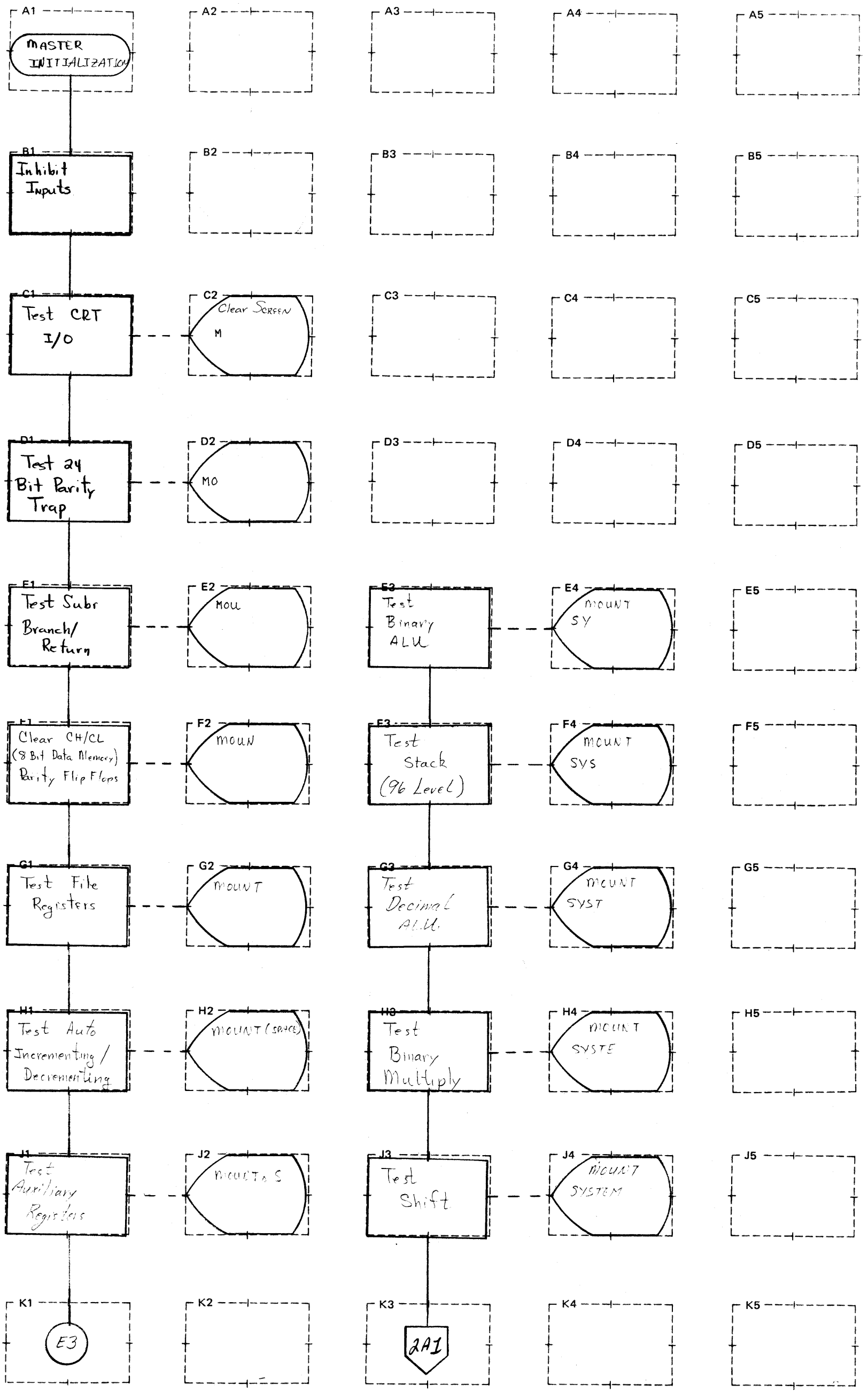
SYSTEM 2200VP PROGRAM BOOTSTRAP

VERSION _____ PREPARED BY J. Sevigny DATE 6-76

CHART MASTER INITIALIZATION

SHEET 1 of 2



Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

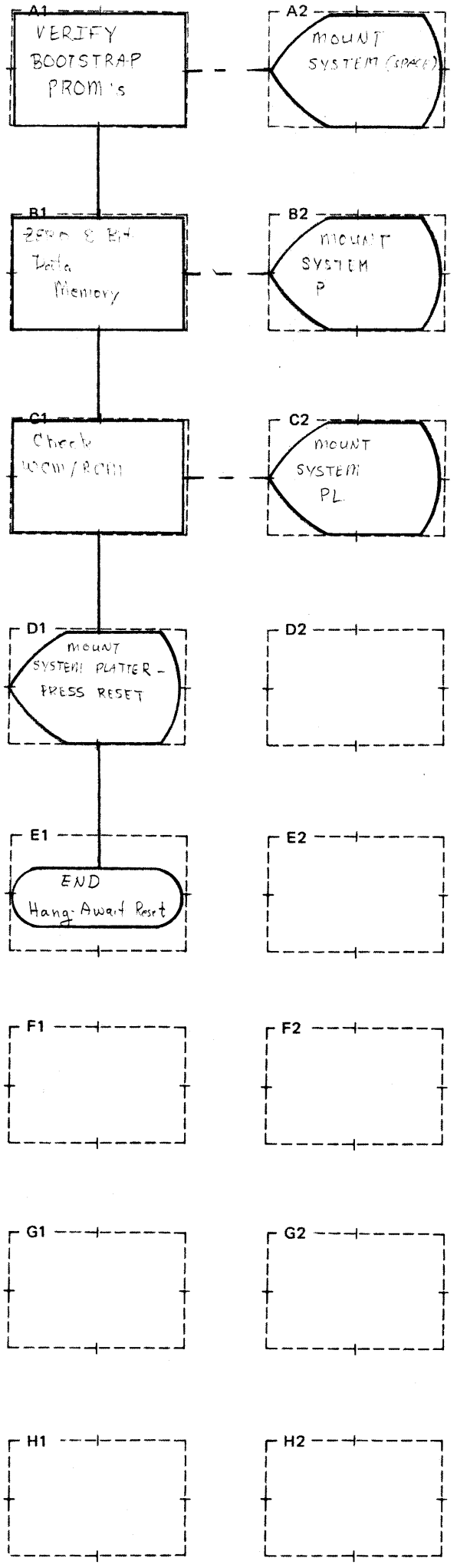
Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM 2200VP PROGRAM BOOTSTRAP

VERSION _____ PREPARED BY J. Sevigny DATE 6/76

CHART MASTER INITIALIZATION

SHEET 2 OF 2



Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

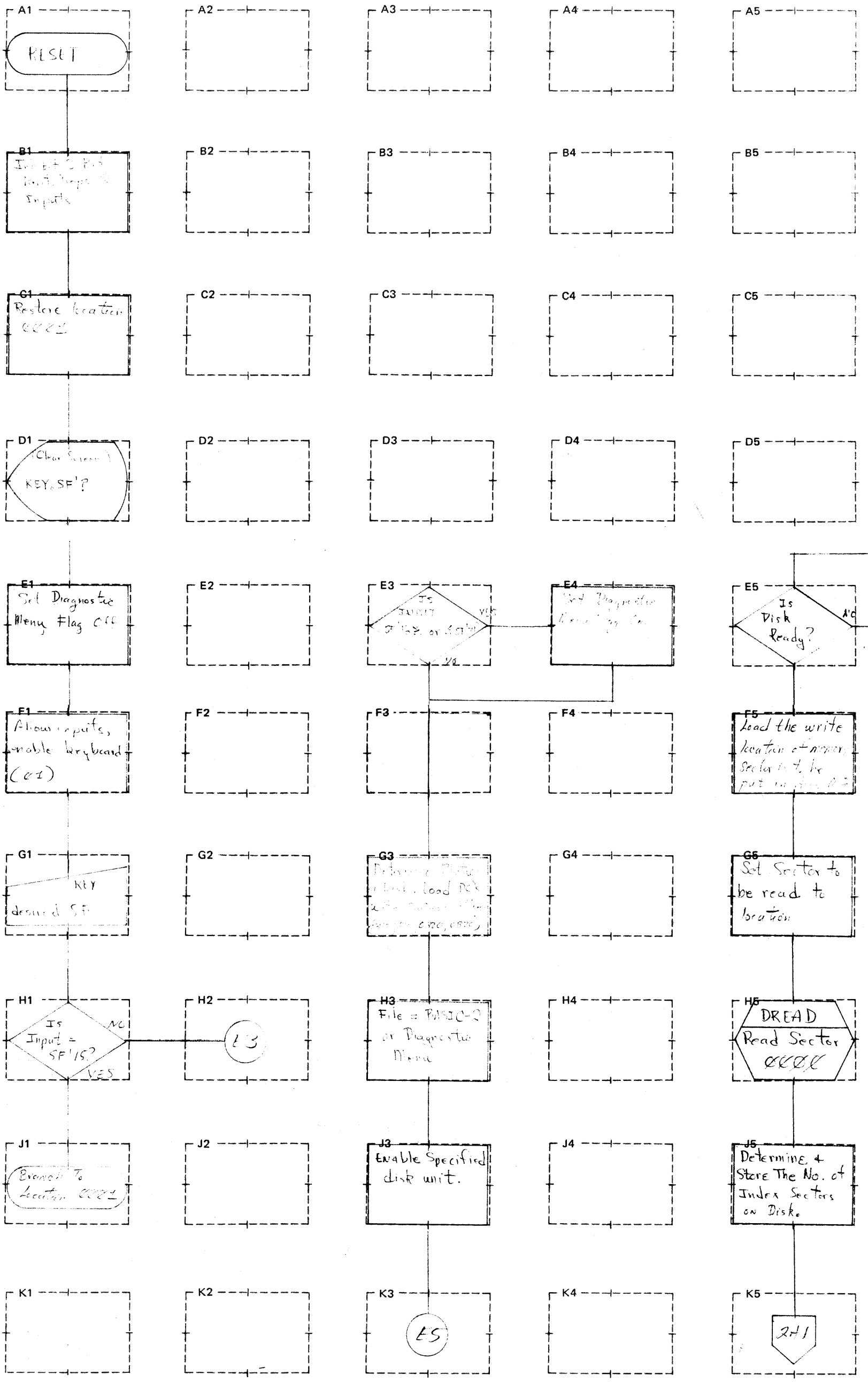
Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM Wang 2466 PROGRAM Diagnostic

VERSION _____ PREPARED BY Jerry Swiggy DATE 6-76

CHART RESET

SHEET 1 OF 4



Fold under at dotted line.

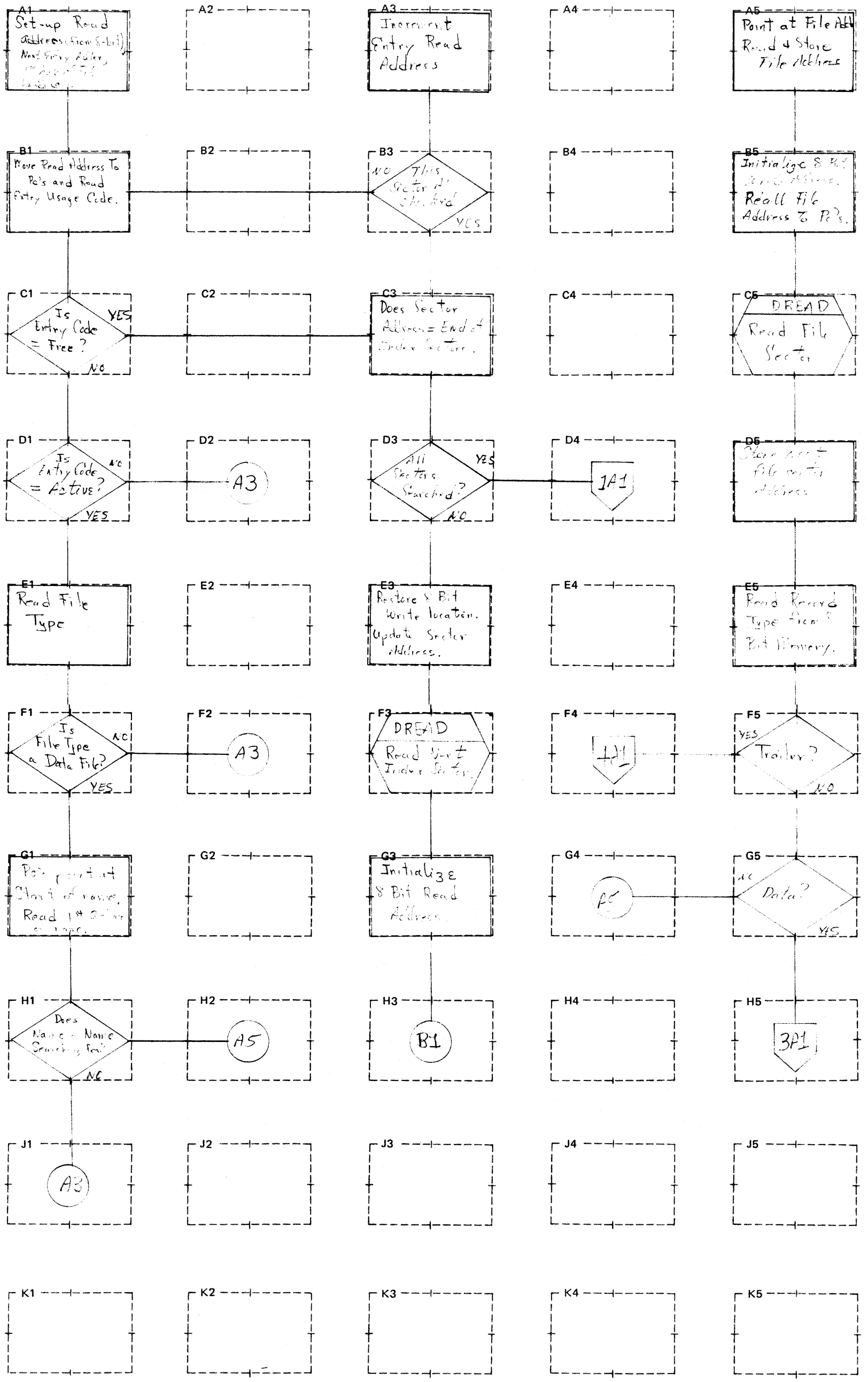
Fold under at dotted line.

WANG LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM 2600	PROGRAM BOOTSTRAP	CHART RESET
VERSION	PREPARED BY Jerry Scvigny	DATE 5-76	SHEET 2 OF 4



Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

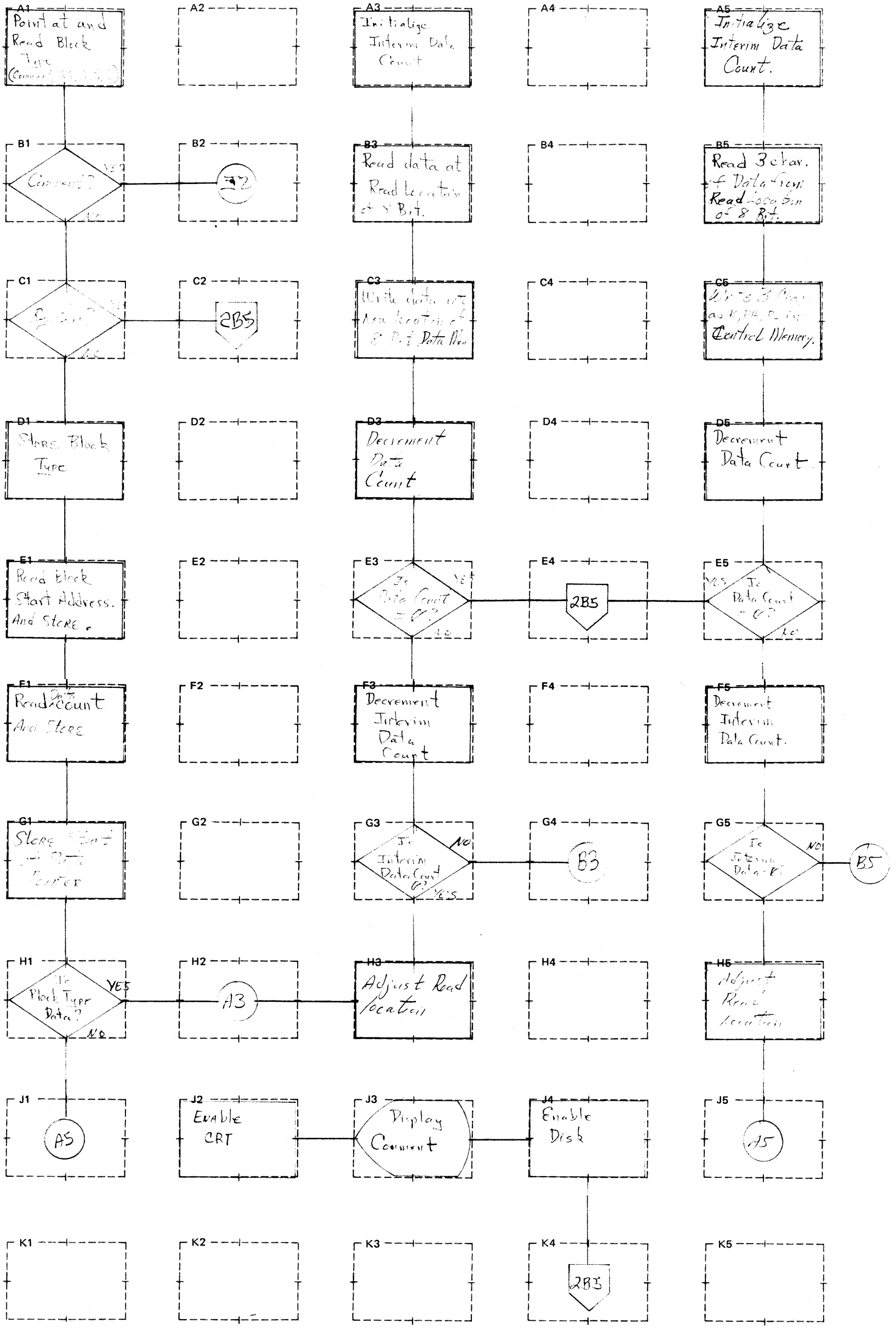
Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM _____ PROGRAM BOOTSTRAP

VERSION _____ PREPARED BY Jerry Savigny DATE 6-76

CHART PLST 1

SHEET 3 OF 4



Fold under at dotted line.

