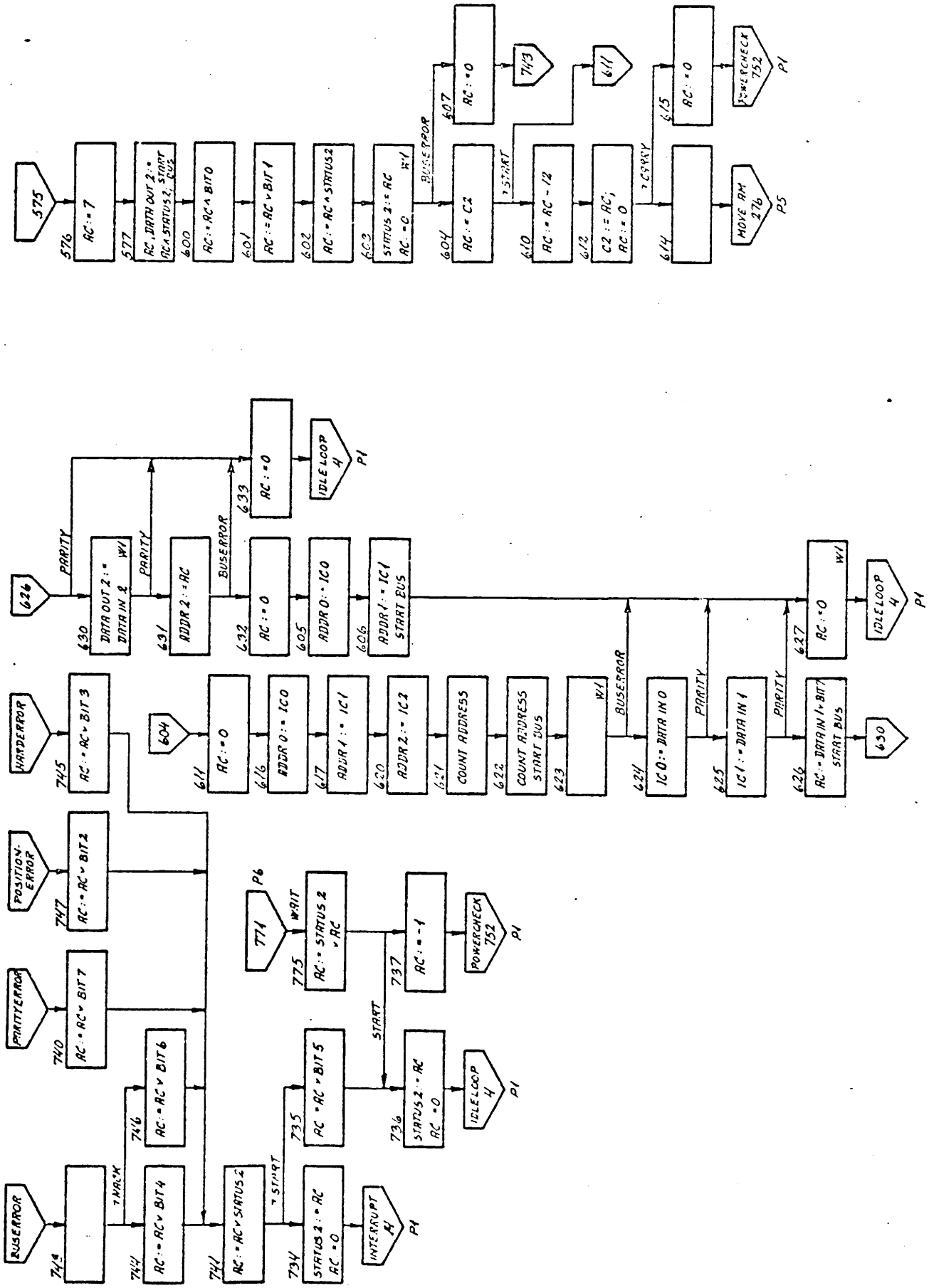

CONTROLE
360
AC ^ INST 2
START BUS
↑BIT 7
ERR RECVRY
370
AC: = 0 W1
TAG: = ERR RECOVERY
371
SEEK
AC: = AC ^ BIT 7
DATA IN 103
P2
BUSERROR
446
AC: = DATA IN 1
447
AC: = 0
BUSERROR 743
P9
PARITY
452
AC: = 0
SEND TAG W3
453
AC: = 0
PARITY 740
P9
DISCCHECK
456
BUSIN; W3
457
AC: = 0
BUSIN,
HARDERROR 745
P9
DISCCHECK
461
INST 1 = AC
463
AC: = 0
POSITION 207
P3
HARDERROR 745
P9
112
402
AC: = C0
403
AM6: = AC
AC: = -1
410
AM0: = AC
AC: = 0
BUSERROR
404
AC: = C1
407
411
AM7: = AC
AC: = 0
412
AC: = C2
413
AM8: = AC
AC: = 0
414
AC: = 5
START BUS
DATA IN 103
P2
112
400
AC: = 0
BUSERROR
405
AC = C0
401
AM9: = AC
AC: = 0
415
AC: = C1
416
AM10: = AC
AC: = 0
417
AC: = C2
420
AM11: = AC
AC: = 0
POSITION 116
P2
473, 754
764
WAIT
AC: = AC-DEVN-1
770
TAG: = SELUNIT
765
SEND TAG W3
AC: = 0
DISCCHECK
766
767
AC: = 0
BUS IN
HARDERROR 745
410
407
AC: = 0
BUSERROR 743
P9
INIT / WAIT 122
AC = INST 1
466
AC
↑BIT 6, ↑BIT 7
470
NOOP
AC: = 0
POWERCHECK 752
P1
471
INIT
TAG: = DRIVE DIAG
AC: = 0
652
472
NOOP
AC: = 0
POWERCHECK 752
P1
473
WAIT
AC: = 0
CLERR START
764
765
766
STATUS 0: = BUSIN W3
DISCCHECK
771
STATUS 0 ∨ AC
773
AC: = 0
INTERRUPT 14
↑BIT 3 (↑ATTENTION) P1
774
AC: 0
775
INTERRUPT 14
P9
STOP 122
AC = INST 1
464
AC - 15
ZERO
465
NOOP
AC: = 0
POWERCHECK 752
P1
467
STOP
AC: = 0
INTERRUPT 14
P1
471
652
AM6: = AC
445
AM7: = AC
653
AM8: = AC
314
AM9: = AC
742
AM10: = AC
772
AM11: = AC;
AC: = -1
452
CLEAR TRACK 140
460
AC: = 0;
TAG = FORMAT WRITE
454
SEND TAG W3
DISCCHECK
455
BUSIN; W3
457
DISCCHECK
462
AC = 0
463
POWERCHECK 752


MOVE STATUS
474
DATAOUT0:=COS COUNTADDRESS W1
BUSERROR
476
200
DATAOUT1:=CIS
503
AC:=0
BUSERROR 743
P9
475
DATAOUT2:=C2S START BUS
476
TAG:=BIT 4 COUNTADDRESS W1
BUSERROR
501
AC:=BIT 6
503
477
AC:=0 SEND TAG W3
DISCCHECK
504
COS:=BUS IN W3
507
DISCCHECK
510
AC:=BIT 7
513
AC:=0
514
AC:=0 SEND TAG W3
HARDERROR 745
P9
DISCCHECK
505
CIS:=BUS IN W3
507
DISCCHECK
511
TAG:=DRIVE DIAG
513
515
AC:=STATUS 2 SEND TAG W3
506
515
CURRENT STATUSWORD:
DISCCHECK
506
C2S:=BUS IN W3
507
AC:=0; BUSIN,
DISCCHECK
512
AC:=0
513
HARDERROR 745
P9
7 BIT 0, PINT
520
AC:=AC - BIT 0
522
AC:=0
523
POWER LOW
AC:=AC v BIT 0
521
STATUS 2:=AC AC:=0
516
STATUS 2:=AC AC:=0
517
AC:=AC v BIT 0
541
STATUS 1
7 BIT 3
525
LOCAL
AC:=AC v BIT 1
524
C2S
7 BIT 6
530
SEEKERROR
AC:=AC v BIT 5
532
DATAOUT0:=AC
526
AC:=0
527
STATUS 0
531
527
7 BIT 0
531
HIGH DENS
AC:=AC + BIT 1
533
C2S
7 BIT 7
534
WRITE PROTECT
AC:=AC v BIT 0
535
STATUS 1
7 BIT 6
540
OFFSET MODE
AC:=AC v BIT 2
542
535 540
542
DATAOUT 1:=AC
536
AC:=0
537
DATAOUT 2:=5 START BUS
543
W1
BUSERROR
502
AC:=STATUS 2
503
EVENT STATUSWORD:
7 BIT 0, 7 BIT 1
544
AC, DATAOUT 1:=0
545
AC:=BIT 1
546
AC:=AC - BIT 1
547
AC, DATAOUT 1:=0
550
STATUS 2:=AC AC, DATAOUT 1:=0
551
AC:=AC v BIT 0
552
AC v COS
7 BIT 2
554
POWERFAIL
AC:=AC v BIT 0
556
AC v C2S
7 BIT 4
555
VOLTAGE FAULT
AC:=AC v BIT 0
557
556 555
557
AC v STATUS 0
7 BIT 3
560
ATTENTION
AC:=AC v BIT 0
561
AC v COS
562
561
7 BIT 5
562
INTAERROR
AC:=AC v BIT 1
563
AC v COS
7 BIT 6
564
DATALOST
AC:=AC v BIT 3
565
SC:=AC; AC:=0
553
STATUS 2
7 BIT 3
570
571
553
570
AC:=AC v SC
573
MAINERROR
AC:=AC v BIT 4
574
AC v STATUS 2
7 BIT 2
565
PARITYERR.
AC:=AC v BIT 5
567
DATAOUT 0:=AC v 0
575
AC:=0
576
553
7 ZERO
571
AC v COS
572
AC:=AC v SC