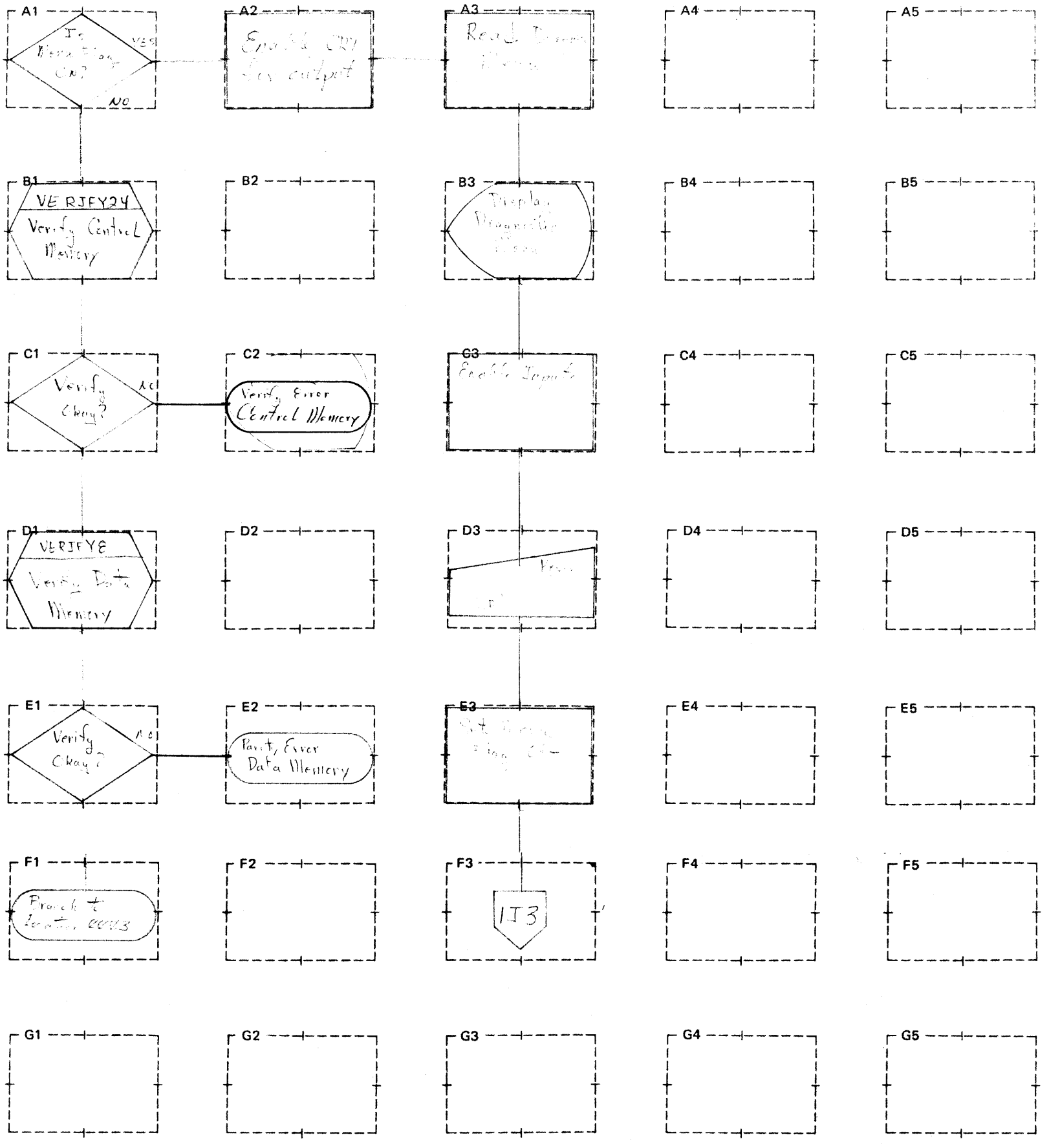
Fold under at dotted line.

WANG LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM 2200	PROGRAM BOOTSTRAP	CHART NESL 1	
VERSION	PREPARED BY Jerry Sedgley	DATE 6-76	SHEET 4 OF 4



Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

```
002                TITLE    CONTROL MEMORY DIAG.
003  *               ROWP24:
004  *                    This routine will perform a test of 24
005  *               bit CONTROL MEMORY. The algorithm used is a
006  *               modified ROWPAT.
007  *
008  *                    To explain the algorithm it is necessary
009  *               to define certain terms -
010  *                    TEST CELL - that location being tested.
011  *                    CONFLICT CELL - that location being
012  *                                    tested for a CONFLICT
013  *                                    with the TEST CELL.
014  *                    ROW - a row of addresses within the
015  *                          memory chip of which the TEST
016  *                          CELL is one of the addresses.
017  *                    COLUMN - a column of addresses within the
018  *                          memory chip of which the TEST
019  *                          CELL is one of the addresses.
020  *                    BOARD ROW - a row of memory chips located
021  *                          on the memory board.
022  *                    TEST PATTERN - that pattern expected to
023  *                          be found in the TEST CELL.
024  *                    FLOOD PATTERN - that pattern expected to
025  *                           be found in all other cells.
026  *                    MEMORY TEST AREA
027  *                         - from 0000 to END OF MEMORY - 0800
028  *                           if program is in low memory.
029  *                         - from 0800 to END OF MEMORY if
030  *                           program is in high memory.
031  *
032  *                    The ALGORITHM is -
033  *                         1. Write zeroes to all locations in
034  *                           the current MEMORY TEST AREA.
035  *                         2. Read current TEST CELL and check
036  *                           for TEST PATTERN.
037  *                         3. Write the TEST PATTERN at the
038  *                           TEST CELL.
039  *                         4. Read TEST CELL and check for TEST
040  *                           PATTERN.
041  *                         5. Read the CONFLICT CELL and check
042  *                           for FLOOD PATTERN.
043  *                         6. Repeat steps 4 and 5 making the
044  *                           CONFLICT CELL the next location
045  *                           within the COLUMN and then within
046  *                           the ROW.
047  *                         7. Repeat steps 2 through 6 making
048  *                           the TEST CELL the next location
049  *                           within the CHIP ROW until the TEST
050  *                           CELL has occpuied each memory
051  *                           location within the CHIP ROW.
052  *                         8. Repeat steps 2 through 7 for each
053  *                           CHIP ROW within the MEMORY TEST
054  *                           AREA.
055  *                         9. Write all one's into the MEMORY
```

```
056  *                    TEST AREA and repeat steps 2
057  *                  through 8.
058  *               10. Move the program from low memory
059  *                  to high memory and repeat steps 1
060  *                  through 9.
061  *
062  *
063  *          MODIFIED REGISTERS:
064  *               F0 - F5, AUX00 - AUX17, PL, PH
065  *
066  *          REGISTER USAGE:
067  *               AUX 00 - start address of CHIP ROW
068  *               AUX 01 - end address of CHIP ROW minus 1.
069  *               AUX 02 - start address of COLUMN
070  *               AUX 03 - end address of COLUMN
071  *               AUX 04 - start address of ROW
072  *               AUX 05 - end address of row minus 1
073  *               AUX 06 - TEST CELL address
074  *               AUX 07 - CONFLICT CELL address
075  *               AUX 08 - END OF MEMORY minus 1
076  *               AUX 09 - constant 0020
077  *               AUX 0A - constant 0FC0
078  *               AUX 0B - constant F03F
079  *               AUX 0C - constant 0FFE
080  *               AUX 0D - temporary storage
081  *               AUX 0E - temporary storage
082  *               AUX 0F - temporary storage
083  *               AUX 10 - loop counter
084  *               AUX 11 - temporary storage
085  *               AUX 12 - old MEMORY TEST AREA pointer
086  *               AUX 13 - new MEMORY TEST AREA pointer
087  *               AUX 14 - end of program pointer
088  *               AUX 15 - end of MEMORY TEST AREA pointer
089  *               AUX 16 - start of MEMORY TEST AREA point.
090  *               AUX 17 - start low memory address pointer
091  *               F0     - TEST PATTERN
092  *               F1     - constant 00
093  *               F3,F2  - constant 0020
094  *               F5,F4  - working register pair
095  *
096  *          SCREEN DISPLAY AND MESSAGES
097  *
098  *               line 0 - ROWPAT TEST - 24 BIT RAM
099  *               line 1 - # LLLL
100  *
101  *               where: line 0 = diagnostic title
102  *                        LLLL = # of completed loops
103  *                               through diagnostic
104  *
105  *          ERROR MESSAGES
106  *
107  *            1. FAILURE AT PC'S FFFF  (EEEEEE/RRRRRR)
108  *                                     XOR = XXXXXX
109  *            2. CONFLICT BETWEEN TTTT AND CCCC
```

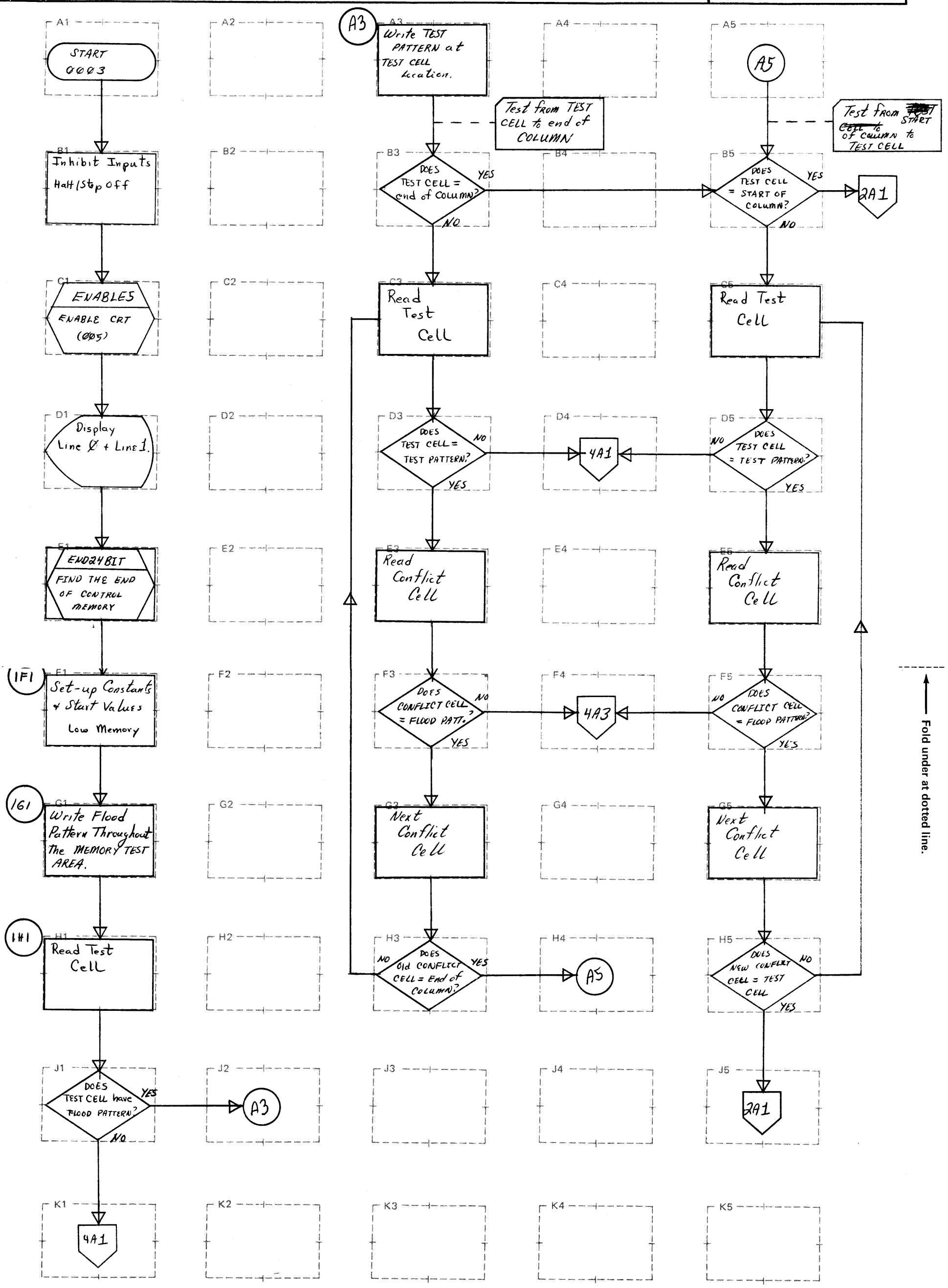
```
110  *
111  *                where: FFFF = failing memory address
112  *                     EEEEEE = expected pattern
113  *                              form (K,PH,PL)
114  *                     RRRRRR = actual pattern
115  *                              form (K,PH,PL)
116  *                     XXXXxX = XOR of expected/actual
117  *                       TTTT = TEST CELL address
118  *                       CCCC = CONFLICT CELL address
119  *
120  *          DISCUSSION OF ERROR MESSAGES
121  *
122  *                 ERROR 1 occurs when either registers K,
123  *          PH or PL fail to contain the expected pattern.
124  *                 ERROR 2 occurs when an ERROR 1 occurs
125  *          on a CONFLICT CELL read and either K,PH or PL
126  *          contain the TEST PATTERN.
127  *
128  *          *****  RESET  *****
129  *               When RESET is keyed a branch to the
130  *          BOOTSTRAP 'MOUNT SYSTEM DISK' routine is
131  *          executed.
132  *
133  *          *****  HALT/STEP  *****
134  *               The program may be interrupted by keying
135  *          HALT/STEP. However, the program will only halt
136  *          after an error message has debb displayed. To
137  *          resume with this program, key HALT/STEP again.
138  *
139  *          *****  I/O SUBROUTINES  *****
140  *               This program makes use of the BOOTSTRAP
141  *          subroutines whenever possible.
142  *
143          SYMBL     JSM0002$
144  *
145          B         PE24
146          B         MOUNT
147          B         PE8
148          B         START
149          B         *
150          B         *
151          B         *
152          B         *
153          B         *
154          B         *
```

SYSTEM _____ PROGRAM Control Memory Diagnostic

VERSION _____ PREPARED BY J. Sevigny DATE 12/75

CHART _____

SHEET 1 OF _____



Fold under at dotted line.

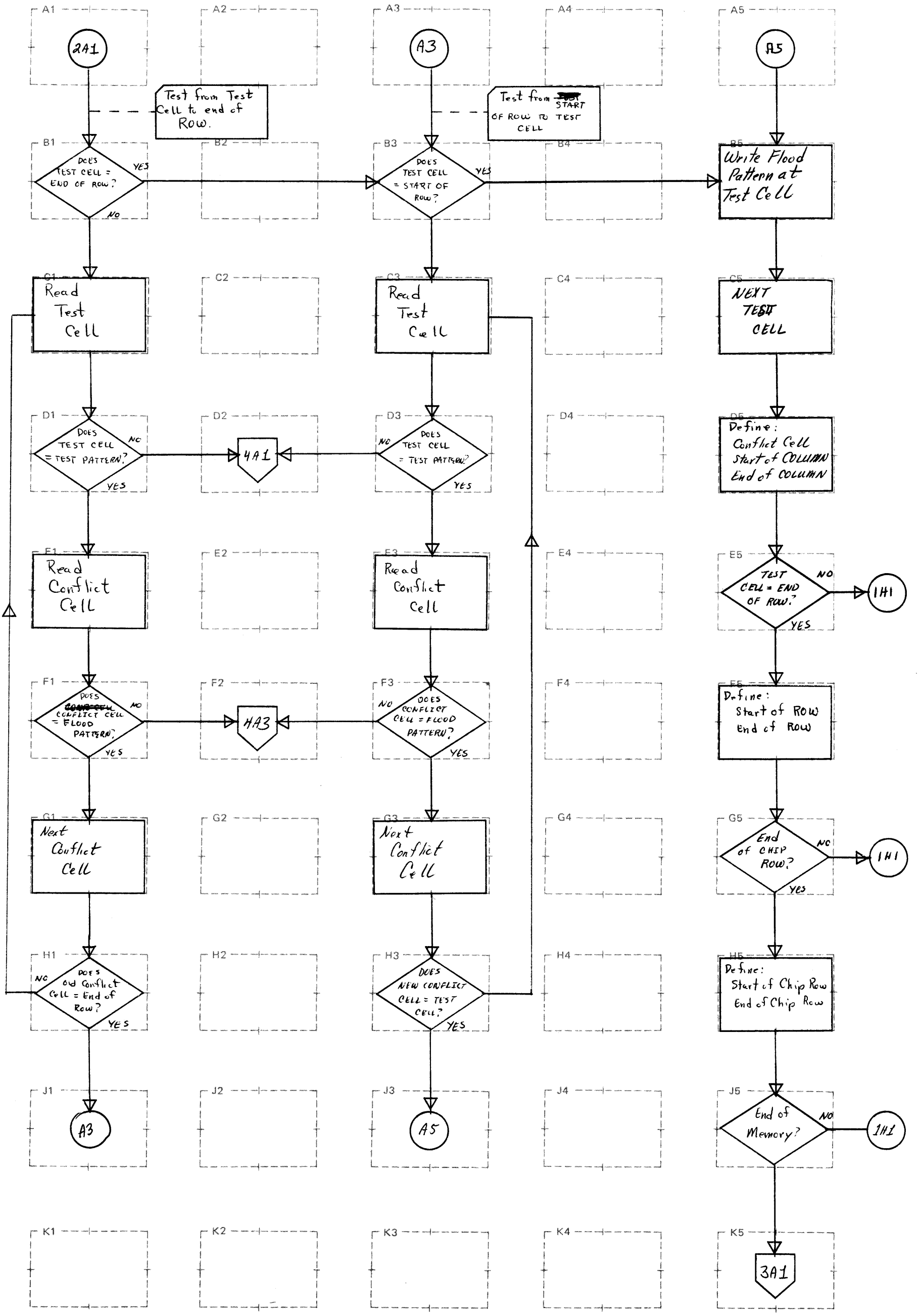
Fold under at dotted line.

WANG LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM	PROGRAM Control Memory Diagnostic	CHART
VERSION	PREPARED BY J Sevigny DATE 12/75	SHEET 2 OF



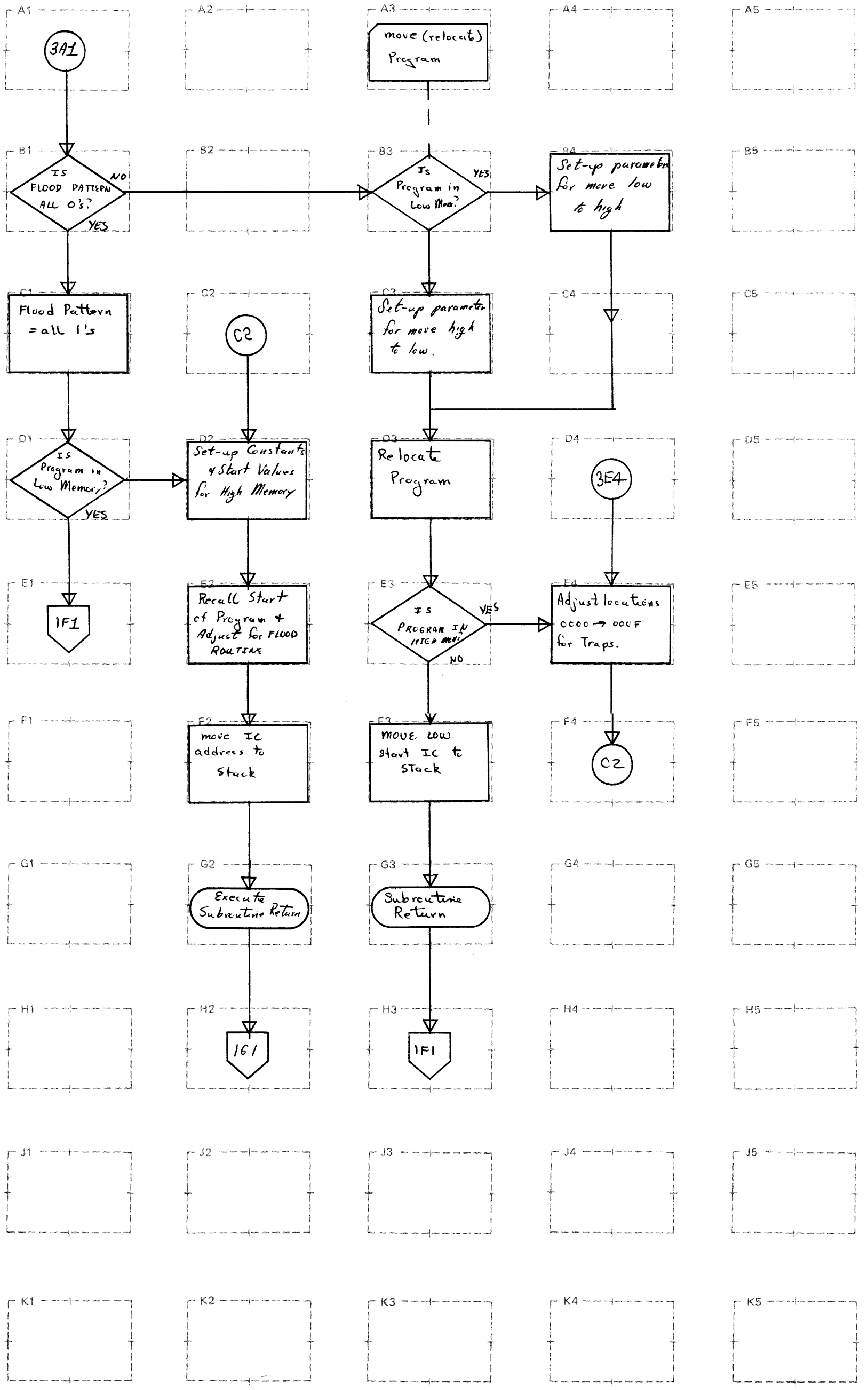
Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM	PROGRAM Control Memory Diagnostic	CHART
VERSION	PREPARED BY J. Sevigny DATE 12-75	SHEET 3 OF



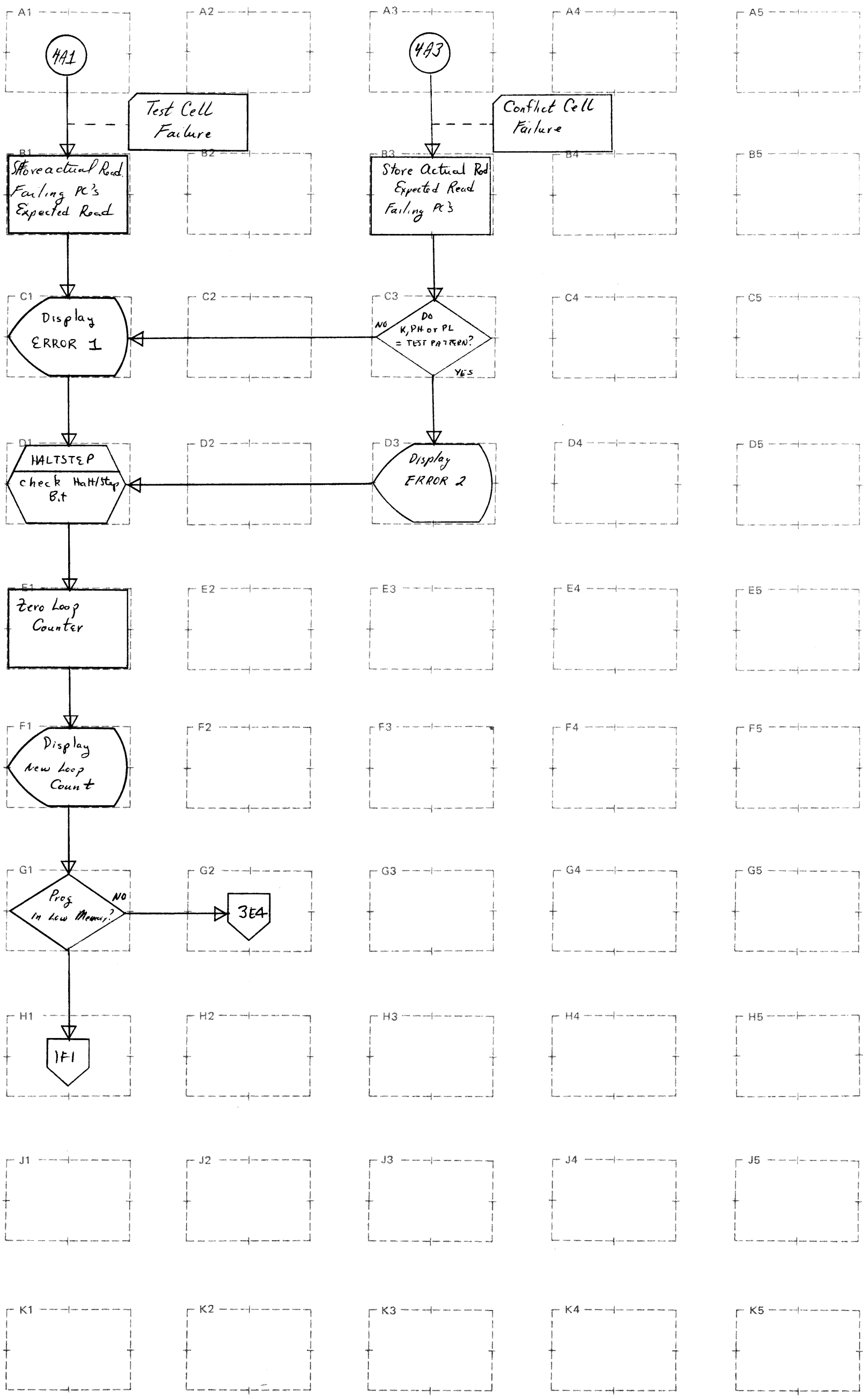
Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM	PROGRAM Control Memory Diagnostic	CHART
VERSION	PREPARED BY J. Sevigny DATE 12/15	SHEET 4 OF



Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

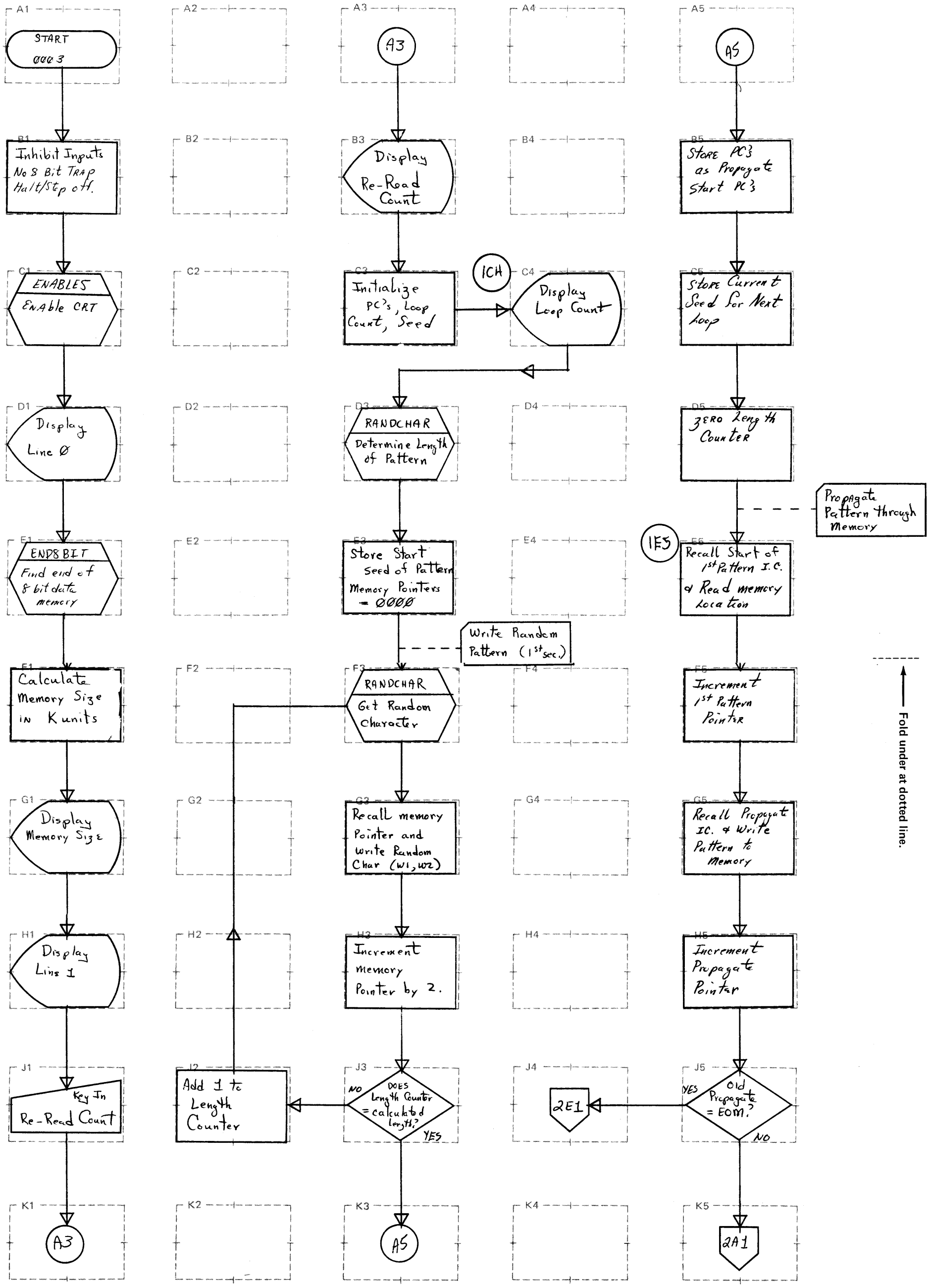
Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

```
002                TITLE     2600 MAT COPY & SEARCH
003   *
004   *            26MAT:
005   *                  This diagnostic attempts to duplicate
006   *               the OPT 5 BASIC diagnostic for memory. The
007   *               algorithm used is as follows:
008   *
009   *                  STEP 1.
010   *                       Determine the length of the RANDOM
011   *                  PATTERN to be written to all of memory.
012   *                  The length is from 1 to 256 8 bit
013   *                  RANDOM CHARACTERS.
014   *
015   *                  STEP 2.
016   *                       Write the RANDOM PATTERN into
017   *                  memory.
018   *
019   *                  STEP 3.
020   *                       Read the last written section of
021   *                  memory and write it into the next
022   *                  section of memory.
023   *
024   *                  STEP 4.
025   *                       Repeat step 3 until all of memory is
026   *                  filled with the RANDOM PATTERN.
027   *
028   *                  STEP 5.
029   *                       Read the contents of the 1st pattern
030   *                  in memory and verify that it is correct
031   *                  by re-generating the pattern.
032   *
033   *                  STEP 6.
034   *                       Using the 1st pattern section of
035   *                  memory as the original,read the remaining
036   *                  patterns in memory and verify them
037   *                  against the 1st pattern section.
038   *
039   *                  STEP 7.
040   *                       Repeat steps 5 and 6 the RE-READ
041   *                  COUNT # of times. The RE-READ COUNT is
042   *                  decided by the operator.
043   *
044   *                  MODIFIED REGISTERS
045   *                       F0 - F7, PC'S, CH, CL, SH
046   *                       AND AUX 00 - AUX 0C
047   *
048   *                  REGISTER USAGE:
049   *                        F0 - high 8 bits of END OF MEMORY
050   *                        F1 - RE-READ COUNT (USER INPUT)
051   *                        F2 - length of RANDOM PATTERN
052   *                        F3 - current byte of pattern
053   *                        F4 - constant FE
054   *                        F5 - length counter
055   *                        F6 - temporary storage
```

```
056  *                    F7 - RE-READ COUNTER
057  *                  CH,CL - read registers
058  *                     SH - status register
059  *
060  *
061  *                 AUX 1,0 - random charactern seed
062  *                 AUX 3,2 - current pattern initial seed
063  *                 AUX 4   - 1st section memory pointer
064  *                 AUX 5   - loop counter
065  *                 AUX 6   - previous section mem pointer
066  *                 AUX 7   - current section mem pointer
067  *                 AUX 9,8 - next loop seed
068  *                 AUX B,A - spare
069  *                 AUX C   - last full section in memory
070  *
071  *
072  *          SCREEN DISPLAY AND MESSAGES
073  *
074  *               line 0 - MATC&S - MEM SIZE = SSK
075  *               line 1 - RC = RR
076  *               line 2 - # XXXX
077  *
078  *                   where: MATC&S - diagnostic title
079  *                             SS - memory size
080  *                             RR - RE-READ COUNT
081  *                           XXXX - # of completed loops
082  *                                  through diagnostic
083  *
084  *          ERROR MESSAGES
085  *
086  *              1. ERR - (PATTERN) L = YYY
087  *             2A. PC'S = XXXX (EE/RR) BIT(S) N N N
088  *           OR 2B. PC'S = XXXX (EE/RR) BIT(S) N N N  OR
089  *                                      PC'S = ZZZZ
090  *              3. CH/CL PARITY BIT ERROR PC'S = XXXX
091  *
092  *                   where: PATTERN = RANDOM PATTERN
093  *                              YYY = length of pattern
094  *                             XXXX = failing PC's
095  *                               EE = expected pattern
096  *                               RR = read pattern
097  *                                N = failing bits
098  *                             ZZZZ = location of another
099  *                                    possible failure.
100  *
101  *     DISCUSSION OF ERROR CONDITIONS
102  *
103  *          1. Error 1 & 2A is displayed when a hard error
104  *             causes all of memory from the discovered
105  *             memory location to the end of memory to be
106  *             to be bad.
107  *          2. Error 1 & 2B is displayed when the
108  *             discovered point of error does not cause
109  *             the rest of memory to be bad. This may be
```

```
110  *          the result of a toggling bit prior to the
111  *          discovered error.
112  *       3. Error 3 is displayed when the system
113  *          detected bad parity but the read was found
114  *          to be good. This may be the result of a bad
115  *          parity bit. However it is not possible to
116  *          decide whether the CH or CL parity bit was
117  *          in error.
118  *
119  *    ***** HALT/STEP *****
120  *         It is possible to halt the program by keying
121  *         HALT/STEP. However the program will only HALT
122  *         after it has reportes an error. To resume with
123  *         this program, key HALT/STEP again.
124  *
125  *    ***** RESET *****
126  *         When RESET is keyed the program will execute
127  *         the POWER ON routine in BOOTSTRAP.
128  *
129  *    ***** I/O SUBROUTINES *****
130  *         The program uses BOOTSTRAP I/O ROUTINES
131  *         whenever possible.
132  *
133  *        SET PARALLEL TRAP LOACTIONS
134  *
135  *
```

SYSTEM	PROGRAM 8 Bit Data Memory Diagnostic	CHART	
VERSION	PREPARED BY J. Sevigny DATE 12/75	SHEET 1	OF



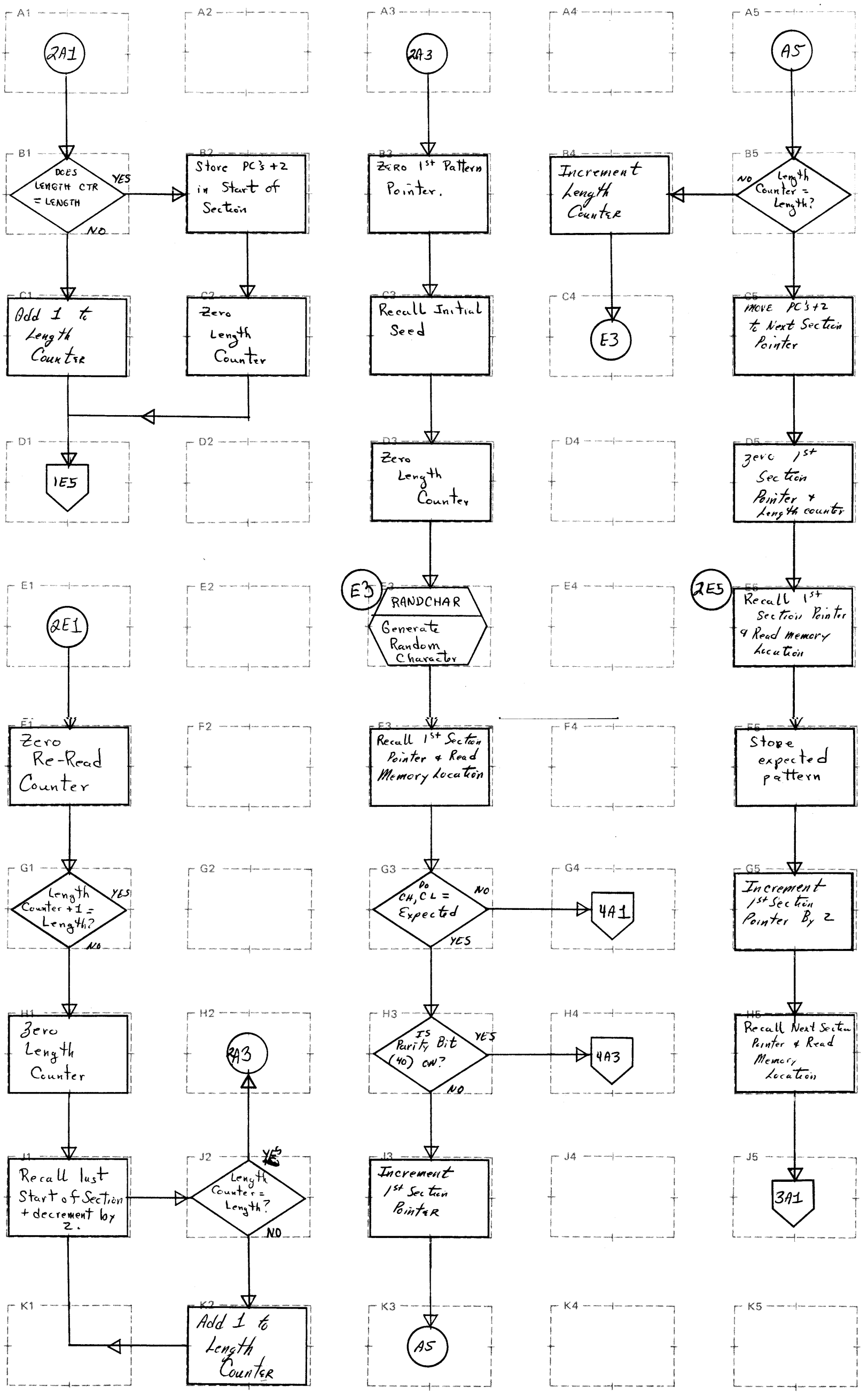
Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM	PROGRAM 8 Bit Data Memory Diagnostic	CHART
VERSION	PREPARED BY J. Sevigny DATE 12/75	SHEET 2 OF



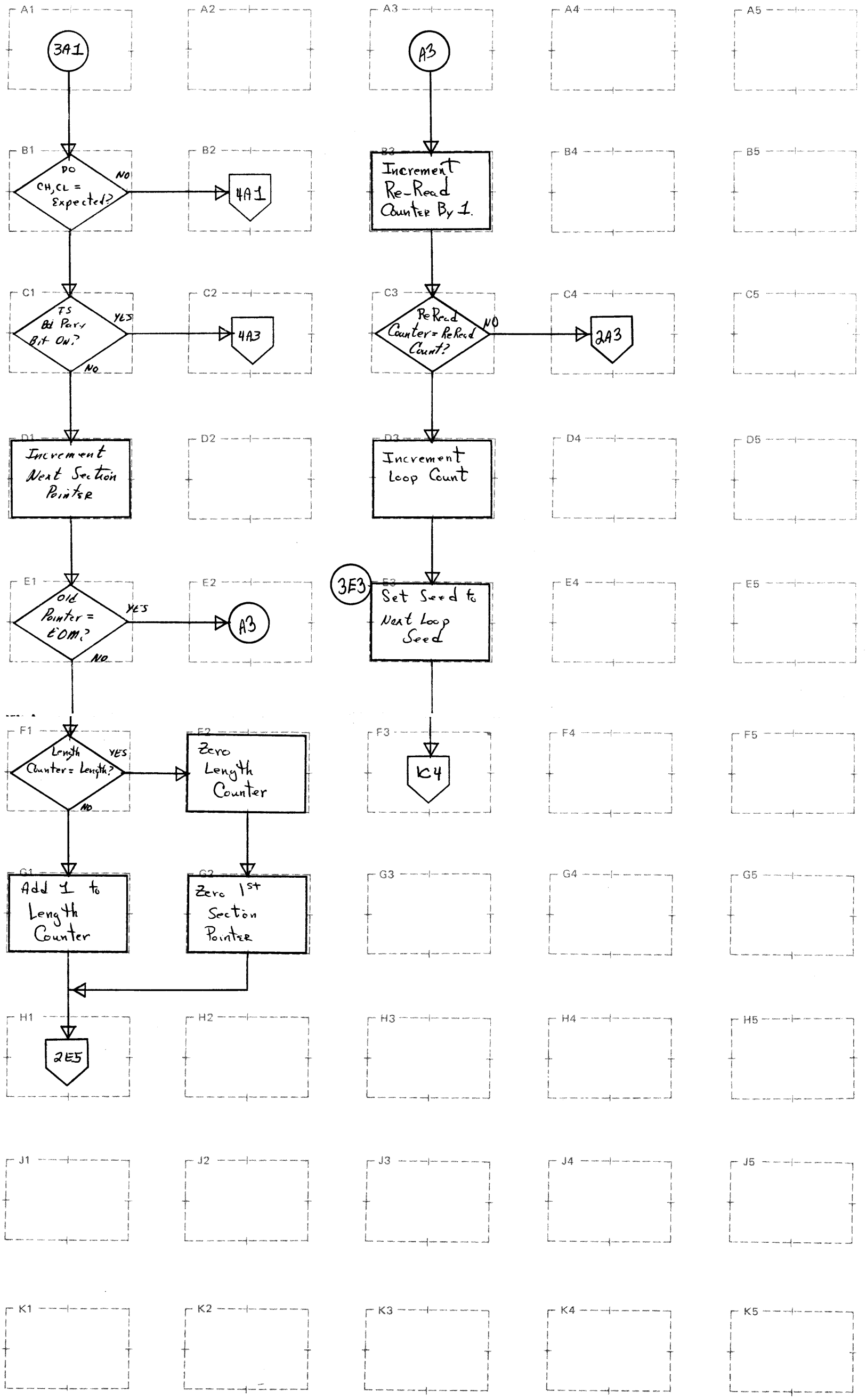
Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM	PROGRAM 8 Bit Data Memory Diagnostic	CHART
VERSION	PREPARED BY J. Sevigny DATE 12/75	SHEET 3 OF



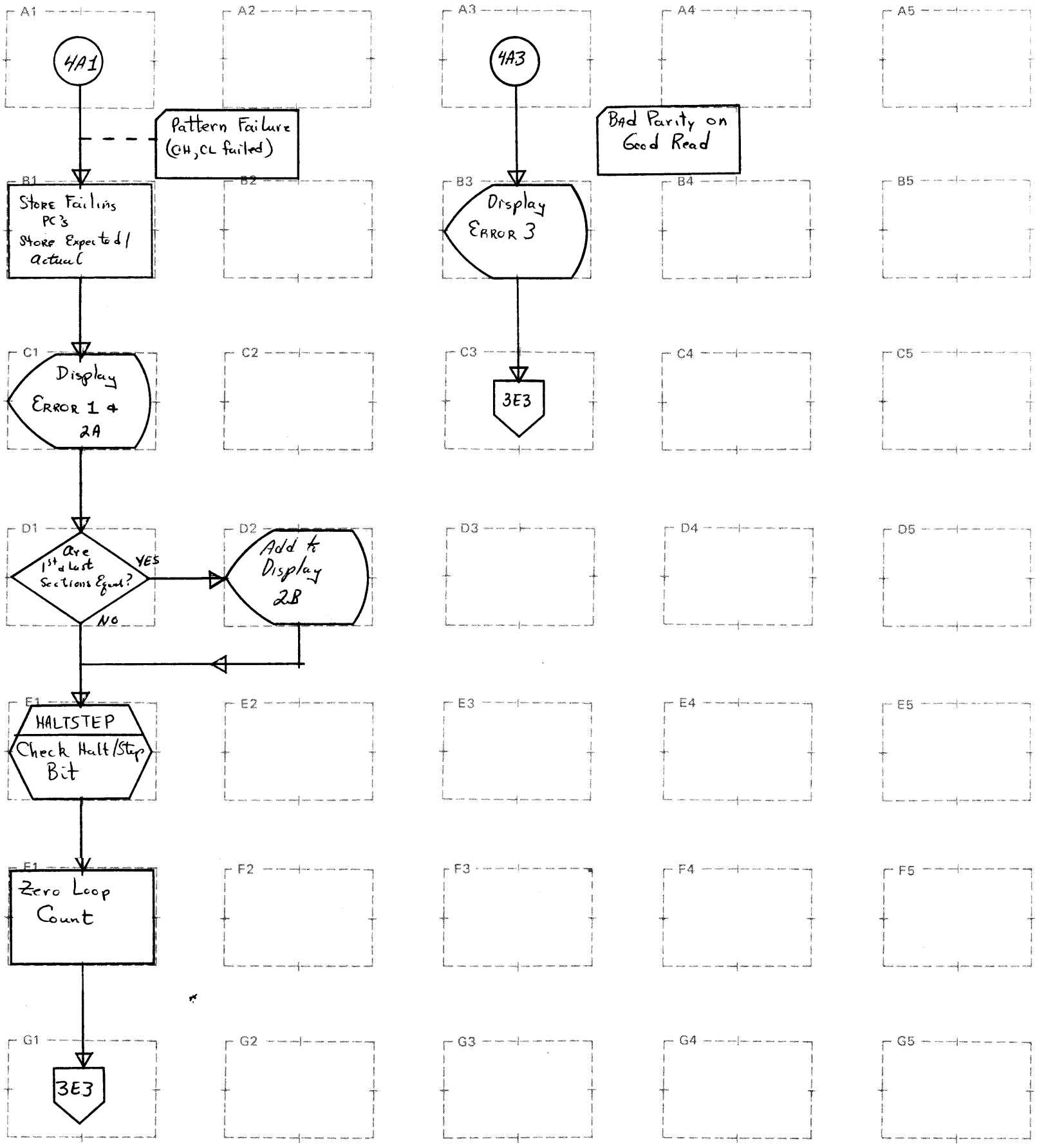
Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM _____ PROGRAM 8 Bit Data Memory Diagnostic		CHART _____
VERSION _____ PREPARED BY J. Sevigny	DATE 12/75	SHEET 4 OF _____



Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

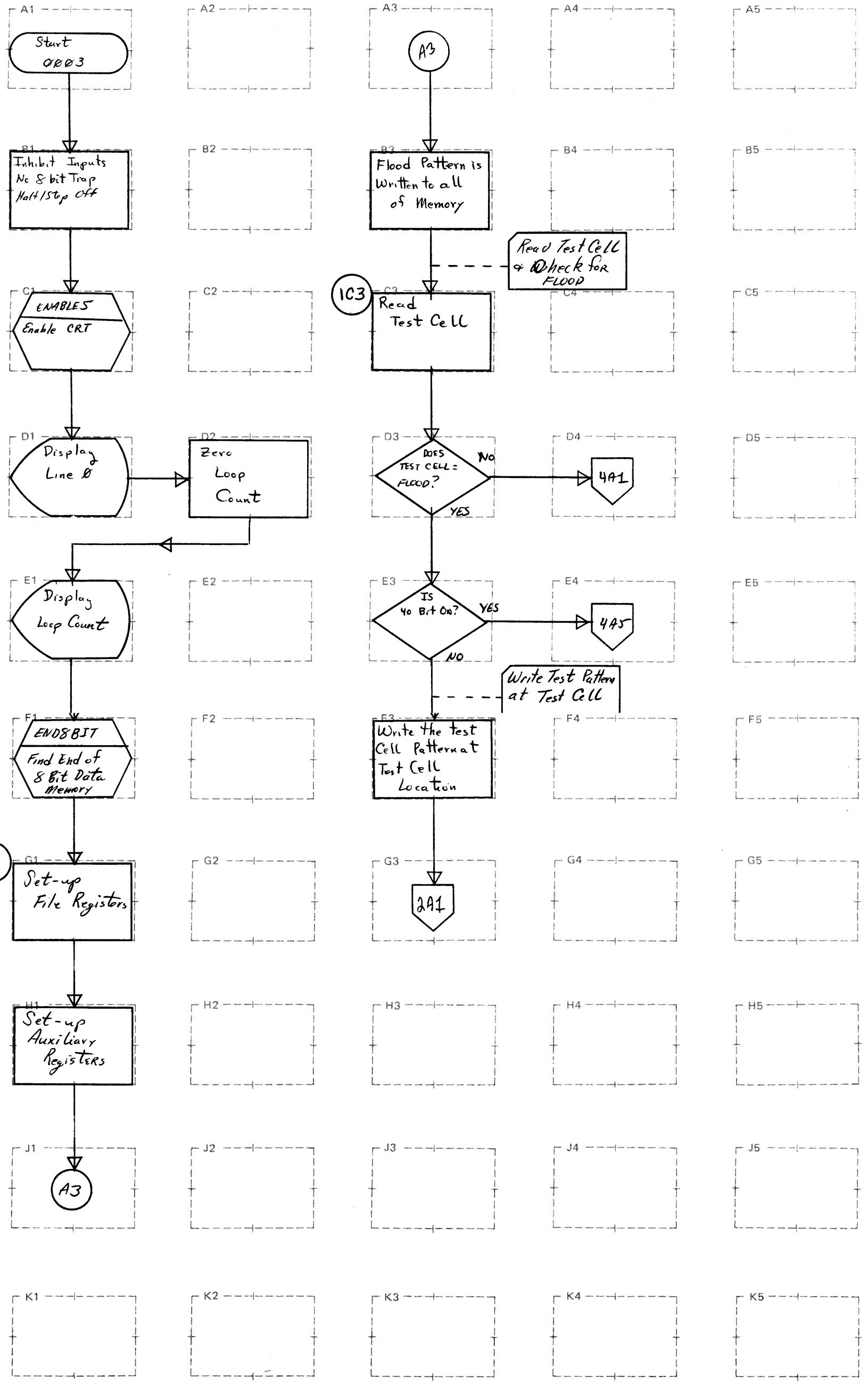
Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

```
002                TITLE     ROWPAT 8 BIT MEMORY
003   *              ROWP08:
004   *                   This routine will perform a test of 8
005   *              BIT DATA MEMORY. The algorithm used is a
006   *              MODIFIED ROWPAT.
007   *
008   *                   To explain the algorithm it is necessary
009   *              to define certain terms -
010   *                   TEST CELL - that memory location being
011   *                               tested.
012   *                   CONFLICT CELL - the other memory location
013   *                               being examined for a conflict
014   *                               with the TEST CELL
015   *                   ROW - those addresses making up a ROW
016   *                         within the memory chip of which
017   *                         the TEST CELL is one of the addr.
018   *                   COLUMN - those addresses making up a
019   *                         COLUMN within the memory chip of
020   *                         which the TEST CELL is one of the
021   *                         addresses.
022   *                   CHIP ROW - a row of memory chips located
023   *                         on a memory board.
024   *                   TEST PATTERN - that pattern written at
025   *                            the TEST CELL location.
026   *                   FLOOD - that pattern written to all
027   *                           memory locations except for the
028   *                           TEST CELL location.
029   *
030   *
031   *                   The ALGORITHM is -
032   *                        1. Write the FLOOD PATTERN of all
033   *                           zeroes throughout all of memory.
034   *                        2. Read TEST CELL and check that it
035   *                           contains the FLOOD PATTERN.
036   *                        3. Write the TEST PATTERN (the
037   *                           complement of the FLLOD PATTERN)
038   *                           at the TEST CELL.
039   *                        4. Read the TEST CELL and check for
040   *                           the TEST PATTERN.
041   *                        5. Read the CONFLICT CELL and check
042   *                           for the FLOOD PATTERN.
043   *                        6. Repeat steps 4 and 5 with the
044   *                           CONFLICT CELL at each location in
045   *                           in the COLUMN.
046   *                        7. Repeat steps 4 and 5 with the
047   *                           CONFLICT CELL at each location in
048   *                           the ROW.
049   *                        8. Write the FLOOD PATTERN at the
050   *                           TEST CELL location.
051   *                        9. Repeat steps 4 through 8 with the
052   *                           TEST CELL at each location in the
053   *                           ROW.
054   *                       10. Repeat steps 4 through 9 with the
055   *                           TEST CELL at each location in the
```

```
056  *                    CHIP ROW.
057  *                11. Repeat steps 4 through 10 with
058  *                    the TEST CELL at each location in
059  *                    memory.
060  *                12. Write a FLOOD PATTERN of all
061  *                    ones into memory.
062  *                13. Repeat steps 2 through 11.
063  *
064  *          MODIFIED REGISTERS:
065  *               F0 - F5, AUX00 - AUX10, SL, PL, PH
066  *
067  *          REGISTER USAGE:
068  *               AUX 00 - start address of CHIP ROW
069  *               AUX 01 - end address of CHIP ROW minus 1
070  *               AUX 02 - start address of COLUMN
071  *               AUX 03 - end address of COLUMN
072  *               AUX 04 - start address of ROW
073  *               AUX 05 - end address of ROW
074  *               AUX 06 - TEST CELL address
075  *               AUX 07 - CONFLICT CELL address
076  *               AUX 08 - end of memory minus 1
077  *               AUX 09 - constant 0040
078  *               AUX 0A - constant 0FC0
079  *               AUX 0B - constant F03F
080  *               AUX 0C - constant 0FFE
081  *               AUX 0D - temporary storage
082  *               AUX 0E - temporary storage
083  *               AUX 0F - temporary storage
084  *               AUX 10 - loop counter
085  *               F0     - TEST PATTERN
086  *               F1     - constant 00
087  *               F3,F2  - constant 0040
088  *               F5,F4  - working register compare
089  *
090  *          MEMORY BOARD LAYOUT (CHIP #'S)
091  *
092  *               !   EVEN    !   ODD     !
093  *               ! ADDRESS   ! ADDRESS   !
094  *               ------------------------
095  *               !16      09!08      01! 2ND 4K
096  *               !          !          !
097  *               ------------------------
098  *               !32      25!24      17! 1ST 4K
099  *               !          !          !
100  *               ------------------------
101  *
102  *          SCREEN DISPLAY AND MESSAGES
103  *
104  *               line # 0 - ROWPAT TEST - 8 BIT RAM
105  *               line # 1 - # LLLL
106  *
107  *                    where: line 0 = diagnostic title
108  *                           LLLL   = # of completed loops
109  *                                    through diagnostic
```

```
110  *
111  *          ERROR MESSAGES
112  *
113  *               1. FAILURE AT PC'S XXXX      BOARD # N
114  *                  (EE/RR)  CHIP(S) CC CC CC CC CC CC CC
115  *
116  *               2. CONFLICT BETWEEN YYYY AND ZZZZ
117  *
118  *               3. CH/CL PARITY BIT ERROR PC'S = XXXX
119  *
120  *                    where: XXXX = failing PC's
121  *                             N = board # ( 1,2,3 OR 4)
122  *                            EE = expected pattern
123  *                            RR = actual pattern
124  *                            CC = chip #(s) in error
125  *                          YYYY = TEST CELL address
126  *                          ZZZZ = CONFLICT CELL addr
127  *
128  *          DISCUSSION OF ERROR MESSAGES
129  *
130  *               ERROR 1 occurs when either the TEST CELL
131  *          or the CONFLICT CELL failed to maintain it's
132  *          expected pattern.
133  *
134  *               ERROR 2 occurs when the CONFLICT CELL
135  *          fails to maintain it's expected pattern and
136  *          the actual pattern equals the TEST PATTERN.
137  *          This error may indicate an addressing problem.
138  *
139  *               ERROR 3 occurs when the actual pattern
140  *          equals the expected, however the hardware
141  *          detected a parity problem. This error may
142  *          indicate a parity bit error.
143  *
144  *          ***** RESET *****
145  *               When RESET is keyed the program will
146  *          execute the 'MOUNT SYSTEM DISK' routine
147  *          located in BOOTSTRAP.
148  *
149  *          ***** HALT/STEP *****
150  *               The HALT/STEP key may be used to HALT the
151  *          program. The program will HALT ONLY after an
152  *          error condition has beed displayed. To resume
153  *          key HALT/STEP again.
154  *
155  *          ***** I/O SUBROUTINES *****
156  *               This program will use the BOOTSTRAP
157  *          I/O subroutines whenever possible.
158  *
159               SYMBL     JSM0D02$
160  *
161               B         PE24
```

SYSTEM	PROGRAM 8 Bit Memory Diagnostic #2	CHART
VERSION	PREPARED BY J. Sevigny DATE 12/75	SHEET 1 OF



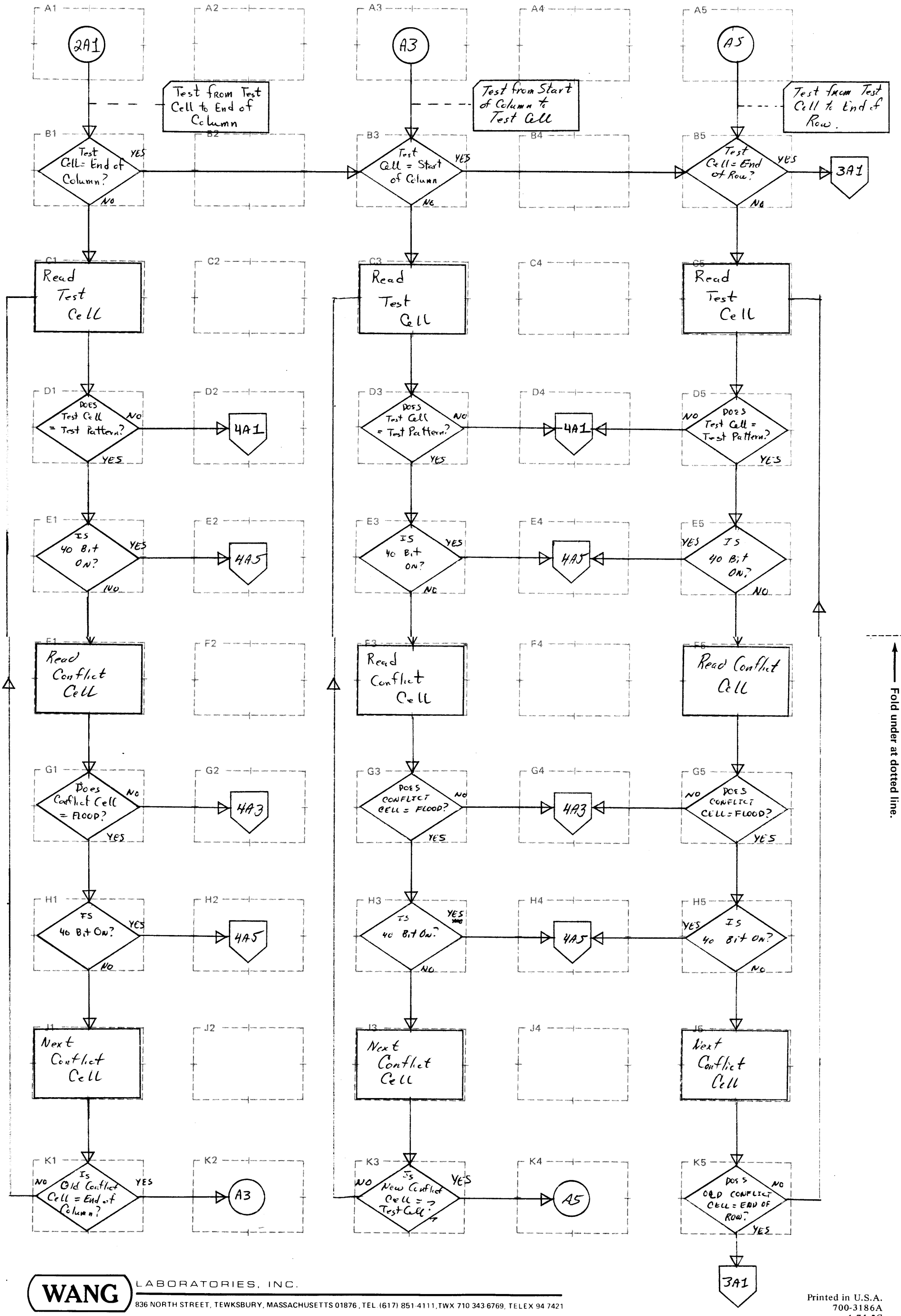
Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM	PROGRAM 8 Bit Data Memory Diagnostic #2	CHART	
VERSION	PREPARED BY I. Sevigny DATE 12/75	SHEET 2	OF



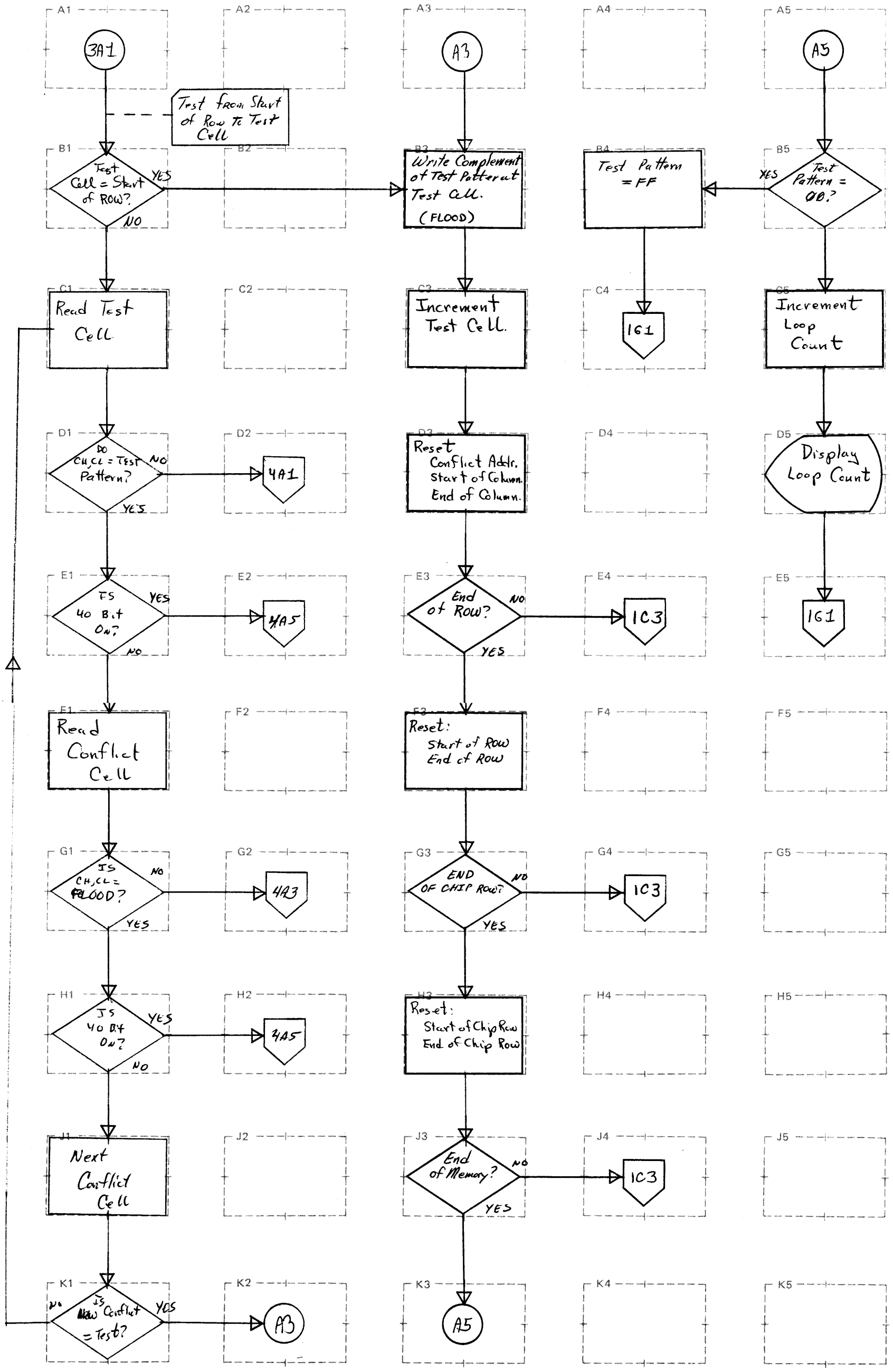
Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM	PROGRAM 8 Bit Data Memory Diagnostic #2	CHART	
VERSION	PREPARED BY J. Sevigny DATE 12/75	SHEET 3	OF



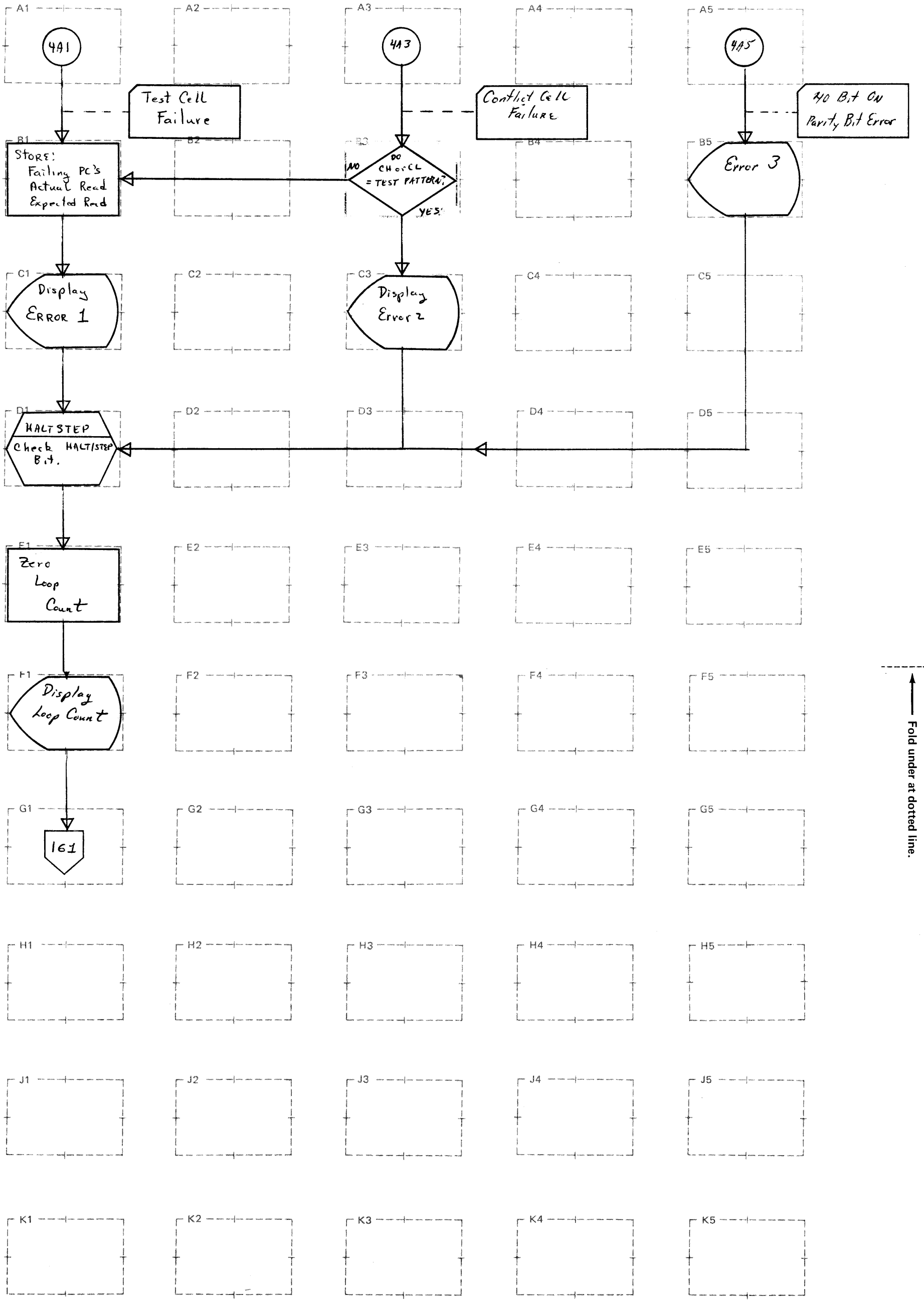
Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM	PROGRAM 8 Bit Data Memory Diagnostic #2	CHART
VERSION	PREPARED BY J. Sevigny DATE 12/75	SHEET 4 OF



Fold under at dotted line.

Fold under at dotted line.

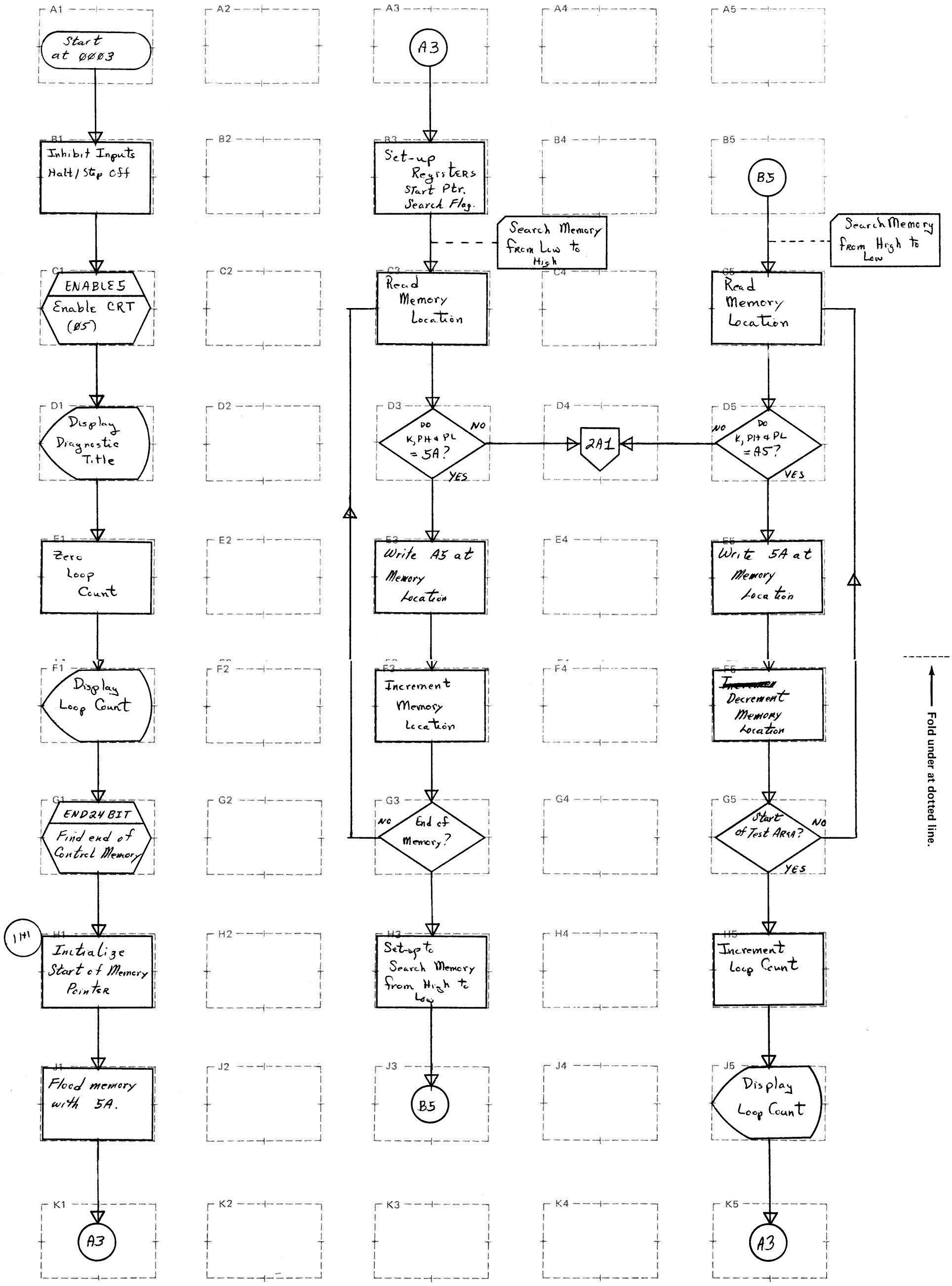
WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

```
002          TITLE    24 BIT ADDRESSING TEST
003  *     24 BIT ADDRESSING TEST
004  *       This diagnostic will read from the end of the
005  *     program to the end of memory and vica versa
006  *     searching for a memory location which may have
007  *     been changed by a previous write instruction.
008  *
009  *       The algorithm goeslike this -
010  *
011  *       STEP 1. flood memory with 5A's (0101 1010)
012  *       STEP 2. starting at 0200 and searching forward
013  *            to the end of memory, read each location
014  *            and check for the 5A pattern. If location
015  *            is okay then write an A5 (1010 0101) at
016  *            that location.
017  *       STEP 3. starting at the end of memory and
018  *            searching down to location 0200, read
019  *            each location and check for an A5. If
020  *            the location is okay then write a 5A
021  *            (0101 1010) at that location.
022  *
023  *       MODIFIED REGISTERS:
024  *            AUX 00 - AUX 07, AUX 11
025  *            F0 - F7
026  *
027  *       REGISTER USAGE -
028  *            F1,F0 - end of memory
029  *            F2    - constant 5A
030  *            F3    - constant A5
031  *            F5,F4 - start of test area (0200)
032  *            F6    - search direction flag
033  *            F7    - constant 00
034  *
035  *            AUX 00 - end of memory
036  *            AUX 01 - memory pointer
037  *            AUX 02 - failing memory pointer
038  *            AUX 03 - loop count
039  *            AUX 04 - failure pattern (PH,PL) (search)
040  *            AUX 05 - failure pattern (K) (search)
041  *            AUX 06 - initial failing pattern (PH,PL)
042  *            AUX 07 - initial failing pattern (K)
043  *            AUX 11 - storage for WCM
044  *
045  *     *****  HALT/STEP  *****
046  *            The program may be interrupted after
047  *       displaying an error message by keying
048  *       HALT/STEP. To resume after the program has
049  *       HALTED key HALT/STEP again.
050  *
051  *     *****  RESET  *****
052  *            When RESET is keyed the program will
053  *       execute the 'MOUNT SYSTEM PLATTER' routine
054  *       in BOOTSTRAP.
055  *
```

```
056  *        ***** I/O SUBROUTINES *****
057  *             The program uses the BOOTSTRAP I/O
058  *        routines whenever possible.
059  *
060  *        SCREEN AND ERROR MESSAGES
061  *
062  *             line #0   24 BIT ADDRESSING TEST
063  *             line #1   # LLLL
064  *
065  *                  where line #0 = TITLE
066  *                        LLLL    = # of completed loops
067  *
068  *        ERROR TYPE 0
069  *        ERROR BETWEEN XXXX AND YYYY (ZZZZZZ)
070  *
071  *        ERROR TYPE 1
072  *        POSSIBLE CHIP FAILURE PC'S = XXXX (ZZZZZZ)
073  *
074  *             where XXXX = failing memory location
075  *                   YYYY = location causing error
076  *                 ZZZZZZ = XOR (expected/actual)
077  *
078  *        DISCUSSION OF ERROR TYPES
079  *             An ERROR TYPE 0 is caused when writing to
080  *        location YYYY all or part of the pattern is
081  *        written at location XXXX. This is indicative
082  *        of an addressing problem. However, never rule
083  *        out a chip failure.
084  *
085  *             An ERROR TYPE 1 is caused when the
086  *        expected pattern is not found and no other
087  *        location seems to cause the problem. This may
088  *        be a chip or chips error.
089  *
090            SYMBL     JSMOD02$
091  *
092            B         PE24
```

SYSTEM _____ PROGRAM CONTROL MEMORY ADDRESSING TEST		CHART _____
VERSION _____ PREPARED BY J. Sevigny	DATE 12/75	SHEET 1 OF _____



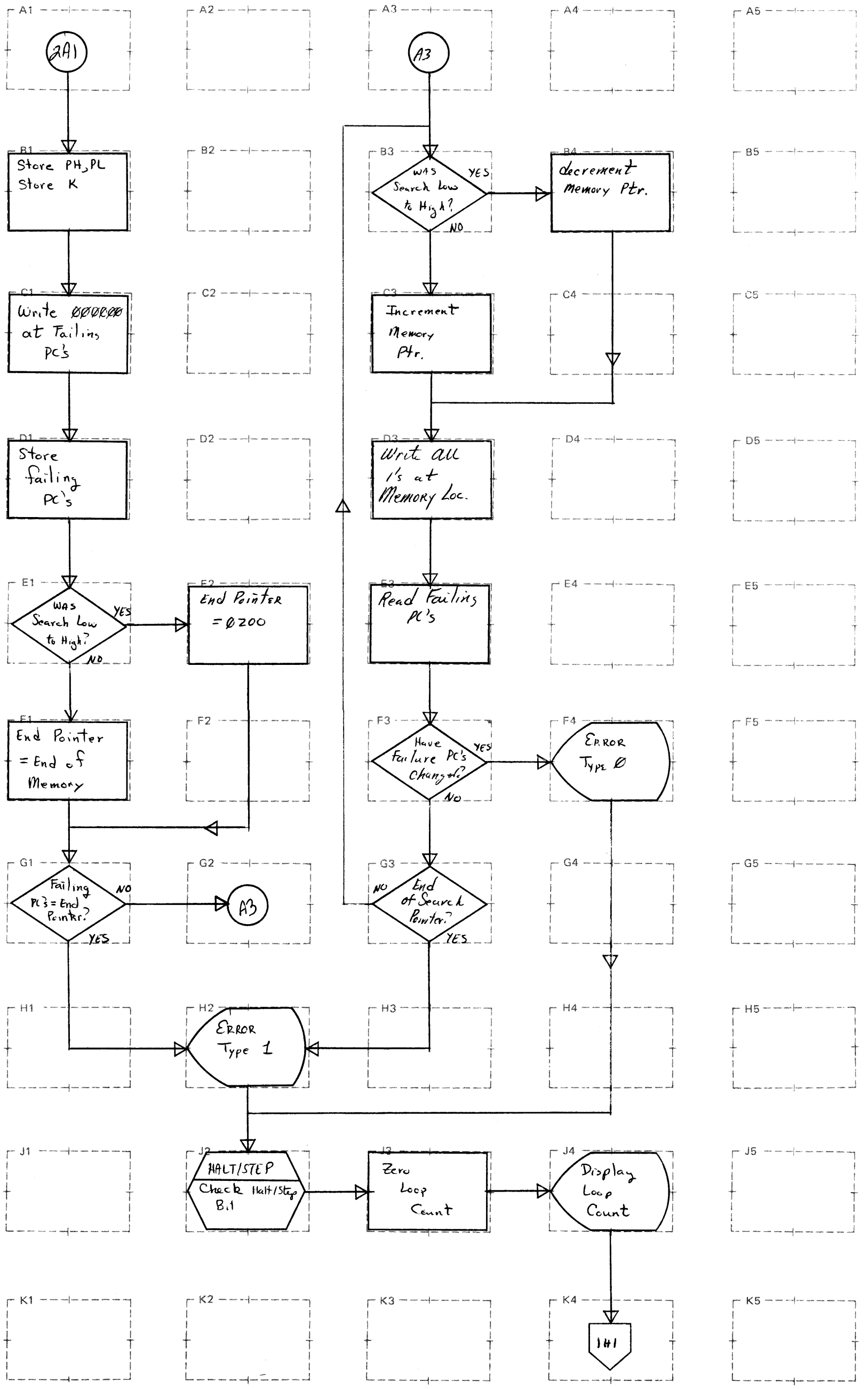
Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM	PROGRAM Control Memory Addressing Test	CHART
VERSION	PREPARED BY J. Sevigny DATE 12/25	SHEET 2 OF



Fold under at dotted line.

Fold under at dotted line.

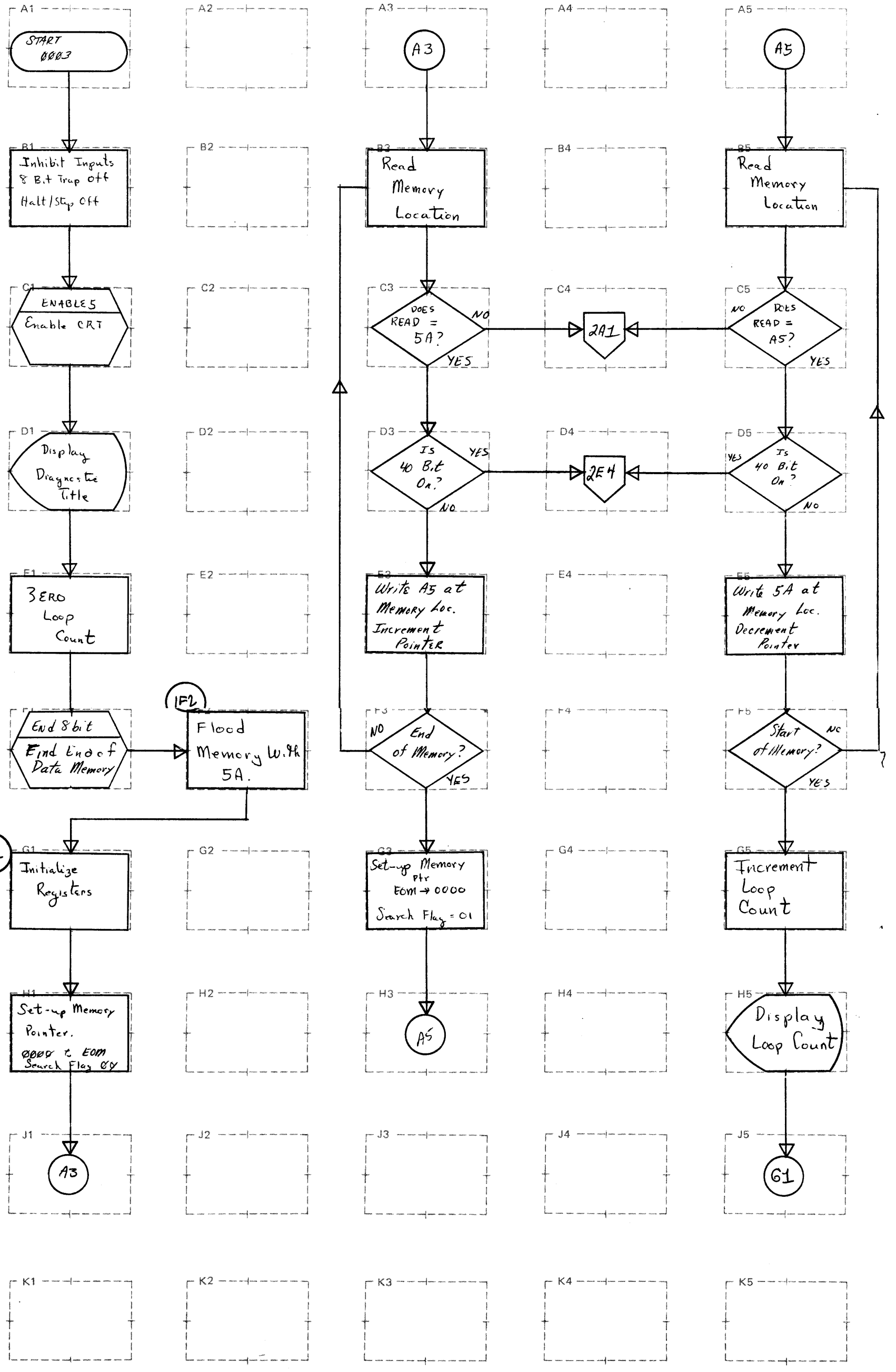
WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

```
002                TITLE    8 BIT ADDRESSING TEST
003  *          8 BIT ADDRESSING TEST
004  *            This diagnostic will read from the start of
005  *          memory to the end of memory and vica versa
006  *          searching for a conflict between two addresses.
007  *
008  *            The ACORITHM goes like this -
009  *
010  *            STEP 1. Flood memory with 5A's (0101 1010)
011  *            STEP 2. For each location from the start to
012  *                 the end of memory, read the location,
013  *                 check for a 5A and write an A5 in it's
014  *                 place.
015  *            STEP 3. For each location from the end to the
016  *                 start of memory, read the location, check
017  *                 for an A5 and write a 5A in it's place.
018  *
019  *            MODIFIED REGISTERS:
020  *                 AUX 00 - AUX 03
021  *                 F0 - F7
022  *
023  *            REGISTER USAGE -
024  *                 F1,F0 - end of memory (XFFF)
025  *                 F2    - constant 5A
026  *                 F3    - constant A5
027  *                 F5,F4 - start of memory (0000)
028  *                 F6    - search direction flag
029  *                 F7    - failing read value
030  *
031  *                 AUX 00 - end of memory (XFFF)
032  *                 AUX 01 - end of memory plus 1 ((X+1)000)
033  *                 AUX 02 - failing memory pointer
034  *                 AUX 03 - loop counter
035  *
036  *          *****  HALT/STEP  *****
037  *                 The program may be interrupted by keying
038  *            HALT/STEP. However, the program will only HALT
039  *            after displaying an error condition.
040  *                 To resume the program, key HALT/STEP
041  *            again.
042  *
043  *          *****  RESET  *****
044  *                 When RESET is keyed, the program will
045  *            execute the 'MOUNT SYSTEM PLATTER' routine
046  *            located in BOOTSTRAP.
047  *
048  *          *****  I/O SUBROUTINES  *****
049  *                 This program uses the BOOTSTRAP I/O
050  *            routines whenever possible.
051  *
052  *            SCREEN AND ERROR MESSAGES
053  *
054  *                 line #0   8 BIT ADDRESSING TEST
055  *                 line #1   # LLLL
```

```
056  *
057  *                    where line #0 = TITLE
058  *                          LLLL    = # of completed loops
059  *
060  *          ERROR TYPE 0
061  *          ERROR BETWEEN XXXX AND YYYY (ZZ)
062  *
063  *          ERROR TYPE 1
064  *          POSSIBLE CHIP FAILURE PC'S = XXXX (ZZ)
065  *
066  *          ERROR TYPE 2
067  *          CH/CL PARITY BIT ERROR PC'S = XXXX
068  *
069  *               where XXXX = failing memory location
070  *                     YYYY = conflict location
071  *                       ZZ = XOR (expected/actual)
072  *
073  *          DISCUSSION OF ERROR TYPES
074  *               An ERROR TYPE 0 is caused when writing to
075  *          location YYYY all or part of the pattern is
076  *          written at location XXXX. This is indicative
077  *          of an addressing problem.
078  *
079  *               An ERROR TYPE 1 is caused when the
080  *          expected pattern is not found and no other
081  *          location seems to cause the problem. This may
082  *          be a bad memory chip or chips.
083  *
084  *               An ERROR TYPE 2 is caused when the system
085  *          detected a parity problem but the read value
086  *          was as expected. This may be a parity bit
087  *          error.
088  *
089       SYMBL     JSM0002$
090  *
091       B         PE24
092       B         MOUNT
```

SYSTEM	PROGRAM 8 Bit Data (Memory) Addressing Test	CHART
VERSION	PREPARED BY J Sevigny DATE 12/75	SHEET 1 OF



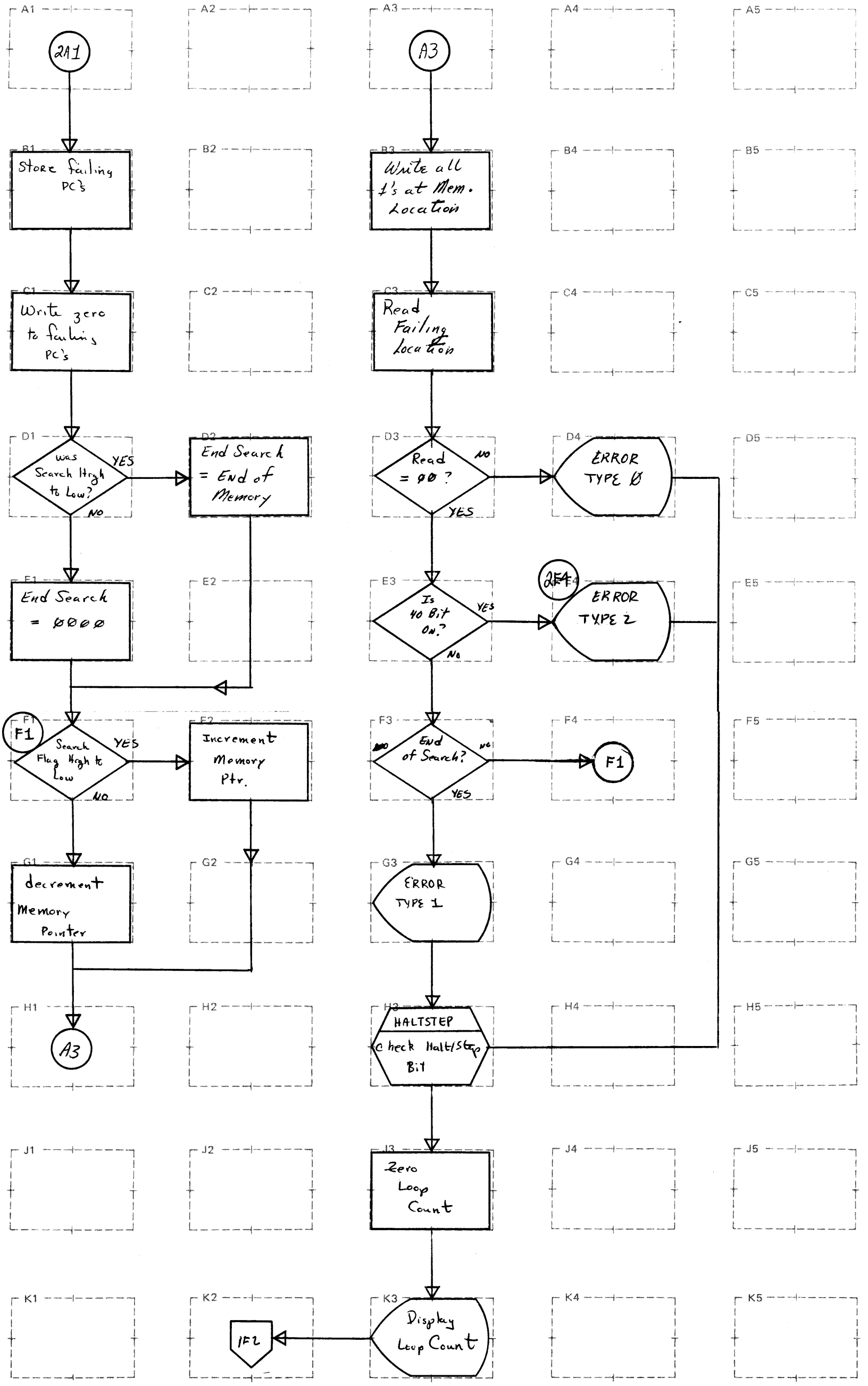
Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM	PROGRAM 8 Bit Data Memory Addressing Test	CHART
VERSION	PREPARED BY J. Sevigny DATE 12/15	SHEET 2 OF



Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

```
002                TITLE     REGISTER TEST
003  *          REGTEST:
004  *                This diagnostic was designed to test the
005  *             register chip in such a manner as to determine
006  *             whether bits have gone bad or a conflict in
007  *             addressing exists.
008  *
009  *                The test is performed in the following
010  *             manner:
011  *
012  *                1. Flood all registers with all zeroes.
013  *                2. For each register (F0-F7,CH,CL,PH,PL
014  *                     ,SL AND K)
015  *                     A) move 00 to test pattern
016  *                     B) set test register to test pattern
017  *                     C) check if test register holds test
018  *                        pattern.
019  *                     D) check if other registers have
020  *                        been changed.
021  *                     E) add 01 to test pattern
022  *                     F) repeat steps B through E until
023  *                        test pattern equals 00.
024  *                     G) repeat steps A through F until
025  *                        all registers have been tested.
026  *
027  *          MODIFIED REGISTERS:
028  *                F0-F7, PL, PH, CL, CH, SL AND K
029  *                AUX 00, 01
030  *
031  *          POSSIBLE PROGRAM HANG-UPS
032  *                A quick check of register K is made
033  *             prior to any displaying being done.
034  *             Should 'TESTING REGISTER' fail to appear
035  *             on the CRT then register K is failing
036  *             which may hinder I/O.
037  *
038  *           *****  HALT/STEP  *****
039  *                The program may be interrupted by
040  *             keying HALT/STEP. However, the program
041  *             will HALT only after displaying an error
042  *             condition.
043  *                To resume the program, key HALT/STEP
044  *             again.
045  *
046  *           *****  RESET  *****
047  *                When RESET is keyed the program will
048  *             execute the 'MOUNT SYSTEM PLATTER'
049  *             routine located in BOOTSTRAP.
050  *
051  *           *****  I/O SUBROUTINES  *****
052  *                The program uses the BOOTSTRAP I/O
053  *             routines whenever possible.
054  *
055  *          SCREEN DISPLAYS AND MESSAGES:
```

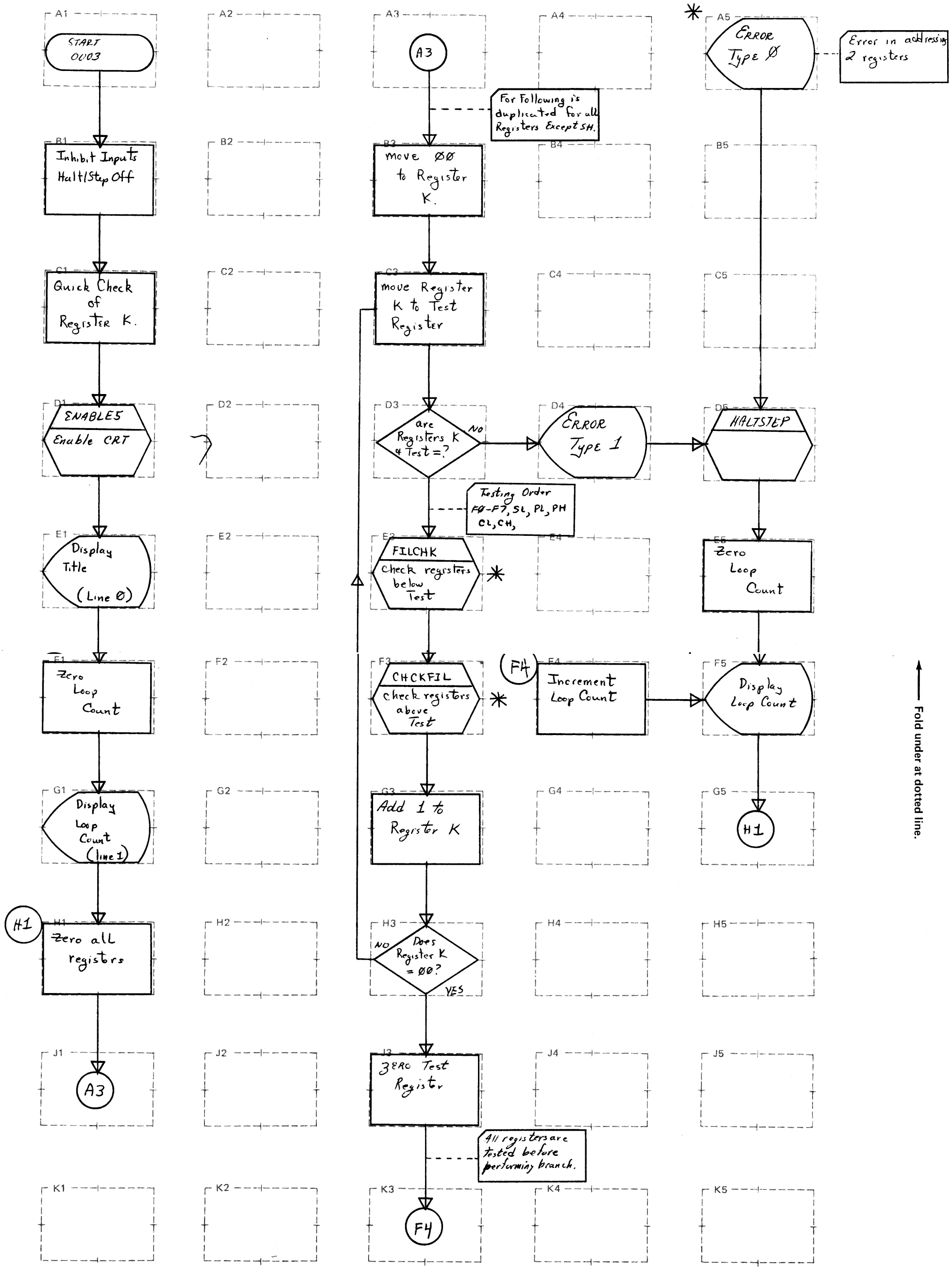
```
056  *
057  *          line 0 - REGISTER TEST
058  *          line 1 - # LLLL
059  *
060  *               where: line 0 = Title
061  *                      LLLL = # of completed loops
062  *                             through diagnostic
063  *
064  *     ERROR TYPE 0
065  *     REGISTER XX AND YY ERROR (XX) (CR/LF)
066  *
067  *     ERROR TYPE 1
068  *     REGISTER XX ERROR (ZZ) (CR/LF)
069  *
070  *     ERROR DISCUSSION
071  *          An ERROR TYPE 0 is caused when while
072  *     testing register XX, register YY was found
073  *     not to contain the expected.
074  *
075  *          An ERROR TYPE 1 is caused when
076  *     register XX fails to hold the test pattern.
077  *
078  *
```

SYSTEM _____ PROGRAM Registee Test

VERSION _____ PREPARED BY J Sevigny DATE 12-75

CHART _____

SHEET 1 OF _____



Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

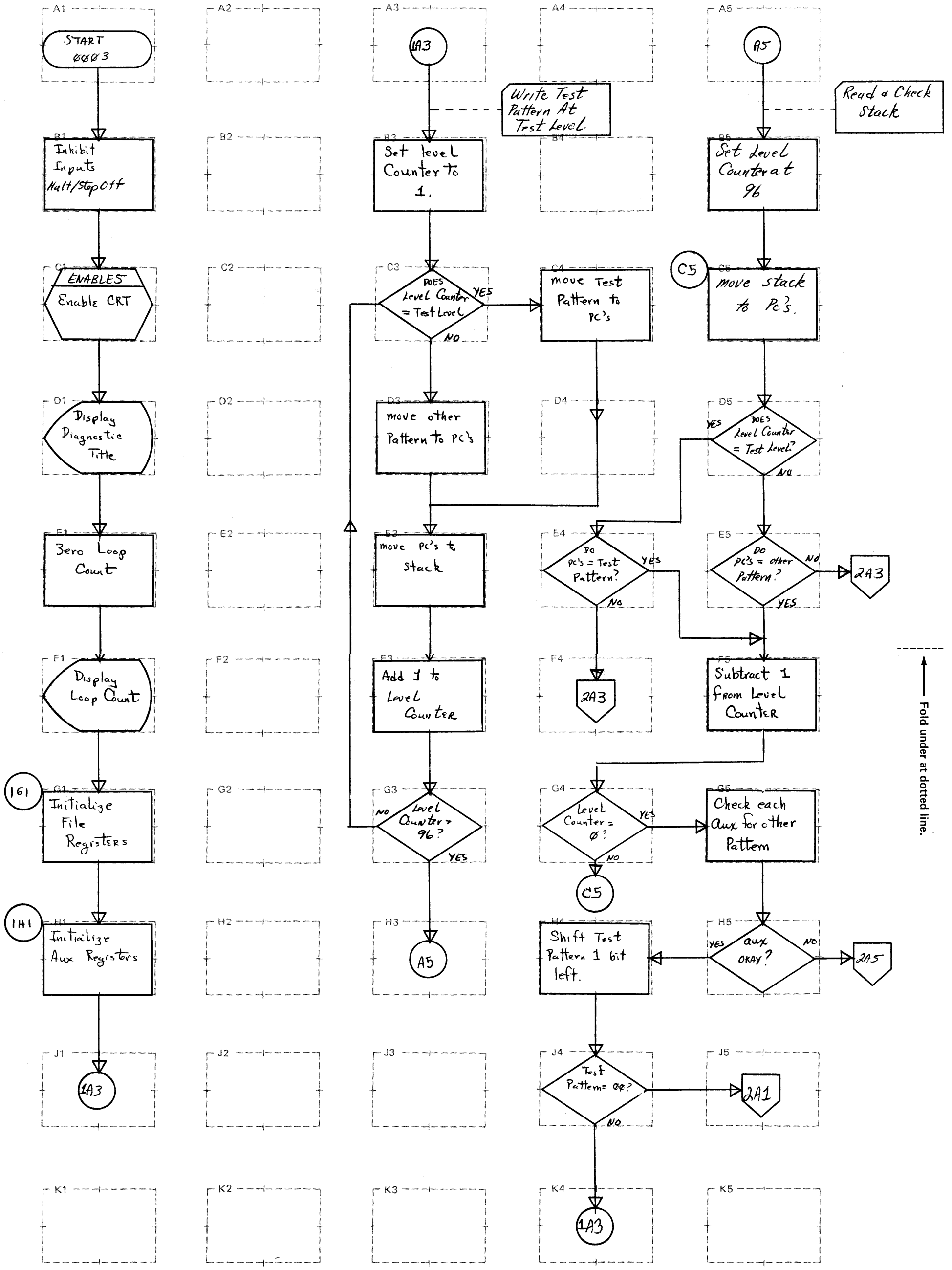
```
002             TITLE    STACK / AUXILIARY TEST
003  *          ST/AUX:  (STAUX1 AND STAUX2)
004  *               This routine checks the STACK to determine
005  *          whether each level in the STACK will -
006  *               1) hold a particular bit pattern.
007  *               2) have any effect upon another STACK
008  *                level.
009  *            and 3) have any effect on an AUXILIARY
010  *                register.
011  *
012  *               Two distinct patterns are used by the
013  *          routine and these are -
014  *               1) TEST PATTERN - the pattern which is
015  *          expected to be at the TEST LEVEL of the STACK.
016  *                  There are 8 TEST PATTERNS -
017  *                     1.   0000 0001   0000 0001
018  *                     2.   0000 0010   0000 0010
019  *                     3.   0000 0100   0000 0100
020  *                     4.   0000 1000   0000 1000
021  *                     5.   0001 0000   0001 0000
022  *                     6.   0010 0000   0010 0000
023  *                     7.   0100 0000   0100 0000
024  *          and        8.   1000 0000   1000 0000
025  *
026  *               2) OTHER PATTERN - either all 0's or all
027  *          1's depending the point of execution.
028  *
029  *               The testing is performed via the following
030  *          ALGORITHM -
031  *               PASS 1:
032  *                    A) The AUXILIARY registers are
033  *               initialized with all 0's (OTHER PATTERN)
034  *                    B) The TEST PATTERN is set to (01 01)
035  *                    C) The current TEST PATTERN is
036  *               written into the current TEST LEVEL of the
037  *               STACK. While all other levels are written
038  *               with the OTHER PATTERN.
039  *                    D) Each level of the STACK is read to
040  *               determine whether that level holds the
041  *               expected pattern.
042  *                    E) All of the AUXILIARY registers are
043  *               checked to determine whether they still
044  *               contain the OTHER PATTERN.
045  *                    F) The next TEST PATTERN is generated
046  *               and steps B, C and D are repeated.
047  *                    G) The STACK LEVEL is incremented
048  *               and TEST PATTERN is set to (01 01) and
049  *               steps B through F are repeated until all
050  *               96 STACK levels have been checked.
051  *
052  *            PASS 2:
053  *
054  *                    H) The AUXILIARY registers are set to
055  *               all 1's (the OTHER PATTERN) and steps B
```

```
056  *          through G are repeated.
057  *
058  *      MODIFIED REGISTERS:
059  *           F0 - F7, PL, PH, and K.
060  *
061  *      REGISTER USAGE:
062  *           F0 - TEST PATTERN
063  *           F1 - TEST LEVEL within STACK
064  *           F2 - level counter
065  *           F3 - max level 96
066  *           F4 - constant 1
067  *           F5 - constant 0
068  *           F6 - OTHER PATTERN (all 0's [00] or
069  *                     all 1's [FF])
070  *           F7 - AUXILIARY register pointer
071  *           K  - I/O register
072  *
073  *      *****  HALT/STEP  *****
074  *           The program may be interrupted by
075  *      keying HALT/STEP. However, a HALT will be
076  *      executed only after an error message is
077  *      displayed.
078  *           To resume executing the program, key
079  *      HALT/STEP again.
080  *
081  *      *****  RESET  *****
082  *           When RESET is keyed the program will
083  *      execute the 'MOUNT SYSTEM PLATTER' routine
084  *      located in BOOTSTRAP.
085  *
086  *      *****  I/O ROUTINES  *****
087  *           The program uses BOOTSTRAP I/O
088  *      routine whenever possible.
089  *
090  *      SCREEN DISPLAYS AND MESSAGES
091  *
092  *           line 0 - STACK TEST
093  *           line 1 - # LLLL
094  *
095  *                where LLLL = # of completed loops
096  *                             through diagnostic
097  *
098  *       STACK FAILURE
099  *          STACK FAILURE (XXXX)
100  *
101  *       AUXILIARY FAILURE
102  *          AUX YY FAILURE (XXXX)
103  *
104  *                where XXXX - XOR of the expected
105  *                             and actual.
106  *
107  *       ERROR DISCUSSION
108  *           A STACK FAILURE is when a particular
109  *        STACK LEVEL fails to maintain the expected
```

```
110  *          pattern. This failure may be due to chip
111  *          failure or bad addressing. If the chip is
112  *          replaced and the problem remains it
113  *          probably is an addressing problem.
114  *
115  *               An AUX FAILURE is caused when a
116  *          particular AUXILIARY register fails to
117  *          maintain the expected pattern. Again, the
118  *          problem may be bad chips or possibliy
119  *          bad addressing lines.
120  *
121              SYMBL     JSM0002$
122  *
123              B         PE24
```

SYSTEM _____ PROGRAM STACK (Aux) TEST | CHART _____

VERSION _____ PREPARED BY T. Sevigny DATE 12/75 | SHEET 1 OF _____



Fold under at dotted line.

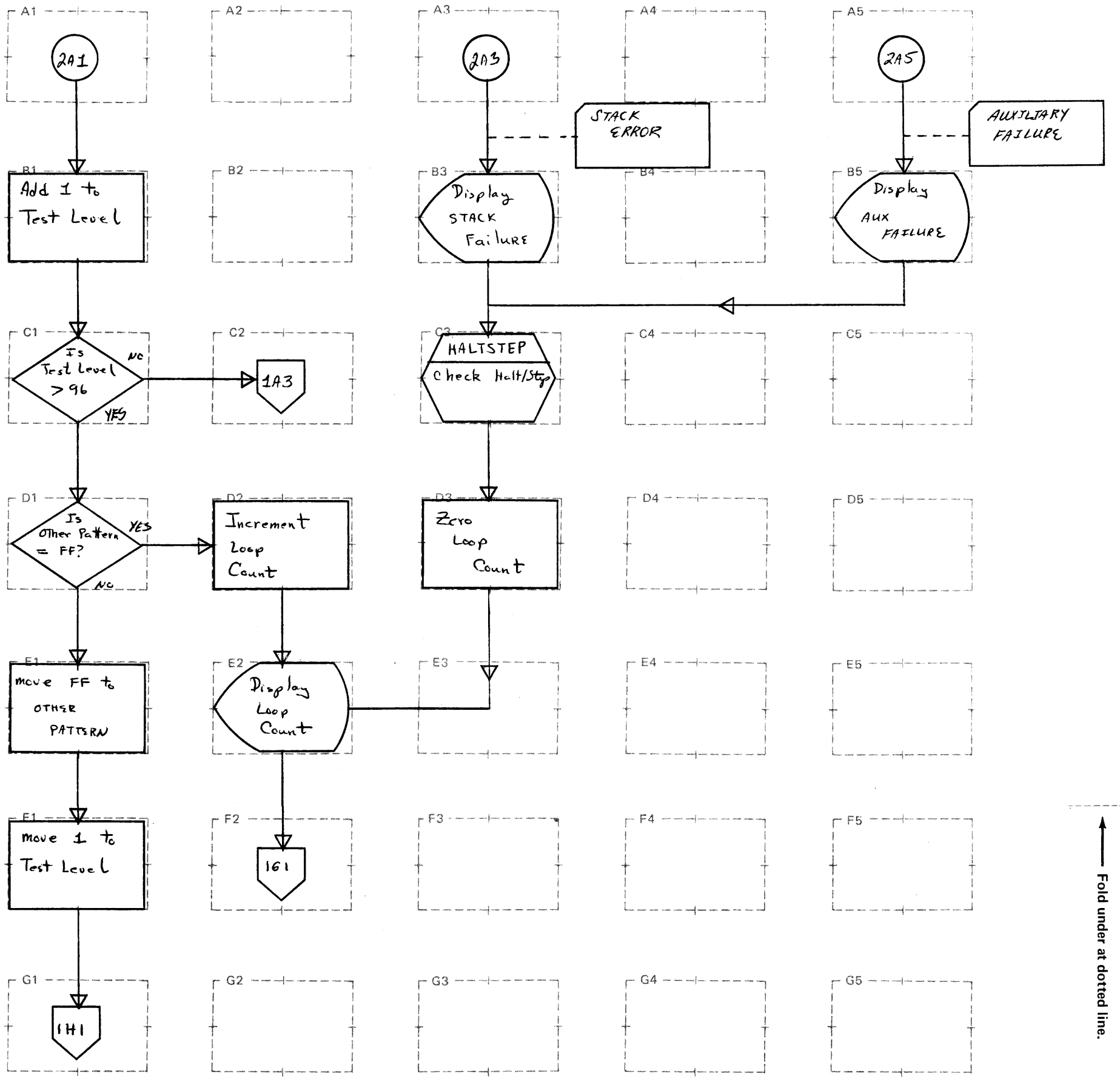
Fold under at dotted line.

WANG LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM	PROGRAM STACK (Aux) TEST	CHART	
VERSION	PREPARED BY J. Sevigny DATE 12/75	SHEET 2	OF



Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

```
3    3    *
4    4    *     MOVI8
5    5    *          This diagnostic is coded from an article
6    6    *     in the May 1976 issue of COMPUTER DESIGN entitled
7    7    *     'MOVING INVERSIONS TEST PATTERN IS THOROUGH, YET
8    8    *     SPEEDY' by J. Henk de Jonge and Andre J. Smulders.
9    9    *
10   10   *     Description
11   11   *          "In principle, the MOVI program inverts the
12   12   *     data of each address sequentially,, thus creating
13   13   *     an access time by the jump from one address to
14   14   *     another which contains different information."
15   15   *
```

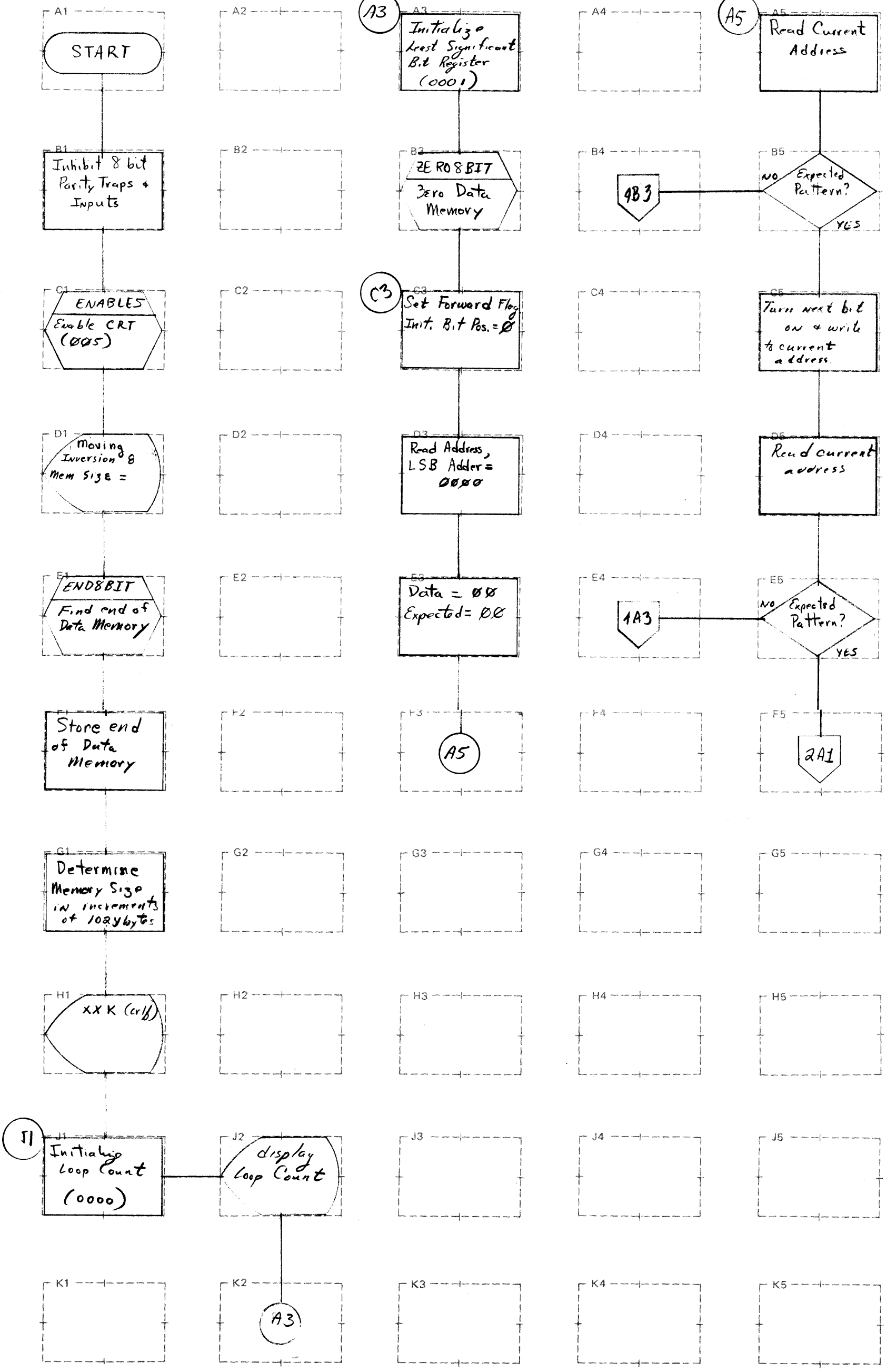
```
17    17   *    MODIFIED REGISTERS:
18    18   *         AUX 00 - 03, SL,F7-F0
19    19   *
20    20   *         AUX 00 - END OF MEMORY
21    21   *         AUX 01 - LOOP COUNT
22    22   *         AUX 02 - CURRENT READ ADDRESS
23    23   *         AUX 03 - LEAST SIGNIFICANT BIT ADDRESS
24    24   *         AUX 04 - CURRENT OFFSET ADDRESS OF LSB ADDR
25    25   *
26    26   *              SL - FORWARD/REVERSE SEQUENCE
27    27   *              F7 - BIT POSITION
28    28   *              F6 - EXPECTED DATA
29    29   *              F5 - DATA FLAG
30    30   *              F0 - WORK
31    31   *              REMAINING FILE REGISTERS MAY BE USED
32    32   *               BY VARIOUS BOOTSTRAP SUBROUTINES.
33    33   *
```

```
35   35   *
36   36   *        MEMORY BORAD LAYOUT
37   37   *
38   38   *                  EVEN                    ODD
39   39   *          !---------------!---------------!
40   40   *  4000    !01          80 P 01          80 P!  5FFF
41   41   *          !---------------!---------------!
42   42   *  6000    !01          80 P 01          80 P!  7FFF
43   43   *          !---------------!---------------!
44   44   *  0000    !01          80 P 01          80 P!  1FFF
45   45   *          !---------------!---------------!
46   46   *  2000    !01          80 P 01          80 P!  3FFF
47   47   *          !---------------!---------------!
```

```
49   *
50   *       SCREEN DISPLAY AND ERROR MESSAGES
51   *
52   *            MOVING INVERSION 8 - MEM SIZE = XXK
53   *            # LLLL
54   *
55   *                  WHERE XX - MEMORY SIZE IN K UNITS
56   *                     LLLL - LOOP COUNT
57   *
58   *       ERROR DISCUSSION
59   *
60   *            1. ERROR PC'S = XXXX (EE/RR) XOR = YY
61   *
62   *                  WHERE: XXXX = FAILING ADDRESS
63   *                         EE = EXPECTED
64   *                         RR = ACTUAL
65   *                         YY = EXCLUSIVE OR
66   *
67   *                  AN ERROR OCCURS WHEN THE CURRENT READ
68   *               LOCATION FAILS TO MAINTAIN THE EXPECTED
69   *               PATTERN.
70   *
71                  SYMBL     JSMDD02$
7-   *

7                   B         *
75                  B         START
76                  B         *
77                  B         START
78                  B         *
79                  B         *
80                  B         *
81                  B         *
82                  B         *
83                  B         *
84                  B         *
85                  B         *
86                  B         *
87                  B         *
88                  B         *
89                  B         *
90   *
91   *       PROGRAM EQUATES
92   *
93   EOMREG    EQU       00          END OF MEMORY REGISTER
94   LOOPREG   EQU       01          LOOP COUNTER
95   ADDRESS   EQU       02          ADDRESS REGISTER
96   LSBADREG  EQU       03          LEAST SIGNIFICANT BIT ADDRESS
97   LSBADDER  EQU       04          LSB ADDRESS OFFSET
E
5          START OF DIAGNOSTIC
100  *
101  START     ORI       80,SH,SH    INHIBIT 8 BIT PARITY TRAPS
102            ANDI      OFD,SH,SH   INHIBIT INPUTS
103  *
```

SYSTEM 2600 Memory Diagnostic PROGRAM Moving Inversion 8
VERSION _____ PREPARED BY _____ DATE 6-1-76
CHART _____ SHEET 1 OF 4



Fold under at dotted line.

Fold under at dotted line.

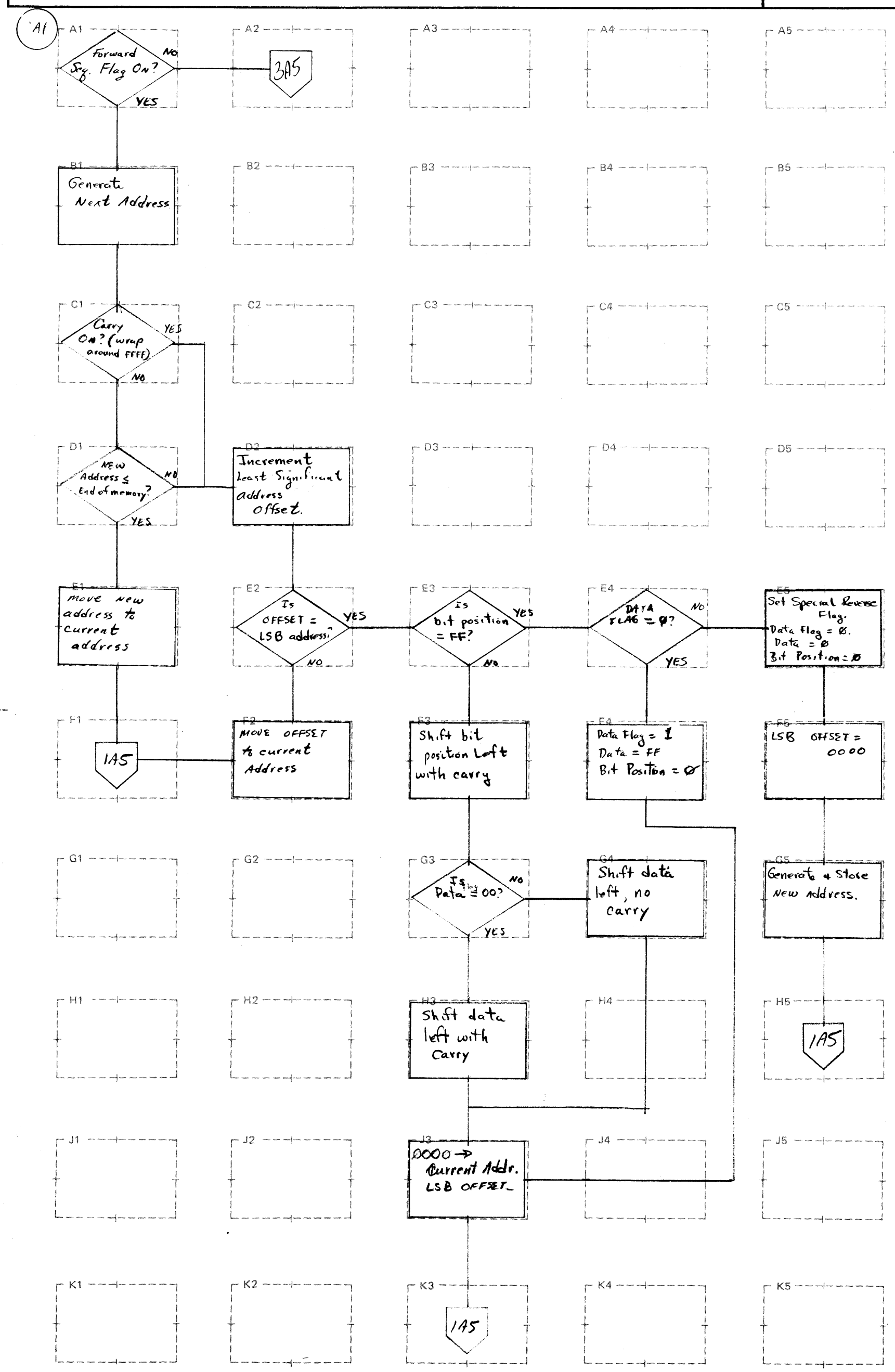
WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM 2600 Memory Diagnostic PROGRAM Moving Inversion 8

VERSION _____ PREPARED BY _____ DATE 6-1-76

CHART _____ SHEET 2 OF 4



Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.

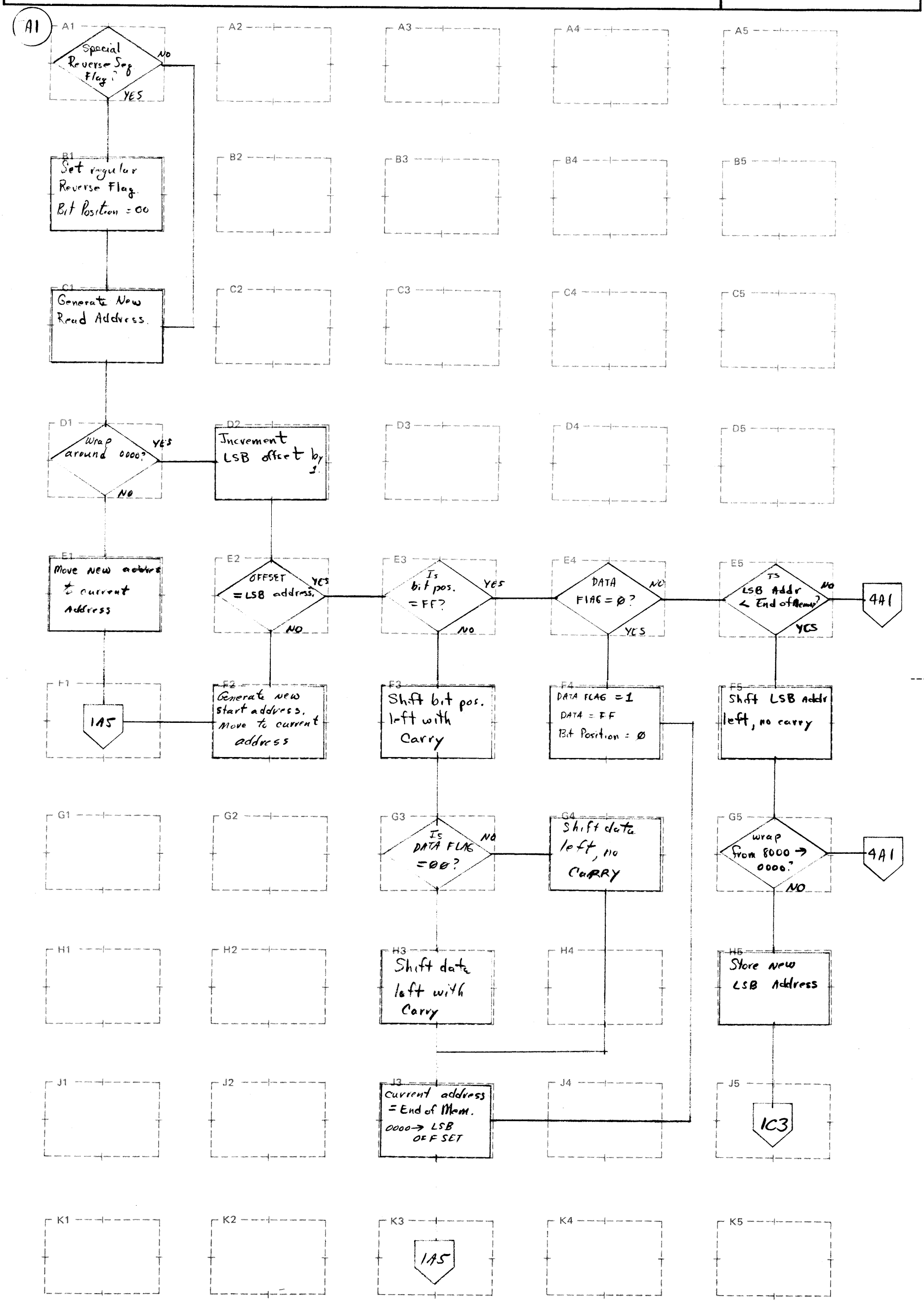
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM 2600 Memory Diagnostic PROGRAM Moving Inversion 8

VERSION _____ PREPARED BY _____ DATE 6-1-76

CHART _____ SHEET 3 OF 4



Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343 6769, TELEX 94 7421

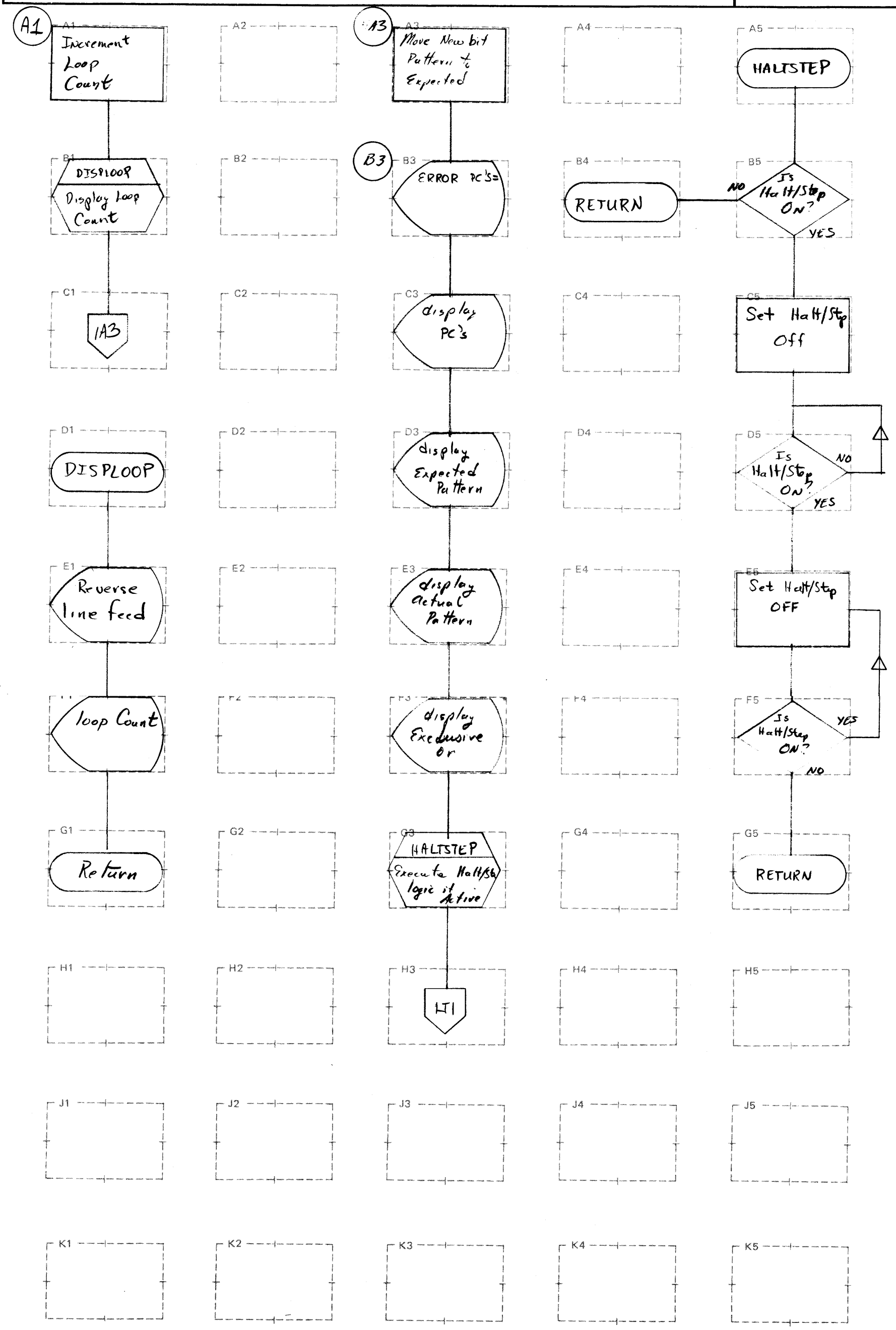
Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM 2200 Memory Diagnostic PROGRAM Moving Inversion 8

VERSION _____ PREPARED BY _____ DATE 6-1-76

CHART _____

SHEET 4 OF 4



Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM Memory Manipulation PROGRAM CHANGE DATA MEMORY	CHART
VERSION PREPARED BY J Sevigny DATE 12-75	SHEET OF

A1 A2 A3 A4 A5

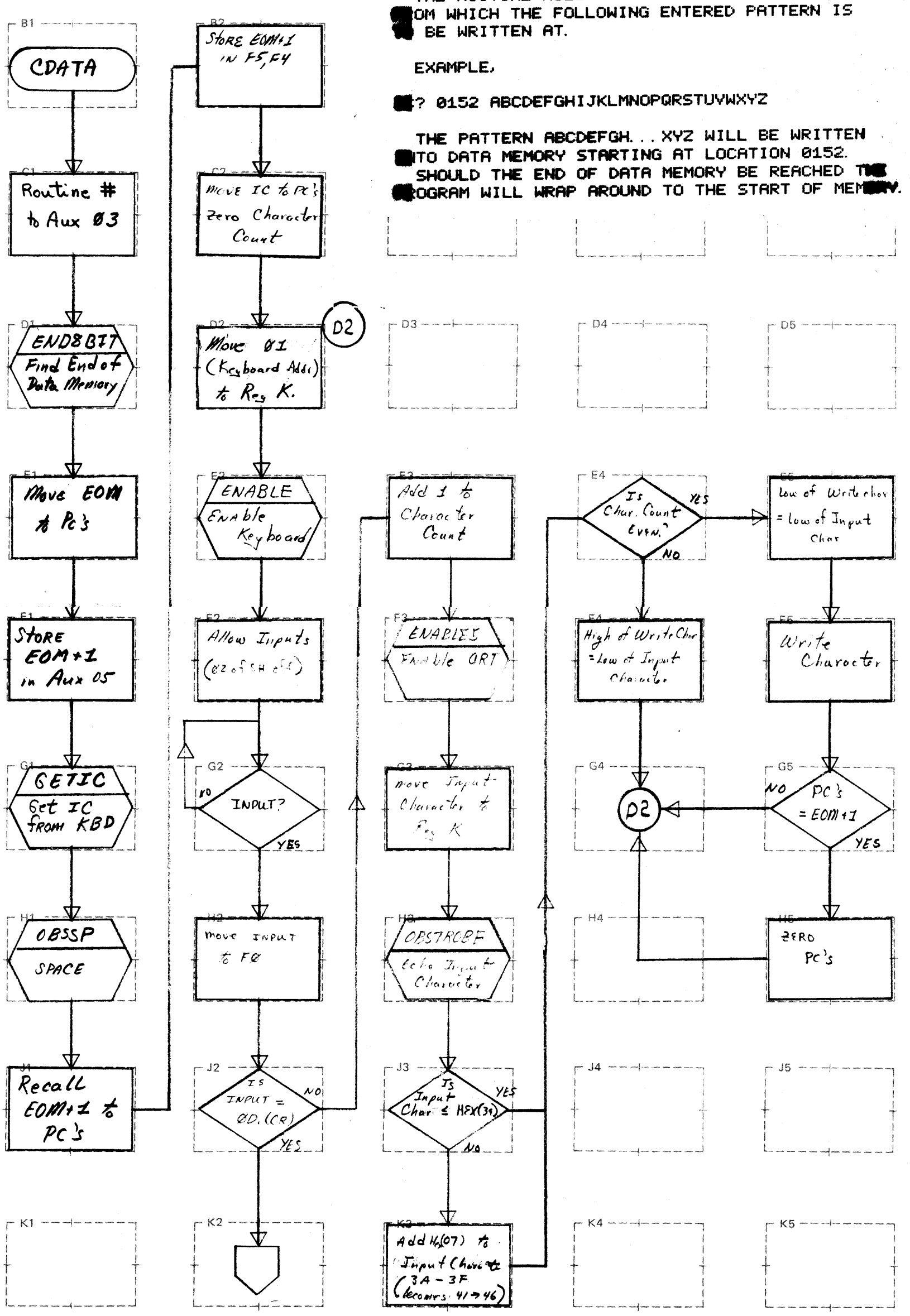
CHANGE DATA MEMORY

THIS ROUTINE WILL ALLOW THE OPERATOR TO
ITE ANY PATTERN ANYWHERE IN DATA MEMORY.
THE ROUTINE WILL REQUEST THE IC LOCATION
OM WHICH THE FOLLOWING ENTERED PATTERN IS
BE WRITTEN AT.

EXAMPLE,

? 0152 ABCDEFGHIJKLMNOPQRSTUVWXYZ

THE PATTERN ABCDEFGH. . . XYZ WILL BE WRITTEN
TO DATA MEMORY STARTING AT LOCATION 0152.
SHOULD THE END OF DATA MEMORY BE REACHED T
OGRAM WILL WRAP AROUND TO THE START OF MEM Y.

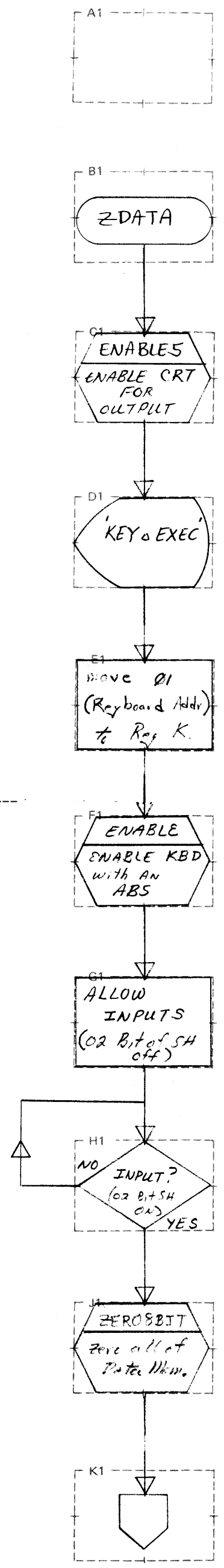


Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM Memory Manipulation PROGRAM Zero Data Memory	CHART
VERSION PREPARED BY J. Sevigny DATE 12-75	SHEET OF



[redacted]O DATA MEMORY

[redacted]HIS ROUTINE ZEROES ALL OF DATA MEMORY BY [redacted]TING HEX(00) AT EACH MEMORY LOCATION IN [redacted]A MEMORY.
[redacted]S A SAFEGUARD AGAINST ACCIDENTAL KEYING OF [redacted]S ROUTINE, THE ROUTINE WILL REQUEST THE [redacted]ERATOR TO KEY EXEC PRIOR TO ZEROING DATA [redacted]MORY.

[redacted]AMPLE,

[redacted]Y EXEC

'THE ROUTINE USES THE ZERO 8 BIT ROUTINE [redacted]CATED IN BOOTSTRAP.

Fold under at dotted line.

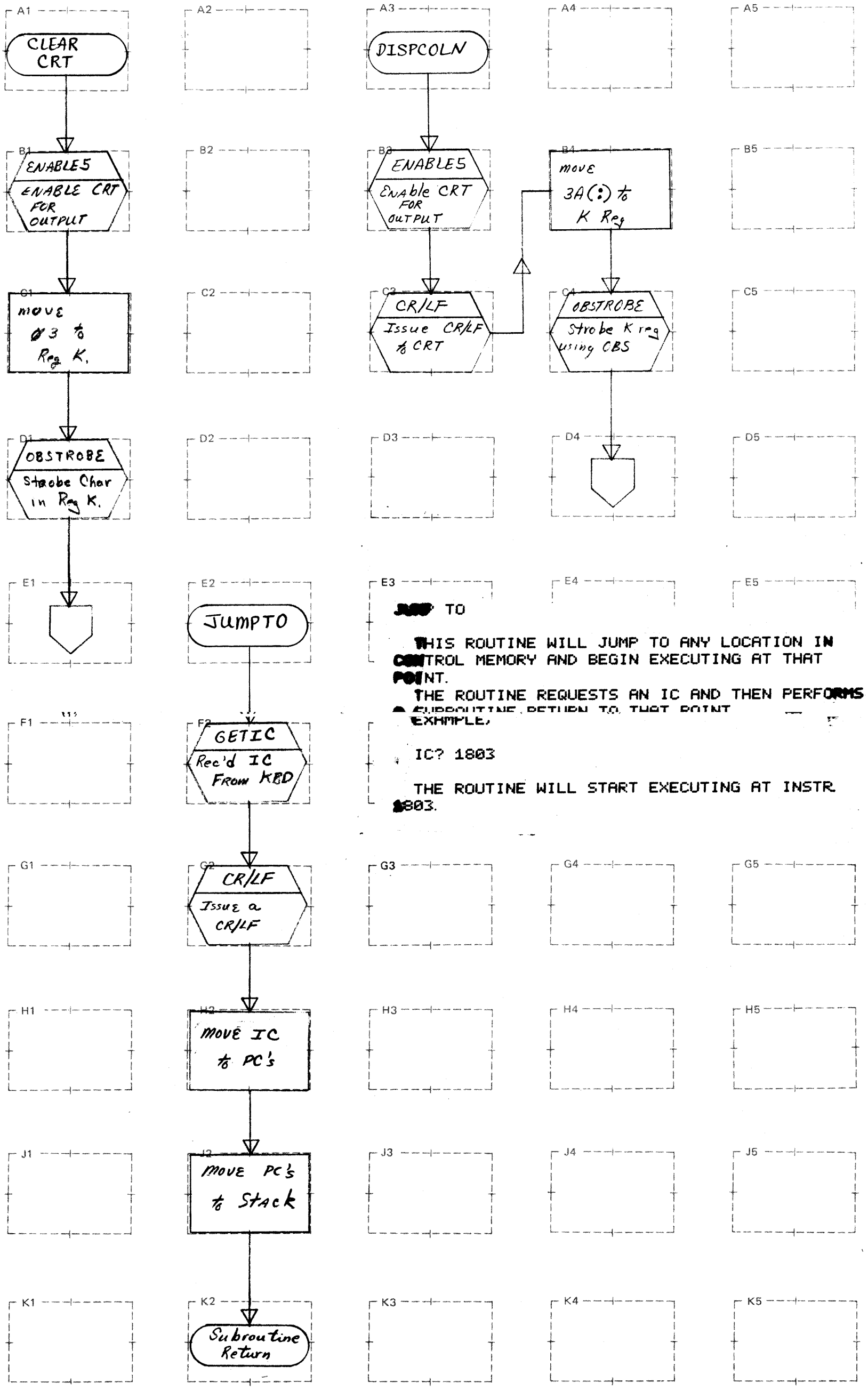
Fold under at dotted line.

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50

SYSTEM Memory Manipulation PROGRAM Jump To CHART

VERSION 12/25 PREPARED BY J Sevigny DATE 12-5-75 SHEET OF



JUMP TO

THIS ROUTINE WILL JUMP TO ANY LOCATION IN CONTROL MEMORY AND BEGIN EXECUTING AT THAT POINT.

THE ROUTINE REQUESTS AN IC AND THEN PERFORMS A SUBROUTINE RETURN TO THAT POINT.

EXAMPLE:

IC? 1803

THE ROUTINE WILL START EXECUTING AT INSTR 1803.

Fold under at dotted line.

Fold under at dotted line.

WANG LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3186A
1-74-5C
Price $2.50