DSC 801
R 11691

MICROPROGRAM
MOVE ECC, DIAGNOSTIC




761207 PKR 761207 AOB
BUSERROR
PARITYERROR
POSITION-ERROR
HARDERROR
575
626
604
771
P6
P1
INTERRUPT
IDLE LOOP
POWERCHECK
MOVE RM
START
WAIT
PARITY
BUSERROR

RGM 171 / RGM 170	0	2	4	6	10	12	14	16
CONSTANTS 0	0 00000000	BIT 1 64 01000000	BIT 2 32 00100000	BIT 3 16 00010000	BIT 4 8 STATUS 00001000	BIT 5 4 TARGET 00000100	BIT 6 2 00000010	BIT 7 1 00000001
CONSTANTS 1	3 00000011	5 00000101	6 00000110	7 HEADSEL 00000111	12 00001100	-2 11111110	15 00001111	-8 11111000
CONSTANTS 2	19 UNITSEL 00010011	21 LOW CYL 00010101	22 HIGH CYL 00010110	11 ECC 00001011	28 DRIVE DIAG 00011100	13 ERR. RECOVERY 00001101	14 FORMAT WRITE 00001110	
CONSTANTS 3	121 READ 01111001	249 READ, RECYCLE 11111001	58 WRITE 00111010	186 WRITE, RECYCLE 10111010			BIT 0 10000000	-1 11111111
VARIABLES 4	AM 0 CYL HIGH	AM 1 CYL LOW	AM 2 HEAD	AM 3 SECTOR	AM 4 SOFT 0	AM 5 SOFT 1	AM 6 CYL HIGH NEXT	AM 7 CYL LOW NEXT
VARIABLES 5	AM 8 HEAD NEXT	AM 9 SECTOR NEXT	AM 10 SOFT 0 NEXT	AM 11 SOFT 1 NEXT	INST 1	INST 2	N 0	N 1
VARIABLES 6	C 0	C 1	C 2	C 0 S	C 1 S	C 2 S	N 2	N 3
VARIABLES 7	STATUS 0	STATUS 1	STATUS 2	IC 0	IC 1	IC 2	SC	MASK, DECR, SAVE

WRITE STORE +10

COND 0 SEL =	MA (7): =
0	1
1	↗ DAINT (↗ ON SECTOR)
2	↗ NACK
3	ZERO
4	↗ SUM 0
5	↗ SUM 2
6	↗ SUM 4
7	↗ SUM 6
8	0

COND 1 SEL =	MA (8): =
0	1
1	↗ BUFFER HALF
2	↗ START
3	↗ CARRYOUT DELAYED
4	↗ SUM 1
5	↗ SUM 3
6	↗ SUM 5
7	↗ SUM 7
8	0

ERR. COND SEL =	IF ERROR THEN MA(7,8):=(1,1) ERROR =
0	FALSE
1	BUSCHECK
2	PARITY ERROR
3	DISCCHECK
4	PINT
5	HANDSHAKE
6	
7	

CPSEL	CLOCKCONDITION =
0	TRUE
1	BUSREADY
2	BUFFERACCESS
3	DISCREADY

761207 PKA · 761207 ROB

DSC 801

R 21058

CONDITIONS

RETURN LETTER

Title: DSC 801 MICROPROGRAM RCSL No.: 30 M 81

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

_____

_____

_____

_____

Do you find errors in this manual? If so, specify by page.

_____

_____

_____

_____

How can this manual be improved?

_____

_____

_____

_____

Other comments?

_____

_____

_____

_____

_____

Name: _____ Title: _____

Company: _____

Address: _____

Date: _____

Thank you

42-i 1288

..... Fold here .....

..... Do not tear - Fold here and staple .....

Affix
postage
here

REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark