Federal Communications Commission Radio Frequency Interference Statement

WARNING: The equipment decribed herein has been certified to comply with the limites for a Class B computing device, persuant to Subpart J of Part 15 of FCC rules. Operation with non-certified peripherals is likely to result in interference to radio and TV reception. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limites may be attached to the computer.

INSTRUCTION TO USER

This equipment generates and uses radio frequency energy and if not installed and used properly, i.e., in strict accordance with the operating instructions, reference manuals, and the service manual, may cause interference to radio or to television reception.

If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

1. Reorient the receiving antenna.
2. Relocate the equipment with respect to the receiver.
3. Move the equipment away from the receiver.
4. Plug the equipment into a different outlet so that equipment and receiver are on different branch circuits.
5. Ensure that card mounting screws, attachment connector screws, and ground wires are tightly secured.
6. Consult dealer service representative for additional suggestions.
7. Use of un-shielded cables with this device will cause radio frequency interference. Therefore shielded cables must be used.
8. Additional information concerning radio frequency interference may be obtained from U.S. Government Printing Office. Washington, D.C. 20402

Stock #004-000-00345-4 this booklet is called: "How to Identify and Resolve Radio – T.V. Interference Problems."

The manufacturer is not responsible for any radio or TV interference caused by unauthorized modifications to this equipment. It is the responsiblity of the user to correct such interference.

CONTENT

1. READ ME BEFORE YOU USE THIS DEVICE 1
2. INSTALLATION 2
3. INTRODUCTION OF IESD 4
4. START THE DEVICE 5
5. IC TESTER 7
   - Search IC 9
   - Test IC Function OK or Bad 11
   - Loop Test 15
6. DRAM TESTER 19
7. SRAM TESTER 23
8. EPROM WRITER 27
   - Edit 39
   - Copy 53
   - File 59
9. APPENDIX 61

1. READ ME BEFORE YOU USE THIS DEVICE

1. Check your IESD diskette, and make a copy of it. After you make sure that the copy is complete, then keep one diskette for reservation.

2. Use MS-DOS Version 2.0 or version higher.

3. The memory of the main board must be 128K at least.

4. Refer to the appendix to make sure your IC can be tested by this program. If you got any interruption during test, refer to the appendix first too.

5. There is a file "readme doc" in the diskette attached, refer to it by key in "type readme" under MS-DOS. Any newest information of IESD card will be added on this file.

6. Quick Pulse:
   New IESD adds a file, named "YLQ", for a much high speed writing in EPROM WRITER.
   If you want to write EPROMs in quicker speed, enter file "YLQ" than "YL" when you start this device.
   Although it is much quicker in writing EPROM use file "YLQ", the way "quick pulse", but this writing way can not promise a perfect writing work.
   And it is not every brands' EPROMs can be written by this way.

2. INSTALLATION

1. First, insert test card to the slot of the PC/XT or PC/AT mother board, shown as fig. 1

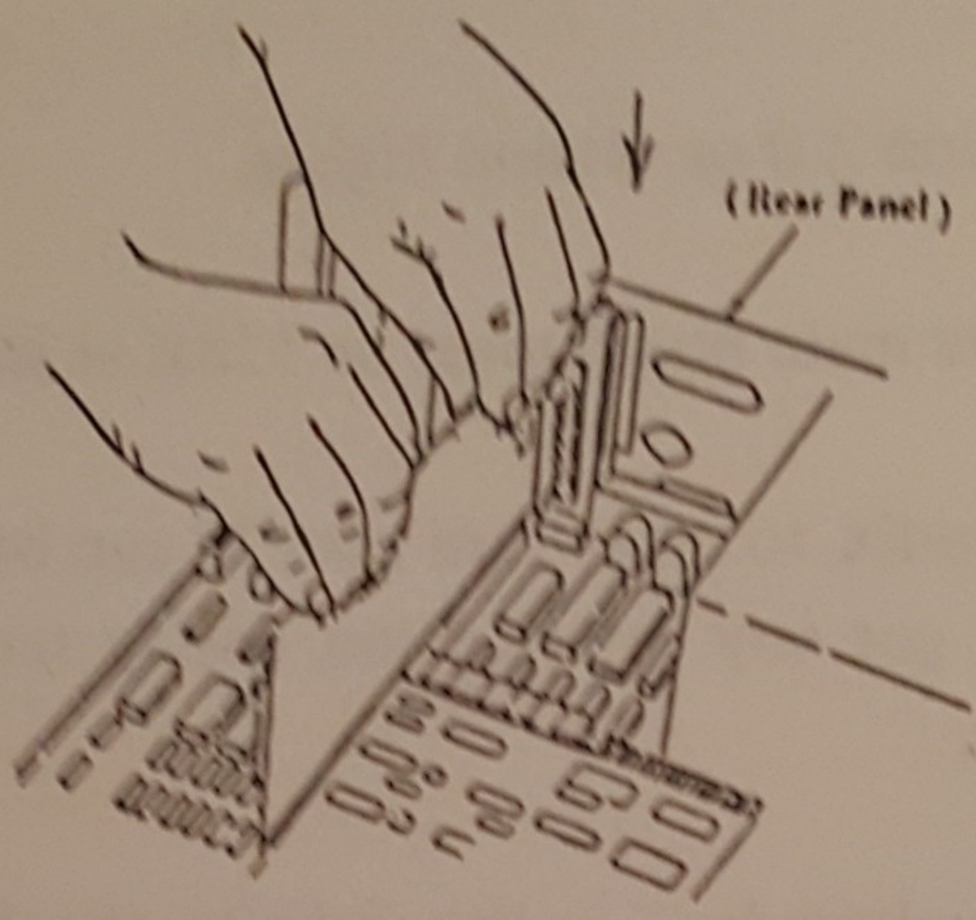


Fig. 1

2. Insert cable to test card and testbox.
   CAUTION: The direction of cable red line must be correct. Be sure to connect the cable with the test card and test box the way the two white spots at the same side. It's shown as fig. 2.

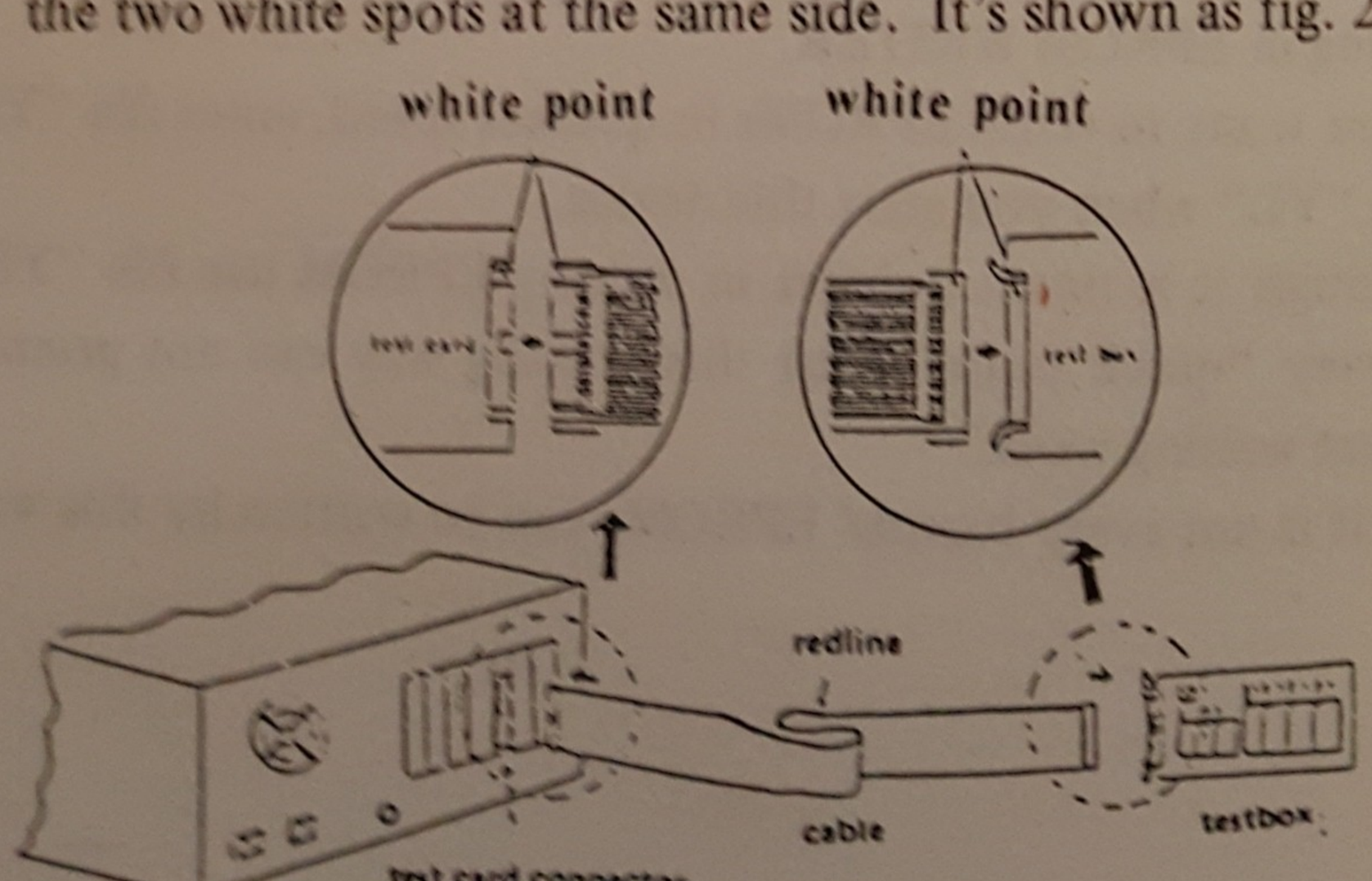


Fig. 2

3. Put the DOS diskette in drive A, I.E.S.D test diskette in drive B. Be sure that the procedure mentioned above is proceeded accordingly, then turn the power on.
4. When enterred the DOS, then type "b:YL" now we are going to I.E.S.D test mode.
5. There are seven textools on the textbox (as figure 3.), one for IC tester, two for DRAM tester, four for EPROM writer and SRAM tester.

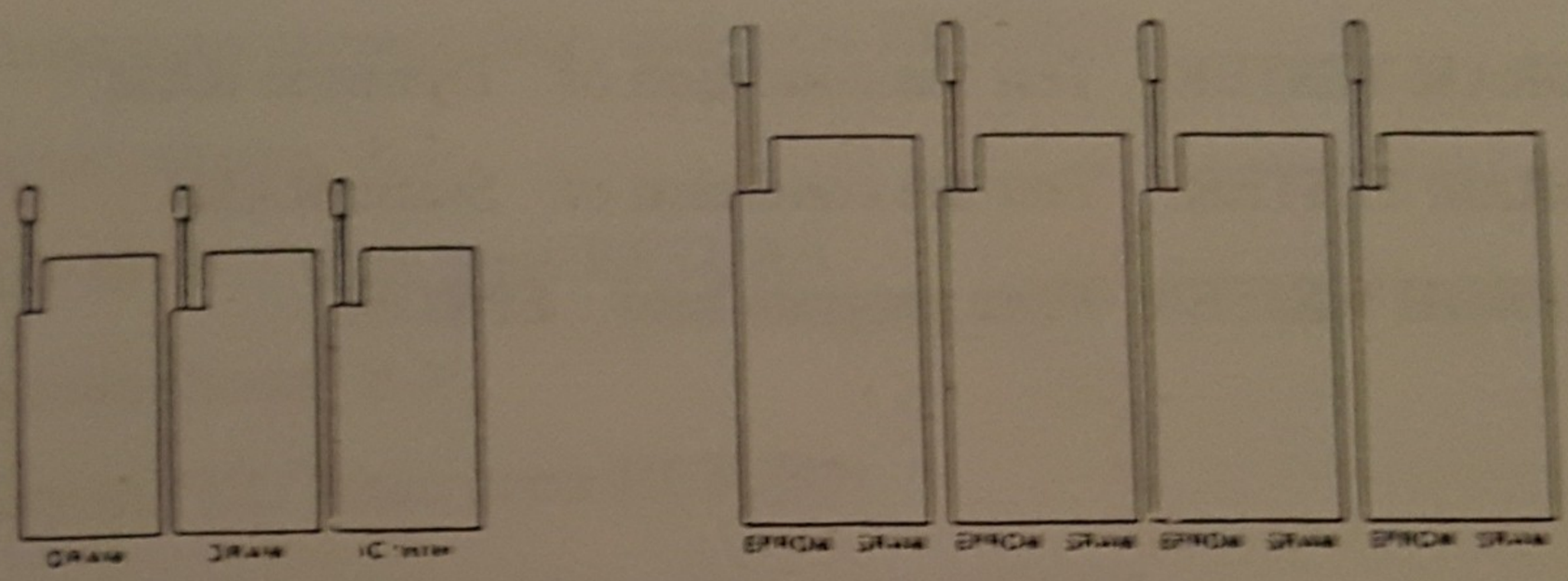


Fig. 3

CAUTION: You must put the tested IC or EPROM on the correct position (textool) as marked on the testboxes, with the recess of IC up, place the tested ICs (TTL, CMOS, SRAM, EPROM) into the testboxes, by the way "bottom to bottom" (14–28 pins) as fig. 4.

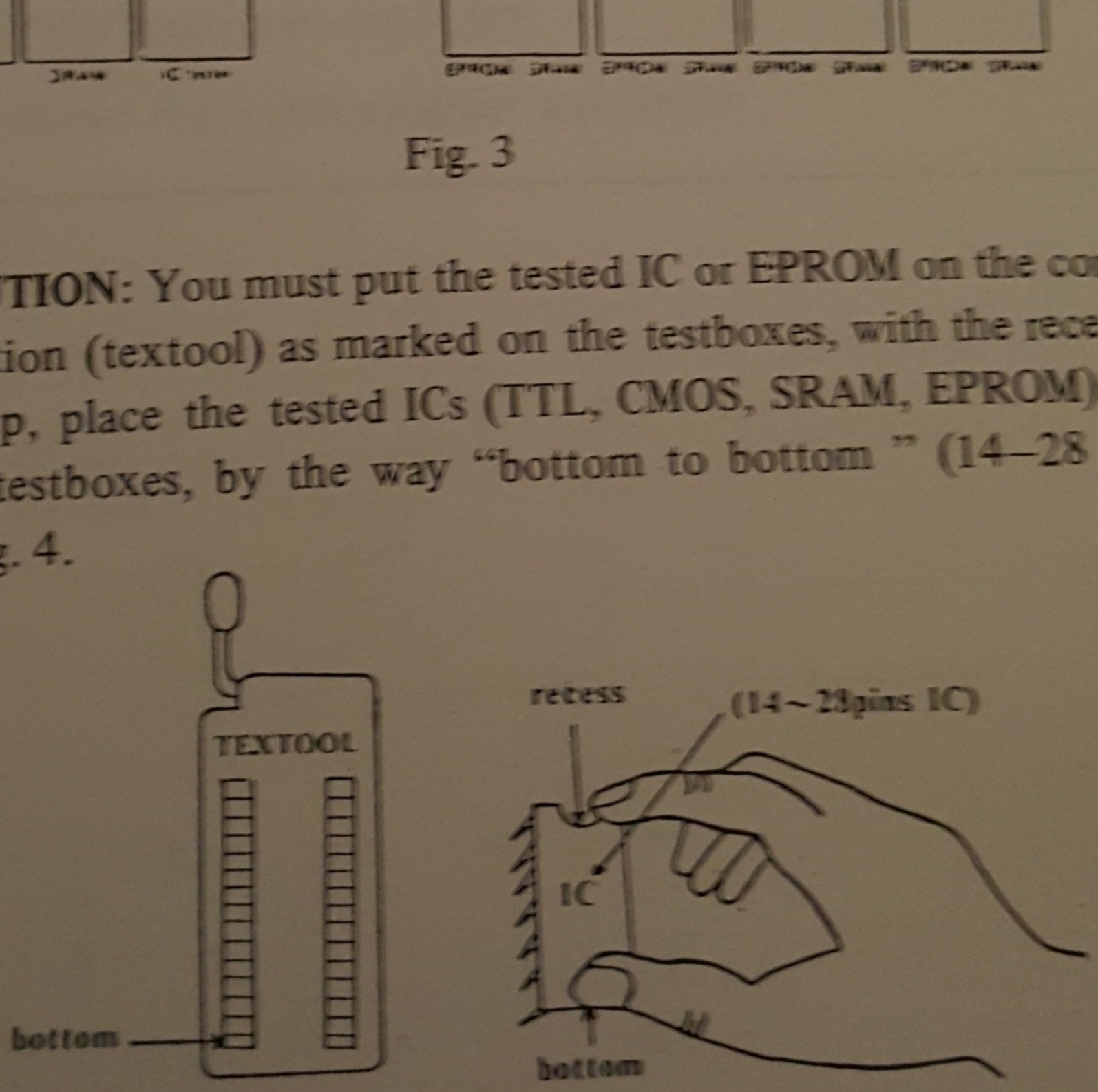


Fig. 4

MS-DOS is trademark of Microsoft Corporation.

3. INTRODUCTION OF IESD

COPY RIGHT 1986 by YIH LUNG ENTERPRISE CO., LTD.

The YIH LUNG IC TESTER & EPROM WRITER & SRAM TESTER & DRAM TESTER is a interface card with four function:

1. IC TESTER: Test the condition of TTL, CMOS
2. DRAM TESTER: Test the condition of Dynamic RAM
3. SRAM TESTER: Test the condition of Static RAM
4. EPROM WRITER: Write program into EPROM

4. START THE DEVICE

1. Put MS-DOS diskette version 2.0 (or higher) into drive A, Put IESD diskette into drive B.
2. Turn monitor and computer on.
3. Enter your current date, and press RETURN.
4. Enter your current time, and press RETURN.
5. Screen shows: A >
6. Key in "b:" then press RETURN.
7. Screen shows: B >
8. Key in "YL" then press RETURN.
9. Screen shows Fig. 1, the program was started.

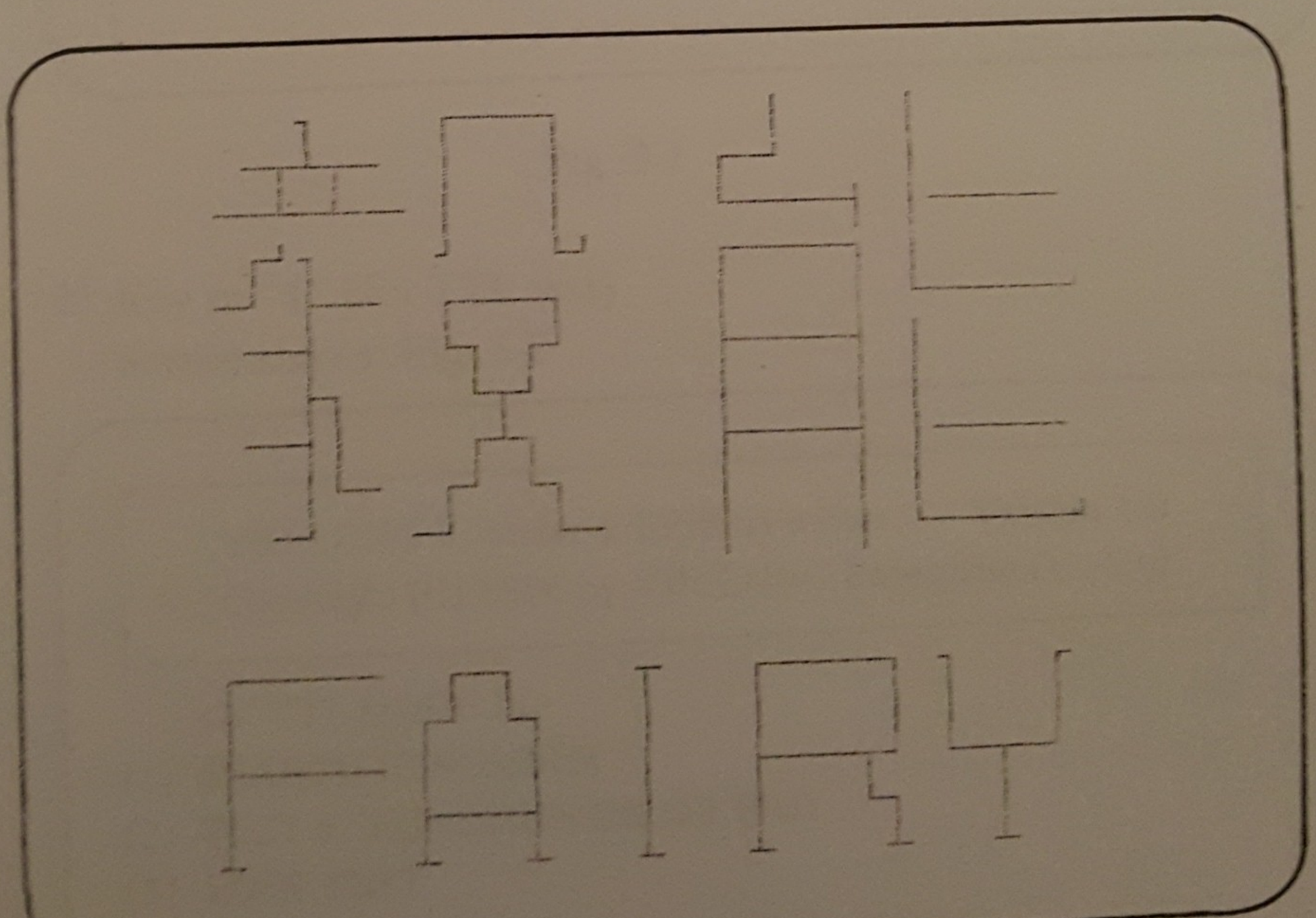


FIG. 1

10. Press SPACE key to enter Fig. 2.

```
YIH-LUNG IC tester & EPROM writer      Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

(1) IC-TESTER   (2) DRAM TESTER   (3) SRAM TESTER
(4) EPROM WRITER

Select a number to continue . . .

Anytime press <ESC>      to exit to system,
              <Pg Up>    to previous screen
```

Fig. 2

5. IC TESTER·

1. Start IESD program, enter Fig. 2
2. Key in "1".
3. Screen shows: Fig. 5-1

```
        YIH-LUNG IC tester & EPROM writer       Version 1.0
        Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

(1) TTL    (2) CMOS

        Select a number to continue . . .

        Anytime press <ESC>      to exit to system,
                      <Pg Up>    to previous screen
```

Fig. 5-1

4. If: Key in "1" (for TTL test)
   Screen shows: Fig. 5-2

```
        YIH-LUNG IC tester & EPROM writer       Version 1.0
        Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

        << TTL MODE >>
(1) SEARCH IC NUMBER
(2) TEST IC FUNCTION OK OR BAD
(3) LOOP TEST

        Select a number to continue . . .
```

Anytime press <ESC> to exit to system,
<Pg Up> to previous screen

Fig. 5-2

If: Key in "2" (for CMOS test)
Screen shows: Fig. 5-2-1

YIH-LUNG IC tester & EPROM writer Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

<< CMOS MODE >>
(1) SEARCH IC NUMBER
(2) TEST IC FUNCTION OK OR BAD
(3) LOOP TEST

Select a number to continute . . .

Anytime press <ESC> to exit to system,
<Pg Up> to previous screen

Fig. 5-2-1

Use TTL test as sample —— the test procedure of both TTL and CMOS are the same as following

5. Screen shows: Fig..5-2
You got three choice:
1. Search IC number
2. Test IC function OK or bad
3. Loop test

Search IC

1. Screen on Fig. 5-2.
2. Key in "1"
3. Screen shows: Fig. 5-3

```
YIH-LUNG IC tester & EPROM writer          Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

<< TTL MODE >>
SEARCH IC NUMBER

Put IC on the socket, then press < CR > to start search . . .

Anytime press <ESC>      to exit to system,
              <Pg Up>    to previous screen
```

Fig. 5-3

4. Put IC on the socket, lock it, press RETURN
5. If: Screen shows: Fig. 5-4

```
YIH-LUNG IC tester & EPROM weriter         Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

<< TTL MODE >>
SEARCH IC NUMBER

Put IC on the socket, then press < CR > to start search . . .
The IC may be . . .    32

Press < CR > to continue . . .

Anytime press <ESC>      to exit to system,
              <Pg Up>    to previous screen
```

Fig. 5-4

"The IC may be . . . xx"
xx is the IC number

If: Screen shows: Fig. 5-5
"Cannot recognize the IC's number"

This means: 1. The IC is failed
or 2. The IC is not in the test range
(Check on Appendix 1)

6. Press "Pg Up" key to previous screen

```
YIH-LUNG IC tester & EPROM writer      Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

<< TTL MODE >>
SEARCH IC NUMBER

Put IC on the socket, then press < CR > to start search . . .
Cannot recognize the IC's number

Press < CR > to continue . . .

Anytime press <ESC>     to exit to system,
              <Pg Up>   to previous screen
```

Fig. 5-5

TEST IC FUNCTION OK OR BAD

1. Screen on Fig. 5-2
2. Key in "2"
3. Screen shows: Fig. 5-6

Note: This test was designed on LS base, only one type of IC assignments can be tested.

YIH-LUNG IC tester & EPROM writer Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

TEST IC FUNCTION OK OR BAD

****

74/54

****

Note:
Some ICs (e.g. 74LS01, 74H01) have same number, but different pin assignments, only one type of pin assignment can be tested. Design of LS base.

Anytime press <ESC> to exit to system,
<Pg Up> to previous screen

Fig. 5-6

4. Key in the number of tested IC, press RETURN
5. Screen shows: Fig. 5-7.

YIH-LUNG IC tester & EPROM writer Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

TEST IC FUNCTION OK OR BAD

****

74/5432

****

Note:
Some ICs (e.g. 74LS01, 74H01) have same number, but different pin assignments, only one type of pin assignment can be tested. Design of LS base.
Put IC on the socket, then press < CR > to start test . . .

Anytime press <ESC> to exit to system,
<Pg Up> to previous screen

Fig. 5-7

6. Put IC on the socket, press RETURN
7. If: Screen shows: Fig. 5-8
   "RESULT . . . OK"
   This means: The IC is in good condition

YIH-LUNG IC tester & EPROM writer Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

TEST IC FUNCTION OK OR BAD

```
**************************************************

                    74/5432

**************************************************

Note:
Some ICs (e.g. 74LS01, 74H01) have same number, but
different pin assignments, only one type of pin assignment can
be tested. Design of LS base.
Put IC on the socket, then press < CR > to start test . . .

RESULT . . . OK
Press < CR > to continue . . .

   Anytime press <ESC>      to exit to system,
                 <Pg Up>    to previous screen
```

Fig. 5-8

If: Screen shows: Fig. 5-9

"RESULT . . . FUNCTION ERROR"

This means: 1. The IC is failed

2. The IC is not in the test range

8. Press "Pg Up" key to previous screen

```
YIH-LUNG IC tester & EPROM writer      Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

                TEST IC FUNCTION OK OR BAD

**************************************************

                    74/5432

**************************************************
```

Note:
Some ICs (e.g. 74LS01, 74H01) have same number, but different pin assignments, only one type of pin assignment can be tested. Design of LS base.

RESULT . . . FUNCTION ERROR
Press <CR> to continue . . .

Anytime press <ESC> to exit to system,
<Pg Up> to previous screen

Fig. 5-9

LOOP TEST

1. Screen on Fig. 5-2
2. Key in "3"
3. Screen shows: Fig. 5-10.
4. Press "Y"
   (If press "N", current time will not show)

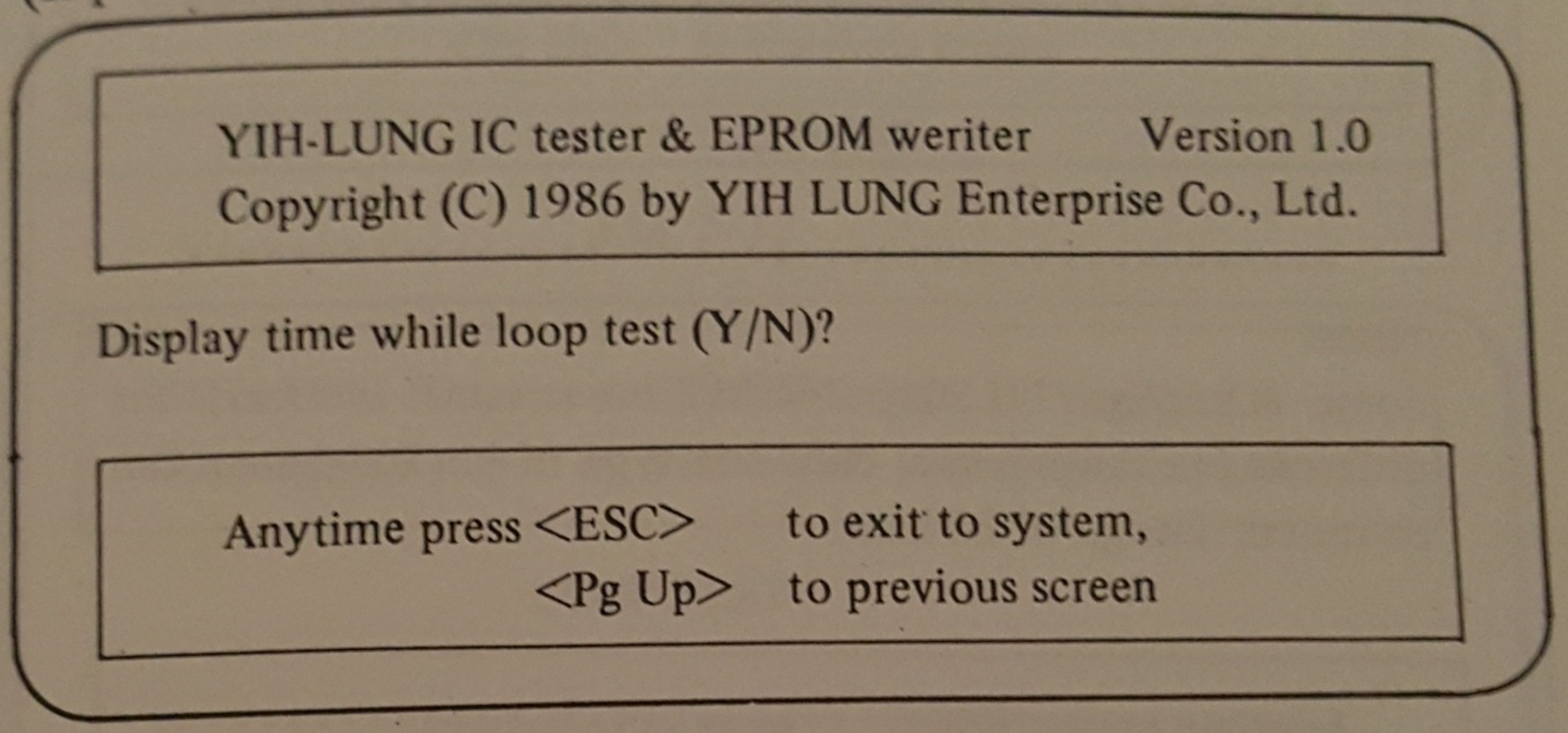


Fig. 5-10

5. Screen shows: Fig. 5-11.
6. Key in the IC number, press RETURN.
7. Screen shows: Fig. 5-12.
8. Put IC on the socket, press RETURN
9. If: Screen shows: Fig. 5-13.
   "RESULT . . . OK"
   This means:The IC is in good condition at the time
   If: Screen shows: Fig. 5-14
   "RESULT . . . FUNCTION ERROR"
   This means: The IC is failed at the time
   Press "Pg Up" key to previous screen.

YIH-LUNG IC tester & EPROM weriter Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

LOOP TEST

****

74/5432

****

Note:

Some ICs (e.g. 74LS01, 74H01) have same number, but different pin assignments, only one type of pin assignment can be tested. Design of LS base.

Anytime press <ESC> to exist to system,
<Pg Up> to previous screen

Fig. 5-11

YIH-LUNG IC tester & EPROM writer Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

LOOP TEST

****

74/5432

****

Note:
Some ICs (e.g. 74LS01, 74H01) have same number, but different pin assignments, only one type of pin assignment can be tested. Design of LS base.

Put IC on the socket; then press < CR > to start loop test, thereafter any kev to stop loop test . . .

Anytime press	<ESC>	to exit to system,
	<Pg Up>	to previous screen

Fig. 5-12

YIH-LUNG IC tester & EPROM writer Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

LOOP TEST

****

74/5432

****

Note:
Some ICs (e.g. 74LS01, 74H01) have same number, but different pin assignments, only one type of pin assignment can be tested. Design of LS base.

00:17:46

Put IC on the socket, then press < CR > to start loop test, thereafter any key to stop loop test . . .
RESULT . . . OK

Anytime press <ESC> to exit to system,
<Pg Up> to previous screen

Fig. 5-13

YIH-LUNG IC tester & EPROM writer Version 1.0
Copyright (C) 1986 by YIH LUNG Enterprise Co., Ltd.

LOOP TEST

****

74/5432

****

Note:
Some ICs (e.g. 74LS01, 74H01) have same number, but different pin assignments, only one type of pin assignment can be tested. Design of LS base.

00:20:48

Put IC on the socket, then press < CR > to start loop test, thereafter any key to stop loop test . . .
RESULT . . . FUNCTION ERROR

Anytime press <ESC> to exit to system,
<Pg Up> to previous screen

Fig. 5-14

6. DRAM TESTER

1. Screen on Fig. 2
2. Key in "2"
3. Screen shows: Fig. 6-1
4. Key in "1" (for 4164)
   Select the number of your IC
   "0" to back to Fig. 2

```
<< DRAM TESTER >>

        0.  Exit
        1.  4164
        2.  41256
        3.  4416
        4.  4464

PLEASE SELECT ONE?
```

Fig. 6-1

5. Screen shows: Fig. 6-2

```
<< DRAM TESTER >>

        0.  Exit
        1.  4164
        2.  41256
        3.  4416
        4.  4464

PLEASE SELECT ONE?  1

---------- Please put DRAM on the textool --------
```

Fig. 6-2

6. Put IC on the socket, press RETURN
7. Screen shows: Fig. 6-3, with "Memory is testing" flashing Refer to Appendix II for test speed.

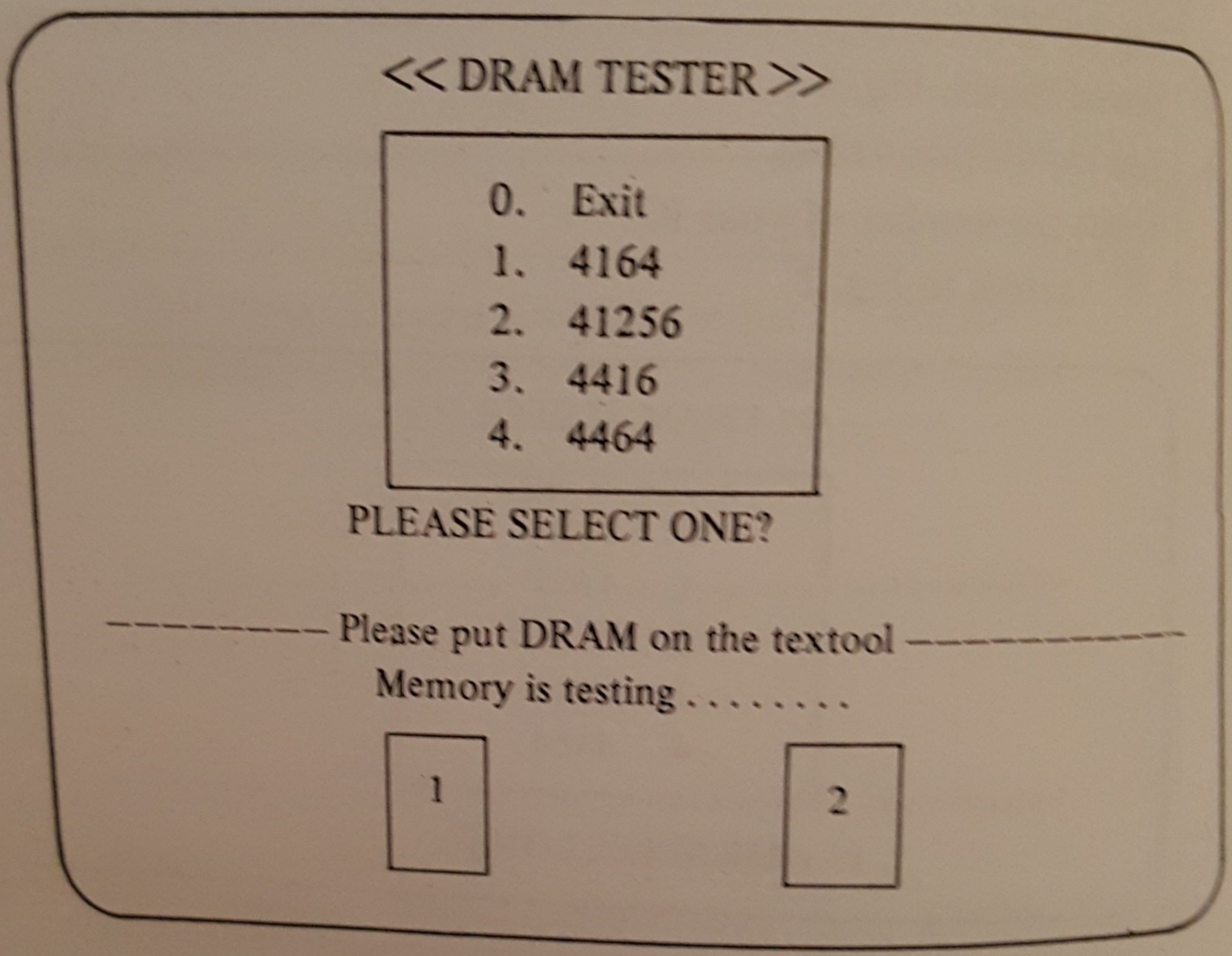


Fig. 6-3

8. If: Screen shows: Fig. 6-4
   "PASS" means the DRAM is in good condition
   If: Screen shows: Fig. 6-5
   "ERROR" means the DRAM is failed
9. Press any Key to continue, or
   Press "0" to back to Fig. 2

```
<< DRAM TESTER >>

    0.  Exit
    1.  4164
    2.  41256
    3.  4416
    4.  4464

PLEASE SELECT ONE?  1

---------- Please put DRAM on the textool ----------
Memory is testing . . . . . .

      1                2

    PASS             PASS

PRESS ANY KEY TO CONTINUE . . . .
```

Fig. 6-4

```
<< DRAM TESTER >>

    0.  Exit
    1.  4164
    2.  41256
    3.  4416
    4.  4464

PLEASE SELECT ONE?  1
```

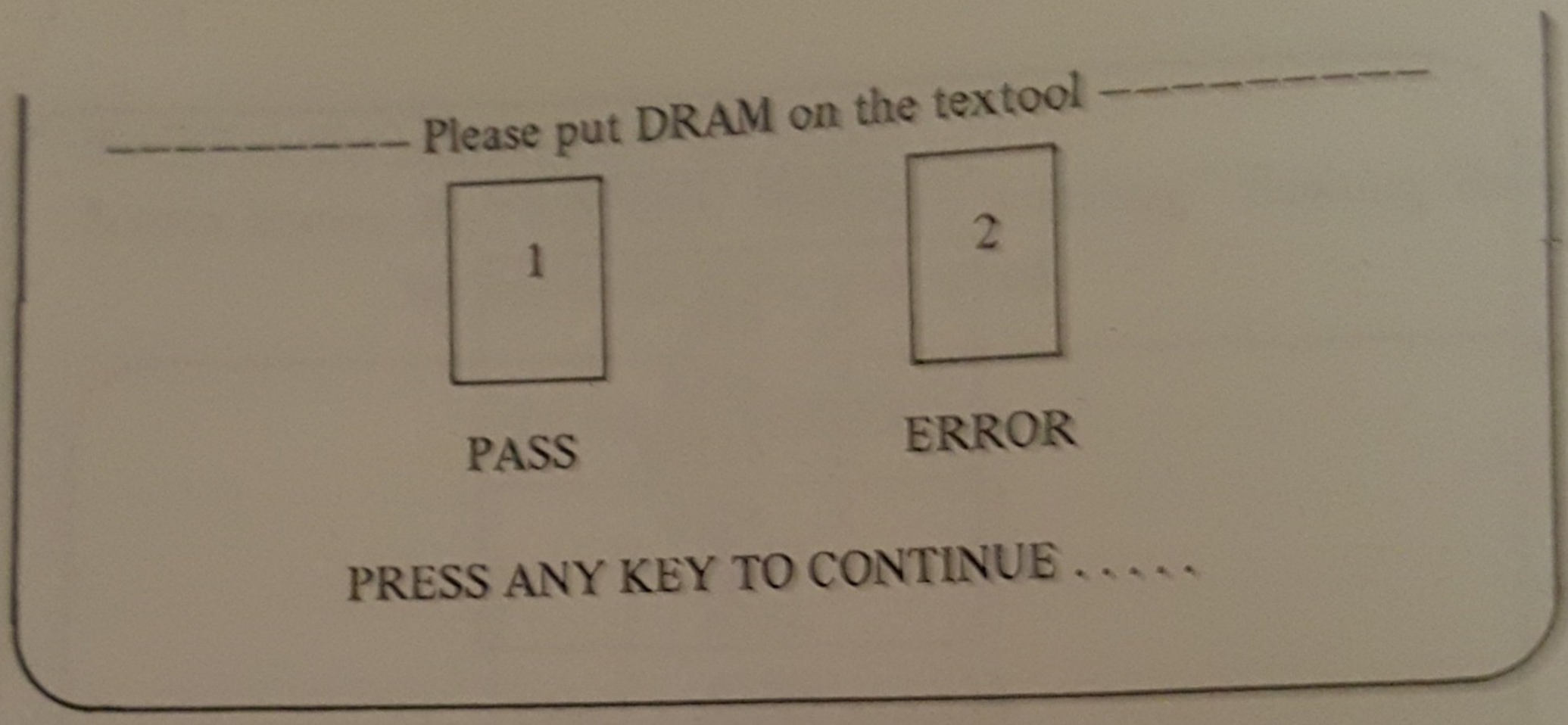

Please put DRAM on the textool
1
2
PASS
ERROR
PRESS ANY KEY TO CONTINUE . . . . .

Fig. 6-5

7. SRAM TESTER

1. Screen on Fig. 2.
2. Key in "3"
3. Screen shows: Fig. 7-1

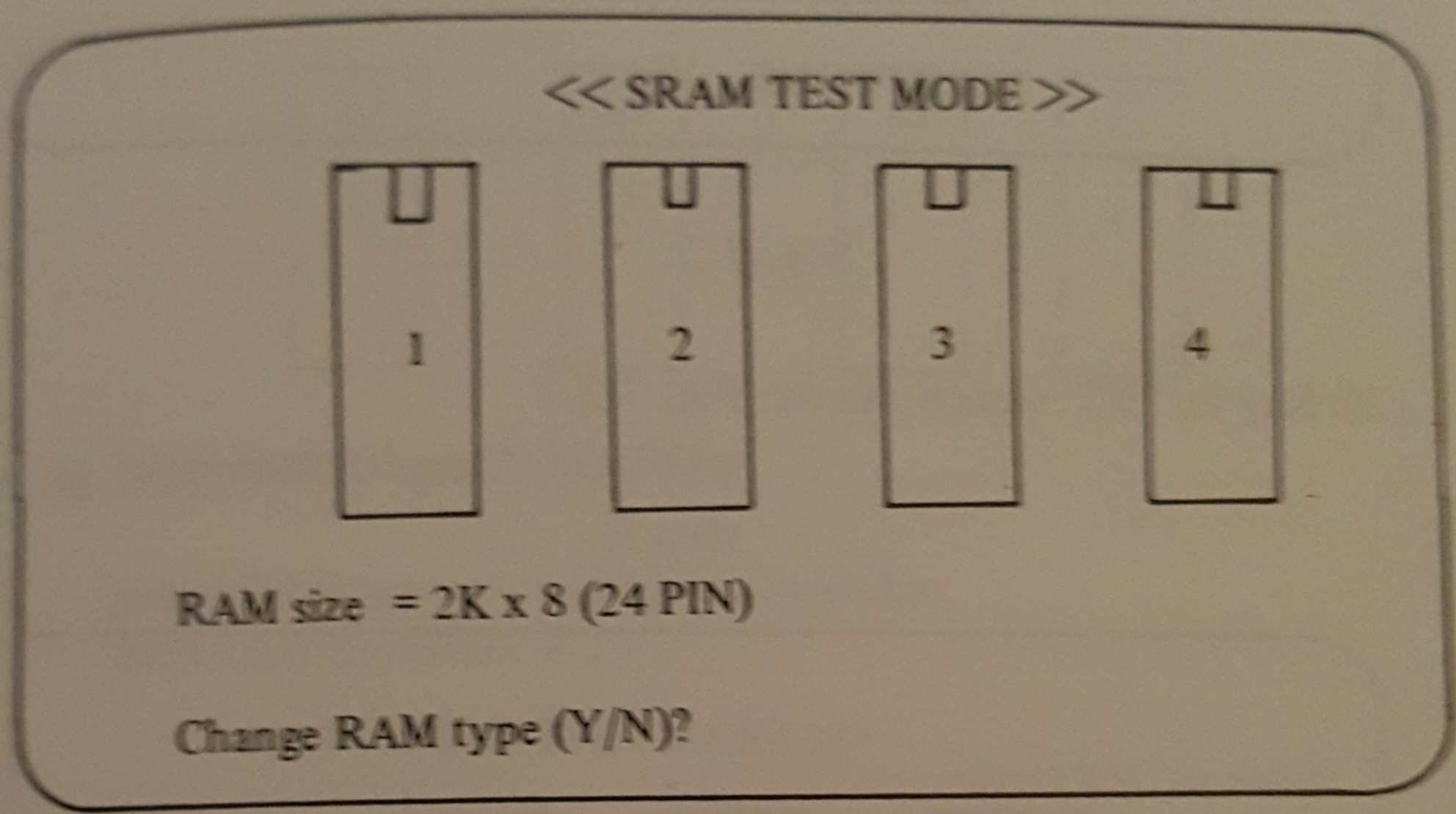
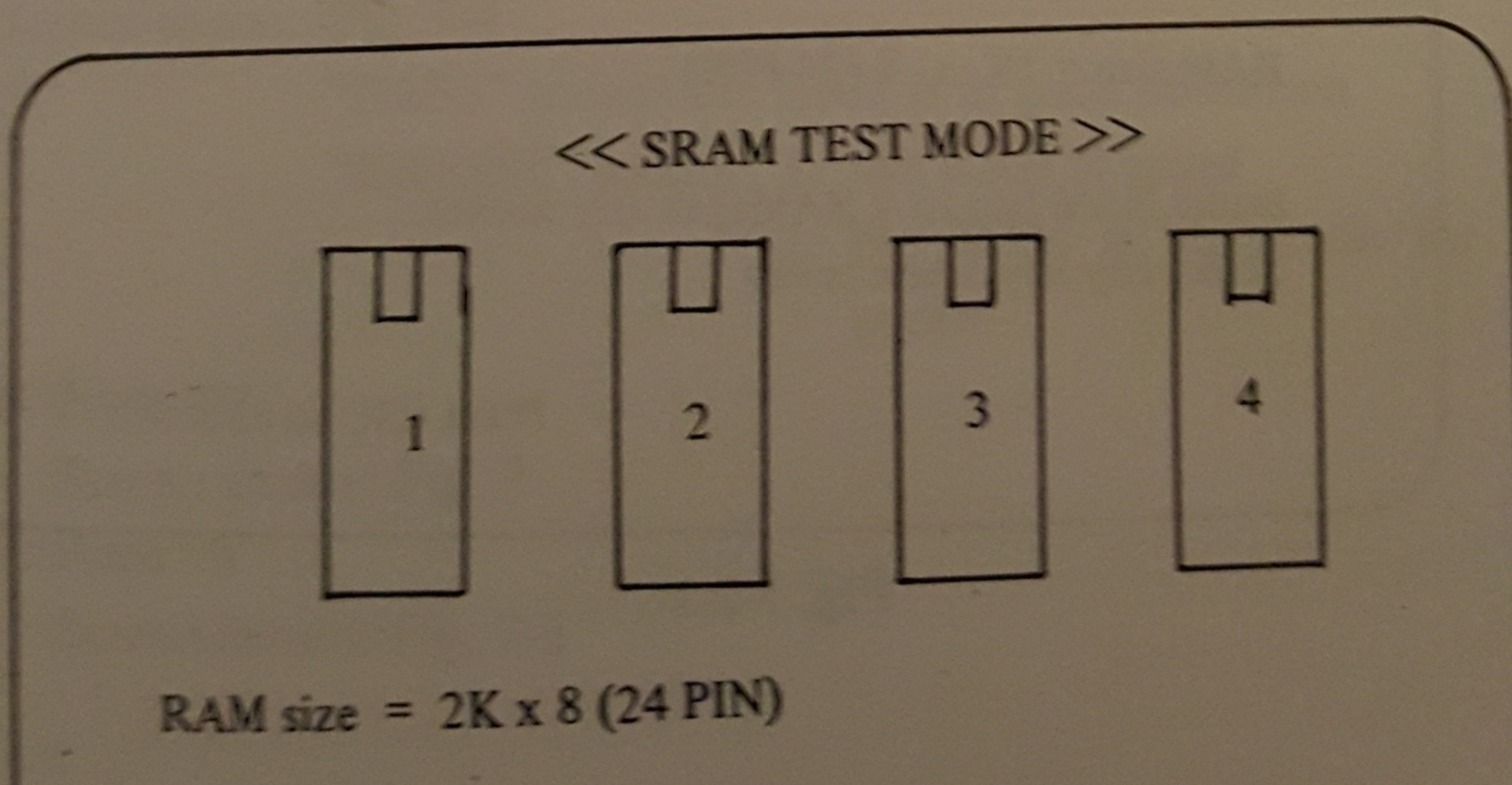


Fig. 7-1

4. If: Key in "Y", screen shows: Fig. 7-2,
   will change RAM type



```
1.  2K   x 8 (24 PIN)
2.  4K   x 8 (24 PIN)
3.  4K   x 8 (28 PIN)
4.  8K   x 8 (28 PIN)
5.  16K  x 8 (28 PIN)
6.  32K  x 8 (28 PIN)

SELECT ONE     ?
```

Fig. 7-2

If: Key in "N", screen shows: Fig. 7-2-1
will not change RAM type, go to step 8

```
                    << SRAM TEST MODE >>

        ┌──┐        ┌──┐        ┌──┐        ┌──┐
        │  │        │  │        │  │        │  │
        │  │        │  │        │  │        │  │
        │  │        │  │        │  │        │  │
```

5. Screen on Fig. 7-2
6. Key in "4" (for RAM size 8K x 8)
   (Select the number of your RAM size)
7. Screen shows: Fig. 7-3

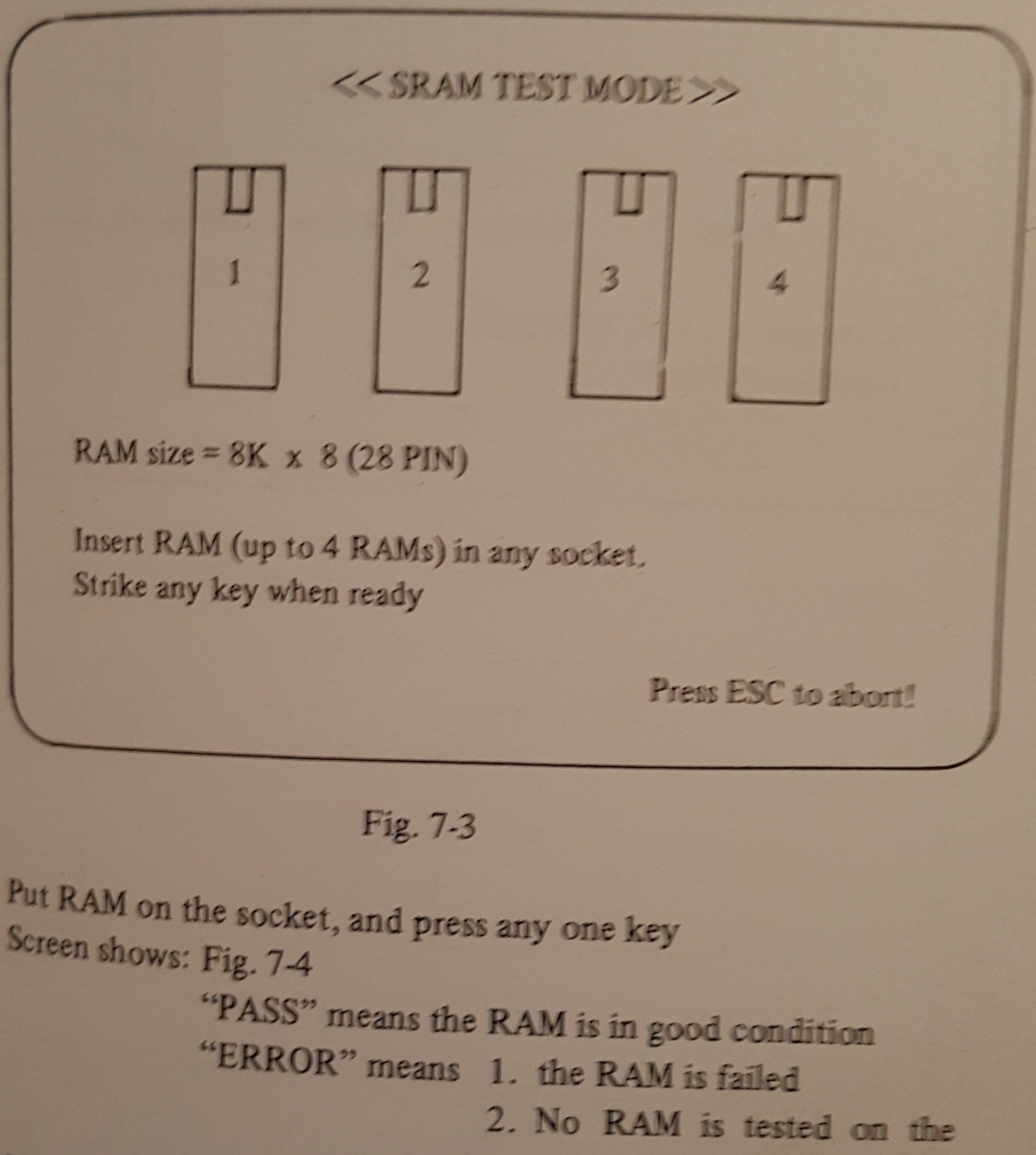


Fig. 7-3

8. Put RAM on the socket, and press any one key
9. Screen shows: Fig. 7-4
   "PASS" means the RAM is in good condition
   "ERROR" means 1. the RAM is failed
   2. No RAM is tested on the socket
10. If: Press "N",
    Screen shows: Fig. 2
    If: Press "Y"
    Screen shows: Fig. 7-1

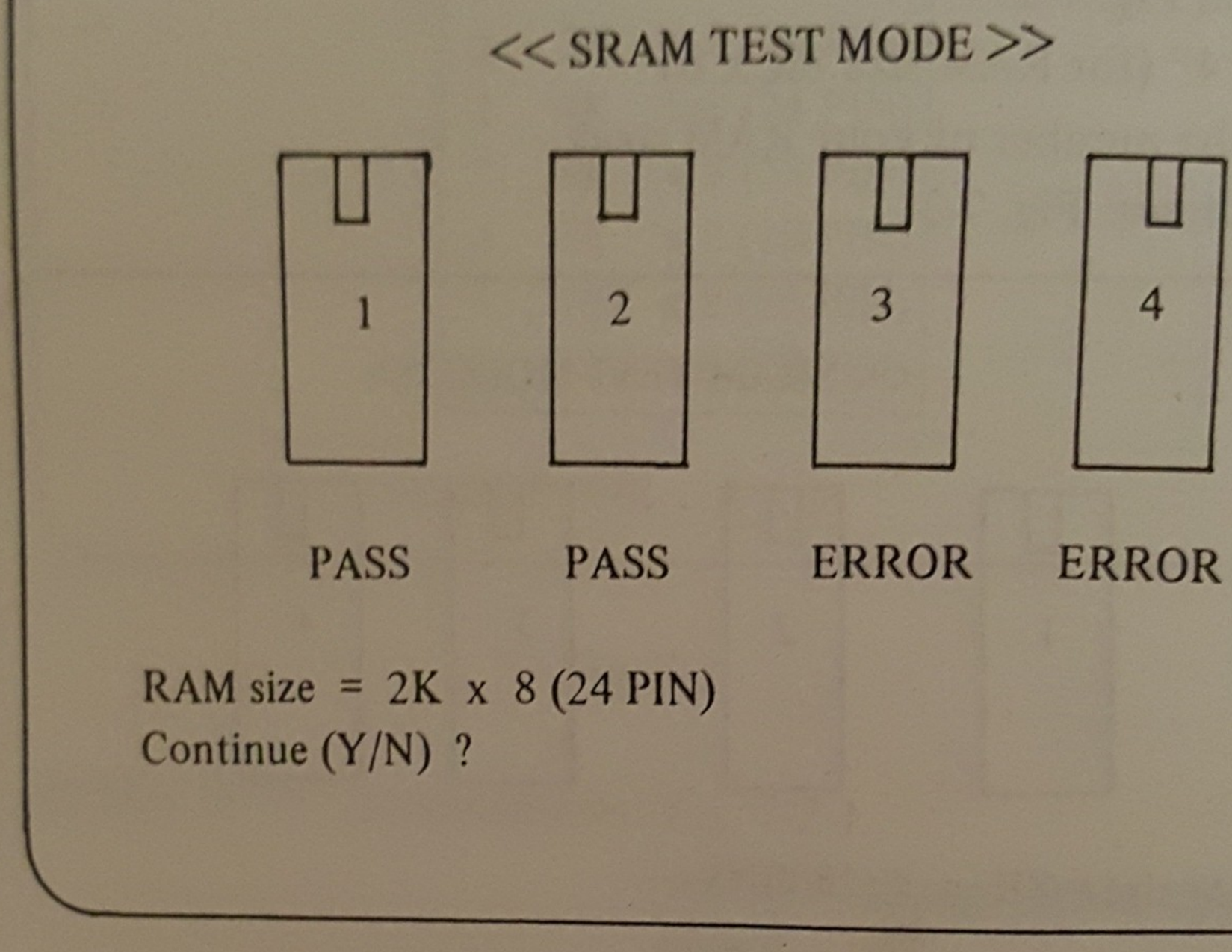

<< SRAM TEST MODE >>
1
2
3
4
PASS
PASS
ERROR
ERROR
RAM size = 2K x 8 (24 PIN)
Continue (Y/N) ?

Fig. 7-4

8. EPROM WRITER

TO READ AND WRITE EPROM, procedure as:

Fig. 2 --> Insert a EPROM --> Type --> Read (--> Edit)
--> Insert a new EPROM --> Blank check --> Write
--> Compare (--> Copy --> File) --> end

1. Screen on Fig. 2
2. Key in "4"
3. Screen shows: Fig. 8-1

<< EPROM WRITER >>

0. Exit
1. Type
2. Read
3. Write
4. Compare
5. Blank check
6. Edit
7. Copy
8. File

SELECT ONE ?

TYPE = 2716 (25V)
SOCKET NUMBER = 4
EPROM START ADDRESS = 0000
BUFFER START ADDRESS = 0000
LENGTH = 00000

Fig. 8-1

Type:

4. Put EPROM on the socket
5. Key in "1" (type)
6. Screen shows: Fig. 8-2
7. Key in "4" (for 2764, 21V)
8. Screen shows: Fig. 8-3 (with type of the EPROM stated)

```
                    << EPROM WRITER >>

        TYPE                VPP
1.      2716,2516           25V
2.      2732,27C32          25V
3.      2732A,27C32A        21V
4.      2764,27C64          21V
5.      2764A               12.5V
6.      27128               21V
7.      27128A              12.5V
8.      27256,27C256        21V
9.      27256,27C256        12.5V
A.      27512,27C512        12.5V

                    SELECT ONE ?

TYPE = 2716 (25V)
SOCKET NUMBER  =  4
EPROM START ADDRESS  =  0000
BUFFER START ADDRESS  =  0000
LENGTH  =  00000
```

Fig. 8-2

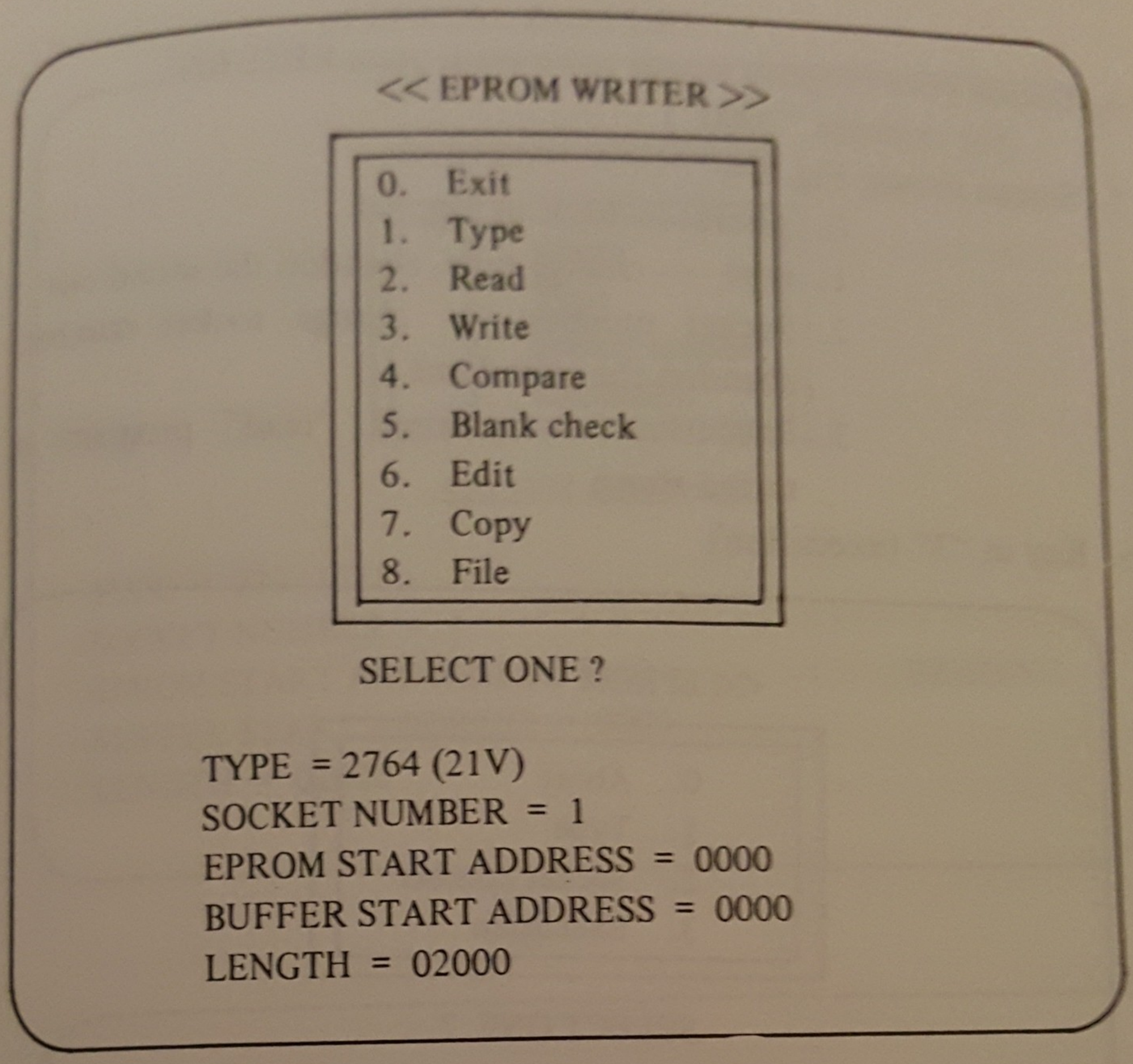


Fig. 8-3

READ (can not escape from reading situation)

9. Screen on Fig. 8-3
10. Key in "2" (read)
11. Select socket number, press RETURN (for example "1")
    (read one piece EPROM once only)
12. Decide your EPROM start address, press return (for example "0000")
    (Refer to Appendix III)
13. Decide your Buffer start address, press RETURN (for example "0000")
    (Refer to Appendix III)

14. Decide your Program length to be read, press RETURN
    (for example "2000")
15. Screen shows: Fig. 8-4
    0. Abort —— back to Fig. 8-1
    1. type —— change type, abandon the setted one
    2. Socket number —— change socket number, abandon the setted one
    3. Execution —— proceed "read" program as setted above
16. Key in "3" (execution)

<< EPROM WRITER >>

0. Abort
1. Type
2. Socket number
3. Execution

SELECT ONE ?

TYPE = 2764 (21V)
SOCKET NUMBER = 1
EPROM START ADDRESS = 0000
BUFFER START ADDRESS = 0000
LENGTH = 02000

Fig. 8-4

17. Screen shows: Fig. 8-5, with "READING" flashing
18. Screen shows: Fig. 8-6, read finished, and CHECKSUM showed, remember this sum to check with the checksum of writing
    (CHECKSUM: the sum of data from start address to end address

of EPROM, BUFFER)

```
<< EPROM WRITER >>

    0.  Abort
    1.  Type
    2.  Socket number
    3.  Execution

SELECT ONE?  3

TYPE = 2764 (21V)
SOCKET NUMBER = 1
EPROM START ADDRESS = 0000                    READING
BUFFER START ADDRESS = 0000
LENGTH = 02000
```

Fig. 8-5

```
<< EPROM WRITER >>

    0.  Exit
    1.  Type
    2.  Read
    3.  Write
    4.  Compare
    5.  Blank check
    6.  Edit
    7.  Copy
    8.  File

SELECT ONE ?
```

TYPE = 2764 (21V)
SOCKET NUMBER = 1
EPROM START ADDRESS = 0000
BUFFER START ADDRESS = 0000
LENGTH = 02000

Checksum = B900

Fig. 8-6

BLANK CHECK (can not escape from Blank checking situation)

19. Take out the original EPROM,
    Put a new EPROM on the socket
20. Key in "5" (blank check)
21. If: Screen shows: Fig. 8-7
    "Blank OK!"
    This means: The EPROM is blank

<< EPROM WRITER >>

0. Exit
1. Type
2. Read
3. Write
4. Compare
5. Blank check
6. Edit
7. Copy
8. File

SELECT ONE ?

TYPE = 27256 (12.5V)
SOCKET NUMBER = 1

EPROM START ADDRESS = 0000
BUFFER START ADDRESS = 0000
LENGTH = 08000

Blank OK!

Fig. 8-7

If: Screen shows: Fig. 8-7-1
"Blank check Error!"

This means: The EPROM has data, can not write again
Different data than the original data in the EPROM can not be written into the EPROM

<< EPROM WRITER >>

0. Exit
1. Type
2. Read
3. Write
4. Compare
5. Blank check
6. Edit
7. Copy
8. File

SELECT ONE ?

TYPE = 27256 (12.5V)
SOCKET NUMBER = 1
EPROM START ADDRESS = 0000
BUFFER START ADDRESS = 0000
LENGTH = 080000

Blank check error!

Fig. 8-7-1

*** **Note**: If a EPROM with "Blank OK", but can not be written program in, then the EPROM could be failed.

WRITE (with auto-compare, and can not escape from comparing situation)

22. Screen on Fig. 8-7
23. Key in "3" (write)
24. Screen shows: Fig. 8-8

0. Abort: back to Fig. 8-1
1. Type: change EPROM type, abandon the setted one
2. Socket number: change socket number, abandon the setted one
3. Execution: proceed "write" program

<< EPROM WRITER >>

0. Abort
1. Type
2. Socket number
3. Execution

SELECT ONE ? 2

TYPE = 2764 (21V)
SOCKET NUMBER = (Number or A = A11)
EPROM START ADDRESS = 0000
BUFFER START ADDRESS = 0000
LENGTH = 02000

Fig. 8-8

25. Key in "3" (execution)

26. Screen shows: Fig. 8-9
The "write" program is proceeding, and screen with "WRITING" flashing.

Please refer to appendix IV —— EPROM WRITER OPERATION TIME

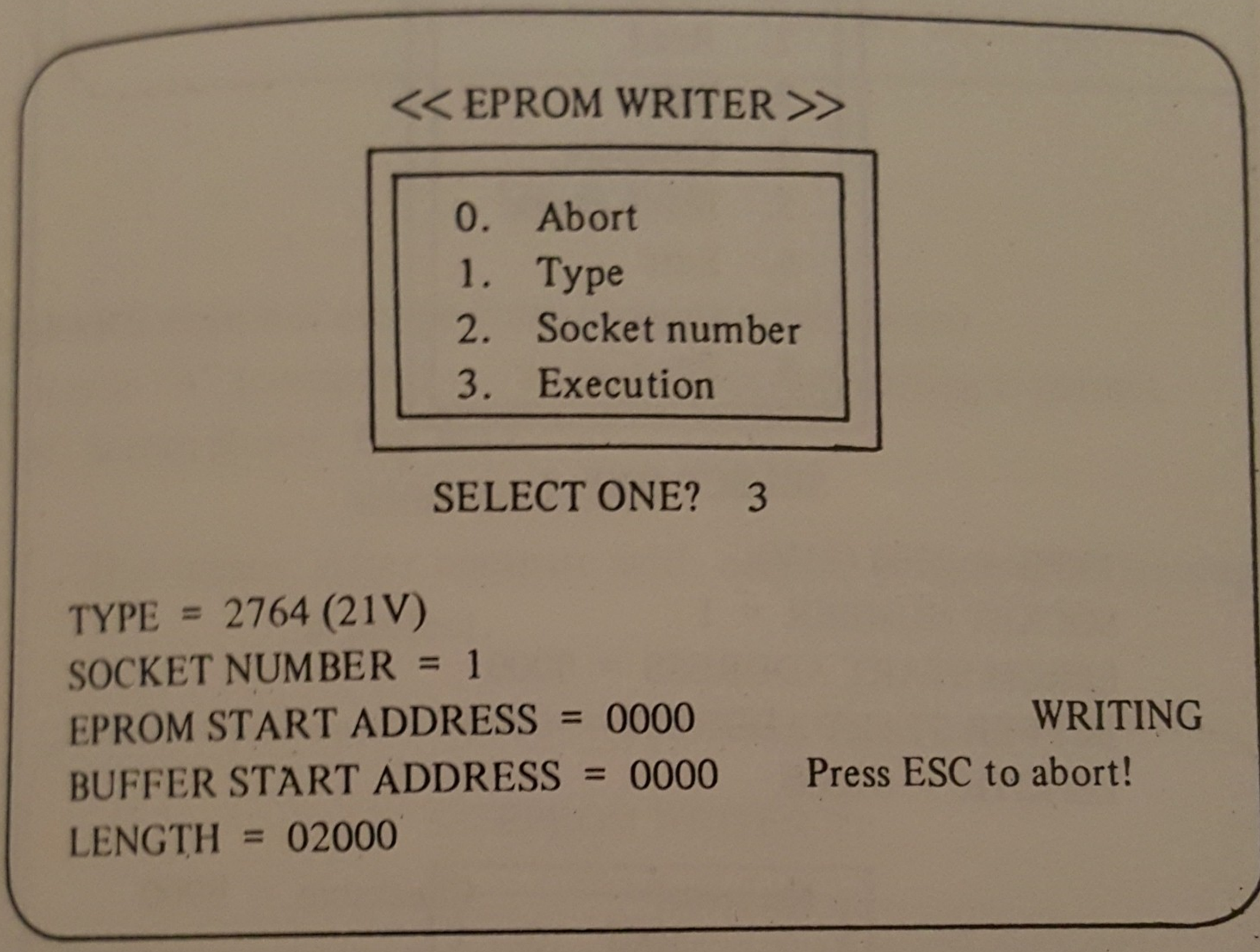


Fig. 8-9

27. If: Screen shows: Fig. 8-10
with "No error" & checksum
This means: After auto-compare with original program, the writing is perfect.
If: Screen shows: Fig. 8-10-1
with "Compare error"
This means: After auto-compare with original program, found the writing is failed.

```
<< EPROM WRITER >>

    0.  Exit
    1.  Type
    2.  Read
    3.  Write
    4.  Compare
    5.  Blank check
    6.  Edit
    7.  Copy
    8.  File

SELECT ONE ?

TYPE = 2764 (21V)
SOCKET NUMBER = 1
EPROM START ADDRESS = 0000
BUFFER START ADDRESS = 0000
LENGTH = 02000

No error!          Checksum = B900
```

Fig. 8-10

```
<< EPROM WRITER >>

    0.  Abort
    1.  Type
    2.  Socket number
    3.  Execution

SELECT ONE?  3
```

```
TYPE = 2764 (21V)
SOCKET NUMBER = 1
EPROM START ADDRESS = 0000                                  WRITING
BUFFER START ADDRESS = 0000     Press ESC to abort !
LENGTH = 02000

                    Compare error!      Press any key to continue.
```

Fig. 8-10-1

COMPARE (can not escape from comparing situation)

28. Key in "4" (compare) —— to make sure the writing is correct.
29. If: Screen shows: Fig. 8-11
    with "No error"
    This means: After compare with original program, the writing is perfect.

```
                    << EPROM WRITER >>

                        0.  Exit
                        1.  Type
                        2.  Read
                        3.  Write
                        4.  Compare
                        5.  Blank check
                        6.  Edit
                        7.  Copy
                        8.  File

                    SELECT ONE ?

TYPE = 2764 (21V)
SOCKET NUMBER = 1
```

```
EPROM START ADDRESS  =  0000
BUFFER START ADDRESS  =  0000
LENGTH  =  02000

                                No error !
```

Fig. 8-11

If: Screen shows: Fig. 8-11-1

This means: The writing is failed.

The difference of EPROM and BUFFER will be put in contrast and showed on screen as Fig. 39.

```
TYPE  =  2764 (21V)
SOCKET NUMBER  =  1
EPROM START ADDRESS  =  0000
BUFFER START ADDRESS  =  0000
LENGTH  =  02000

   BUFFER                    EPROM
   [1631] = 23               [1631] = 04
   BUFFER                    EPROM
   [1632] = 45               [1632] = 32
   BUFFER                    EPROM
   [1633] = 67               [1633] = C8
   BUFFER                    EPROM
   [1634] = 26               [1634] = EB
   BUFFER                    EPROM
   [1635] = 57               [1635] = 04
   BUFFER                    EPROM
   [1636] = 88               [1636] = 22
   BUFFER                    EPROM
   [1637] = 90               [1637] = CC
                   Compare error!  Press any key to continue.
```

Fig. 8-11-1

EDIT — To modify data of Buffer

1. Screen on Fig. 8-1
2. Key in "6" (Edit)
3. Screen shows: Fig. 9-1 (the EPROM with data)
   or Fig. 9-1-1 (the EPROM is blank)

<<EDIT MODE>>

```
0000 : 28 43 29 20 49 42 53 20 -- 55 6C 74 72 61 20 62 6F
0010 : 61 72 64 20 20 76 65 72 -- 73 69 6F 6E 20 32 2E 30
0020 : 20 31 39 38 35 0D 0A 20 -- 49 6E 74 65 72 6E 61 74
0030 : 69 6F 6E 61 6C 20 42 75 -- 73 69 6E 65 73 73 20 53
0040 : 79 73 74 65 6D 0D 0A 0A -- 00 00 00 00 00 00 00 00
0050 : 00 00 00 00 00 BC 03 78 -- 03 78 02 B8 40 00 8E D8
0060 : C7 06 72 00 00 00 FA 33 -- C0 72 48 70 46 78 44 75
0070 : 42 7B 40 05 01 00 74 3B -- 7A 39 2D 02 80 78 34 40
0080 : 71 31 D1 E0 73 2D 75 2B -- D1 E0 72 27 BB 55 55 8B
0090 : EB 8B DC 8B E1 8B D4 8E -- D2 8C D6 8E C6 8C C7 8E
00A0 : DF 8C D8 3D 55 55 75 06 -- F7 D0 8B D8 EB E1 35 AA
00B0 : AA 74 01 F4 FC B0 00 E6 -- A0 BA D8 03 EE BA B8 03
00C0 : FE C0 EE B0 99 E6 63 B0 -- A5 E6 61 B0 54 E6 43 B0
00D0 : 12 E6 41 32 C0 E6 83 E6 -- 81 E6 82 E6 0D B0 58 E6
00E0 : 0B B0 41 E6 0B B0 42 E6 -- 0B B0 43 E6 0B B0 FF E6
00F0 : 01 E6 01 B0 00 E6 08 E6 -- 0A B0 36 E6 43 B0 00 E6
0100 : 40 E6 40 BA 13 02 B0 01 -- EE B8 40 00 8E D8 8B 36
0110 : 72 00 33 C0 8B E8 8B D8 -- BA AA 55 FC 33 FF 8E C3
0120 : 26 89 15 26 3B 15 75 0D -- B9 00 20 F3 AB 80 C7 04
0130 : 80 FF C0 75 E7 89 36 72 -- 00 33 C0 8E C0 B8 80 00
0140 : 8E D0 BC 00 01 55 53 BD -- 02 00 E8 C1 18 58 B1 06
0150 : D3 E8 A3 13 00 58 73 02 -- 0C 10 A2 15 00 33 C0 50
0160 : 50 50 50 50 B8 30 00 8E -- D0 BC 00 01 1E BB 00 E0
0170 : 0E 1F F4 01 E8 8A 18 1F -- 74 05 80 0E 15 00 00 FA
0180: : B0 13 E6 20 B0 08 E6 21 -- B0 09 E6 21 B0 FF E6 21
```

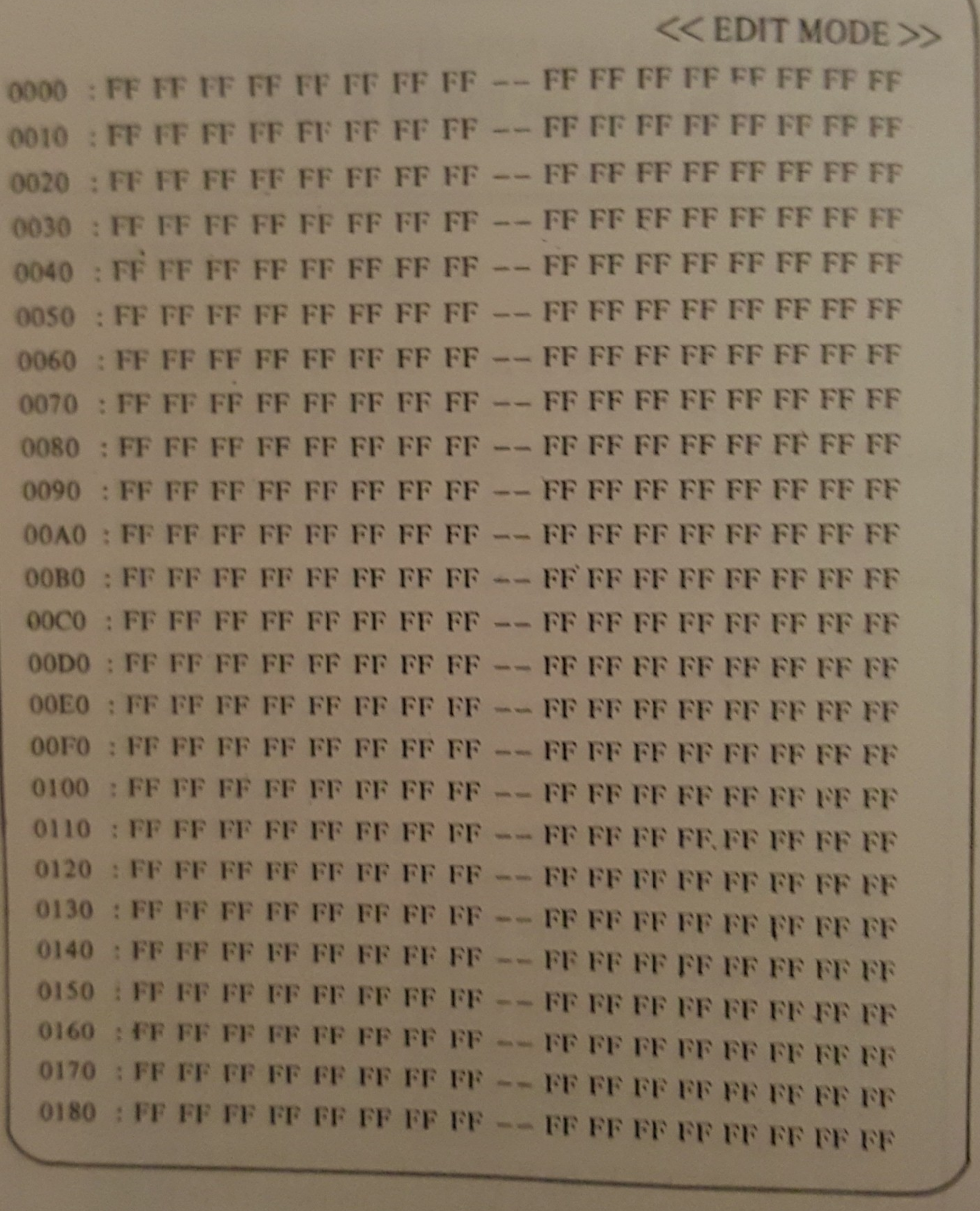


Fig. 9-1-1

4. Press "F1" key to enter COMMAND mode
5. Screen shows: Fig. 9-2

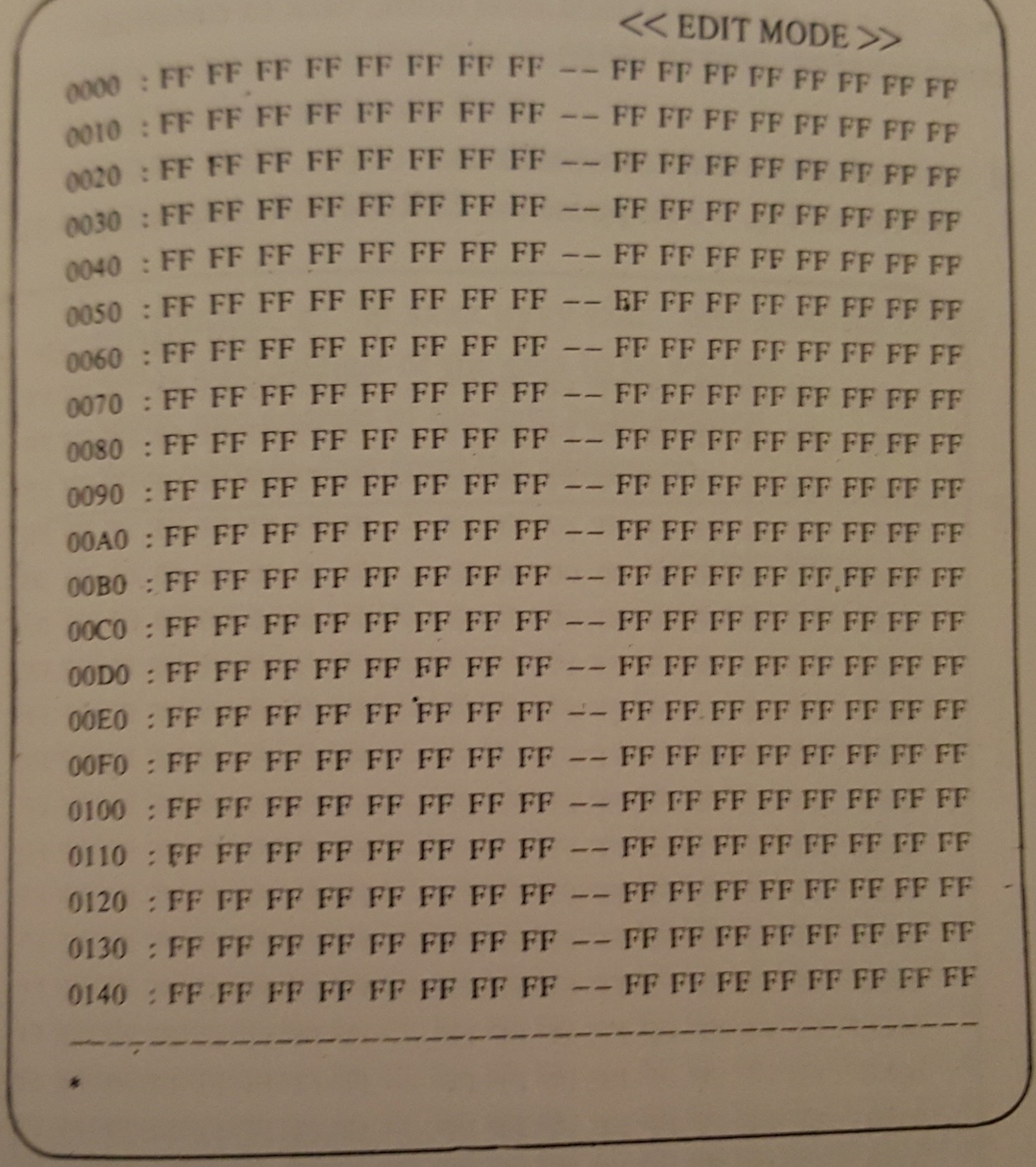


Fig. 9-2

*** COMMAND **********

** There are five signals of COMMAND to proceed Edit:

** 1. M -- move, write data into a address

** 2. F -- fill, fill data into a address

** 3. J -- jump, move cursor to a pointed address

** 4. S -- check, make the sum between two addresses

** 5. Q —— quit, leave COMMAND mode, back to edit mode
****

M —— move, write data into a address

1. Screen on Fig. 9-3 (suppose the EPROM with data as Fig. 9-3)

```
                                                    << EDIT MODE >>
0000  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0010  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0020  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0030  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0040  : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0050  : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0060  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0070  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0080  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0090  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00A0  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00B0  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00C0  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00D0  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00E0  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00F0  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0100  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0110  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0120  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0130  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0140  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
--------------------------------------------------------------
*
```

Fig. 9-3

2. Key in "M 0040 005F 0100", then press RETURN
   or "M 40 5F 100", then press RETURN as Fig. 9-4
   this means: write the data between 0040 & 005F to a block,
   and the start address is 0100
   (0040, 005F, 0100 are addresses, showed on left line of figure)

   the formula is: "M xxxx yyyy zzzz" then press RETURN
   xxxx: start address of the data to be written
   yyyy: end address of the data to be written
   zzzz: write in data from the address on

```
                                    << EDIT MODE >>
0000 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0010 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0020 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0030 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0040 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0050 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0060 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0070 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0080 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0090 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00A0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00B0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00C0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00D0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00E0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00F0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0100 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0110 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0120 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0130 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0140 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
-------------------------------------------------------
*M 40 5F 100
```

Fig. 9-4

. Screen shows: Fig. 9-5

The data has been written to the destined address, the same data length.

```
                                                  <<EDIT MODE>>
0000 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0010 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0020 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0030 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0040 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0050 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0060 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0070 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0080 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0090 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00A0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00B0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00C0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00D0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00E0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00F0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0100 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0110 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0120 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0130 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0140 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
-------------------------------------------------------
*M 40 5F 100
```

Fig. 9-5

****

F -- fill, fill data into a address

1. Screen on Fig. 9-3
2. Key in "F 0100 013F 33" then press RETURN
   or "F100 13F 33" then press RETURN as Fig. 9-6
   this means: Fill data "33" into the block from 0100 to 013F
   (0100, 013F are addresses, showed on left line of Figure, 33
   is the data to be written on)

   the formula is: F xxxx yyyy AA
   or F xxxx yyyy BBBB
   xxxx: start address to write the data on
   yyyy: end address to write the data
   AA: two signal data to be written into the block
   BBBB: four signal data to be written into the block

```
                                        << EDIT MODE >>
0000 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0010 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0020 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0030 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0040 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0050 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0060 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0070 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0080 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0090 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00A0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00B0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00C0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00D0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00E0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00F0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0100 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0110 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0120 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0130 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0140 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
*F 100 13F 33
```

Fig. 9-6

3. Screen shows: Fig. 9-7
The data "33" has been written into the block

```
                                    <<EDIT MODE>>

0000 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0010 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0020 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0030 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0040 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0050 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0060 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0070 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0080 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0090 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00A0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00B0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00C0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00D0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00E0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00F0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0100 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0110 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0120 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0130 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0140 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF

*F 100 13F 33
*
```

Fig. 9-7

****

J — Jump, move cursor to a pointed address

1. Screen on Fig. 9-7
2. Key in "J 0400"
   or "J 400" as Fig. 9-8
   this means: move cursor to address 0400
   (0400 is a address showed on left line of figure)

the formula is: J xxxx

xxxx: a address destined for cursor moving to

```
                                                    << EDIT MODE >>
0000  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0010  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0020  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0030  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0040  : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0050  : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0060  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0070  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0080  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0090  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00A0  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00B0  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00C0  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00D0  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00E0  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00F0  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0100  : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0110  : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0120  : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0130  : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0140  : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
-------------------------------------------------------------

*J 400
```

Fig. 9-8

3. Screen shows: Fig. 9-9

The first row is the 0400.

```
                                    ≪EDIT MODE≫
0400 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0410 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0420 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0430 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0440 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0450 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0460 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0470 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0480 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0490 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
04A0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
04B0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
04C0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
04D0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
04E0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
04F0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0500 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0510 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0520 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0530 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0540 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
--------------------------------------------------------
*J 400
*
```

Fig. 9-9

****

S -- check, make the sum between two address

1. Screen on Fig. 9-9

2. Key in "S 0000 014F"
   or "S 0 14F" as Fig. 9-10
   this means: check the sum from address 0000 to 014F

the formula is: S xxxx yyyy
xxxx: start address
yyyy: end address

```
                                        << EDIT MODE >>
0000 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0010 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0020 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0030 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0040 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0050 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0060 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0070 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0080 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0090 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00A0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00B0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00C0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00D0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00E0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00F0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0100 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0110 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0120 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0130 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0140 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
-------------------------------------------------------

*S 0 14F
```

Fig. 9-10

3. Screen shows: Fig. 9-11

The checksum showed

```
                                        <<EDIT MODE>>
0000 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0010 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0020 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0030 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0040 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0050 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0060 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0070 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0080 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0090 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00A0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00B0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00C0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00D0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00E0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00F0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0100 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0110 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0120 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0130 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0140 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
--------------------------------------------------------

CHECK-SUM = FBD0
*
```

Fig. 9-11

```
**************
Q --- quit, leave COMMAND mode, back to edit mode
1. Screen on Fig. 9-11
2. Key in "Q" as Fig. 9-12
```

```
                                                    <<EDIT MODE>>
0000 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0010 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0020 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0030 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0040 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0050 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0060 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0070 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0080 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0090 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00A0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00B0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00C0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00D0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00E0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00F0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0100 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0110 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0120 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0130 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0140 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
-------------------------------------------------------------

*Q
```

Fig. 9-12

3. Screen shows: Fig. 9-13

```
                                    <<EDIT MODE>>
0000 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0010 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0020 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0030 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0040 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0050 : 00 00 00 00 00 00 00 00 -- 00 00 00 00 00 00 00 00
0060 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0070 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0080 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0090 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00A0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00B0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00C0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00D0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00E0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
00F0 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0100 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0110 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0120 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0130 : 33 33 33 33 33 33 33 33 -- 33 33 33 33 33 33 33 33
0140 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0150 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0160 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0170 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
0180 : FF FF FF FF FF FF FF FF -- FF FF FF FF FF FF FF FF
```

Fig. 9-13

COPY — copy from one EPROM to another or others
(Auto-Blank check)

1. Screen shows: Fig. 8-1
2. Key in "7" (copy)
3. Screen shows: Fig. 10-1

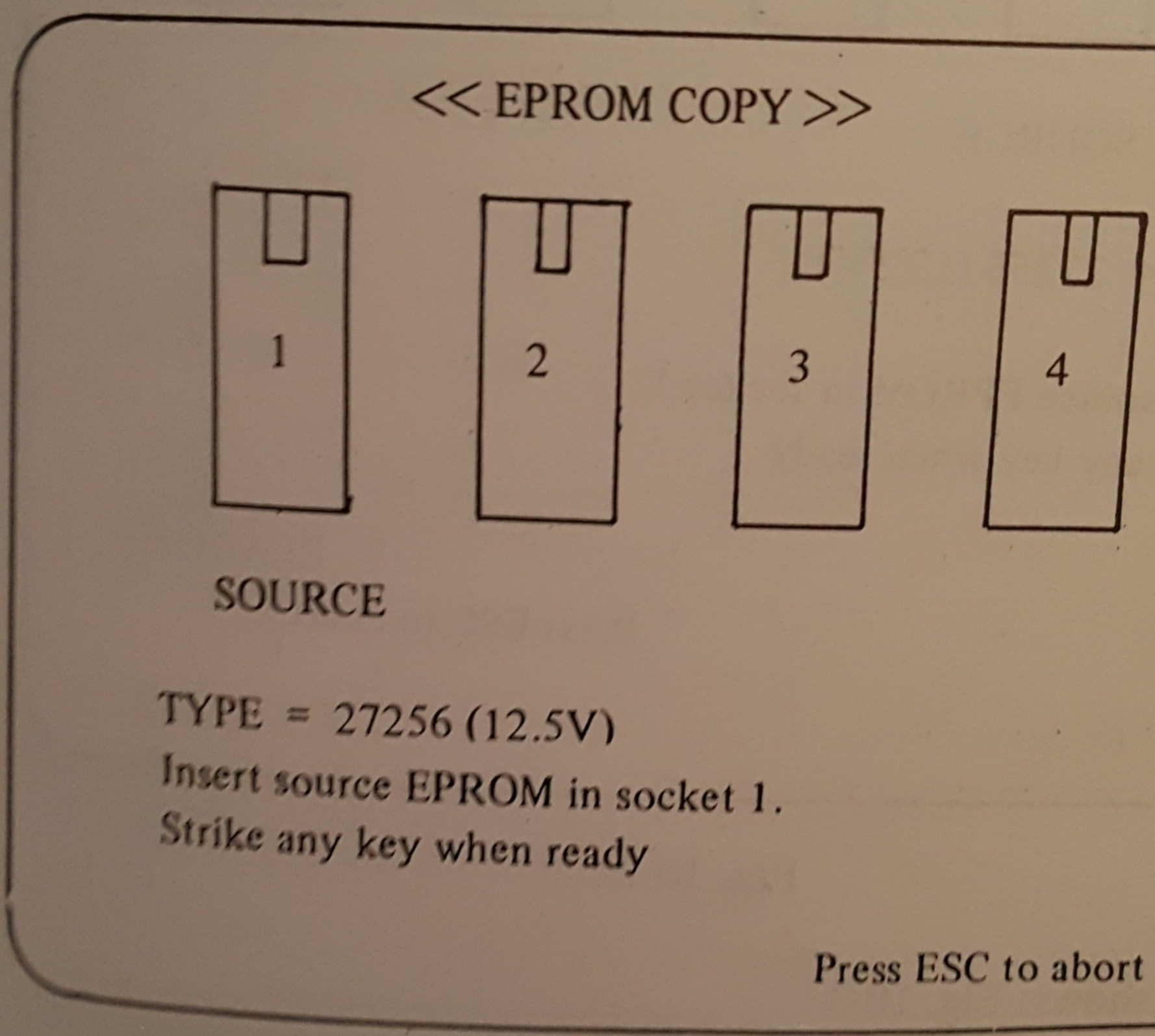


Fig. 10-1

4. Put one EPROM on the socket, press RETURN
5. Screen shows: Fig. 10-2
with "READING" flashing

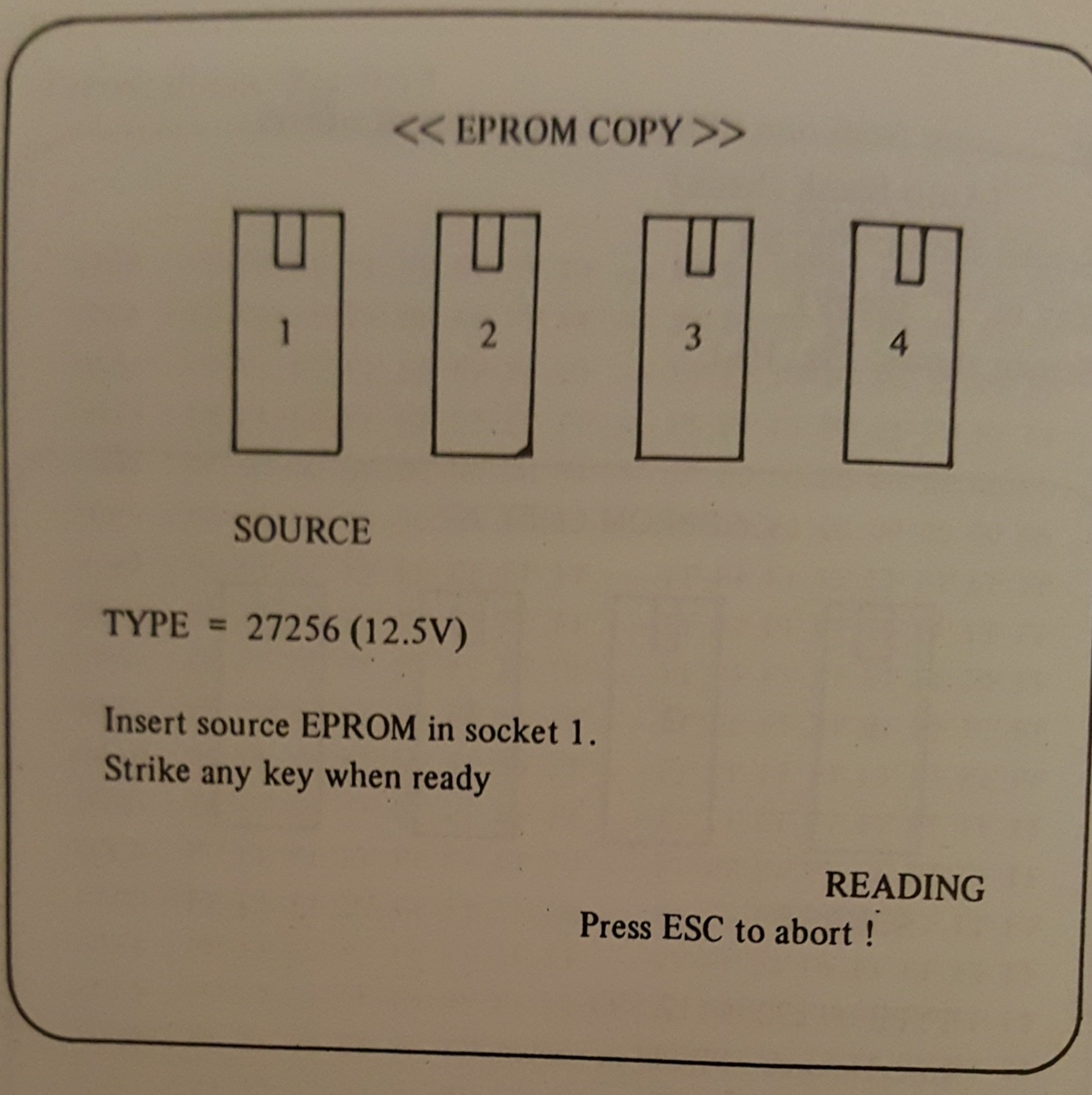


Fig. 10-2

6. Screen shows: Fig. 10-3

   Reading finished, with checksum showed.
7. Take out the source EPROM, put blank EPROM on sockets, then press any one key.
   (the EPROM number could be 1 to 4)
8. Screen shows: Fig. 10-3

   Blank check is in process automatically from 1 to 4, one by one. And the LED on the device will light up one by one in blank checking.

   If: Screen shows: Fig. 10-3

   This means: all EPROMs are blank, copy can be proceeded normally.

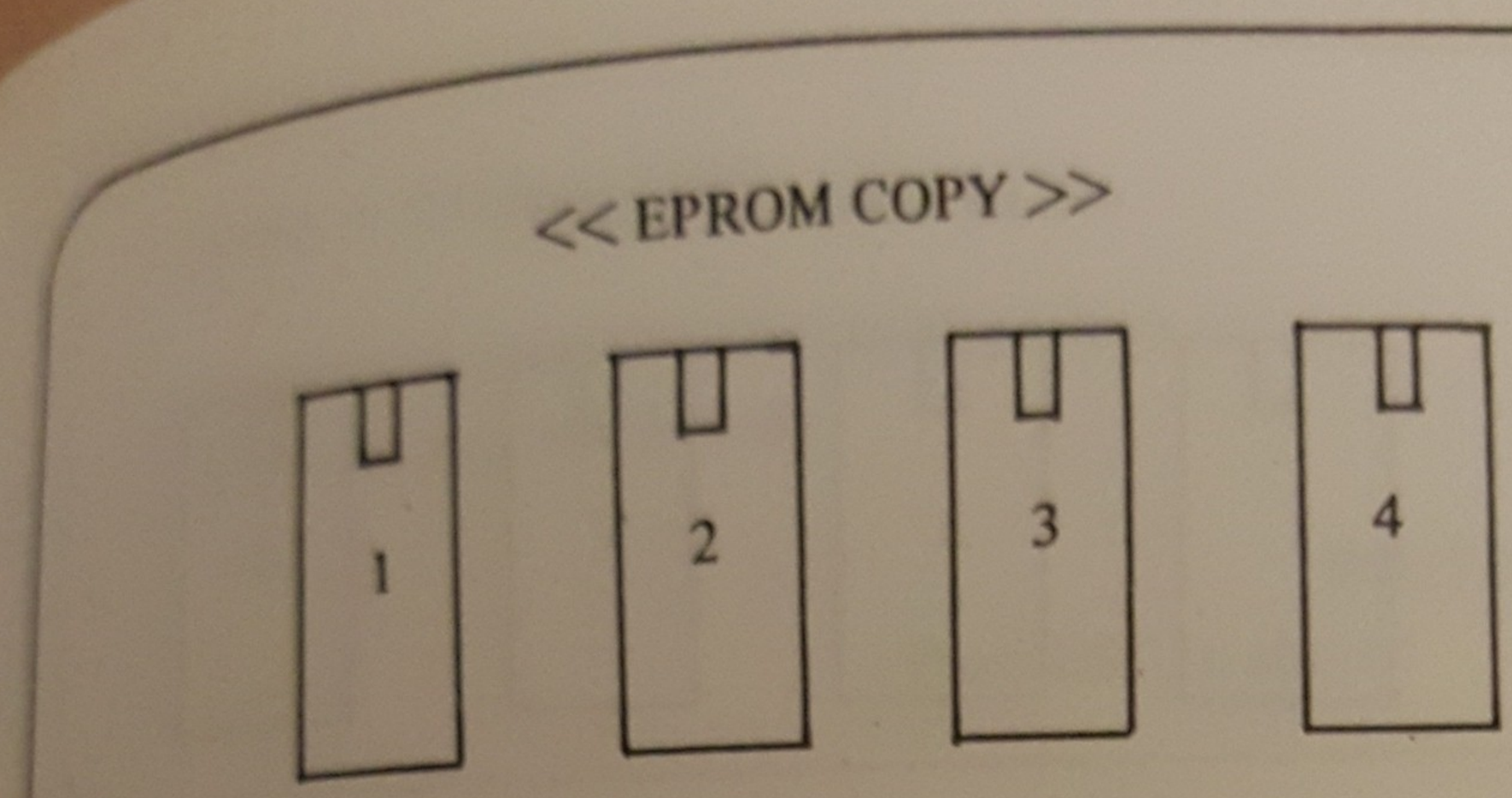


Fig. 10-3

If: Screen shows: Fig. 10-3-1

This means: the 4th EPROM is not blank, and the device will not work. Take out the 4th EPROM (the one is not blank) so that the COPY procedure can go forth; or replace a blank EPROM.

When all destinations got ready, press any one key.

<< EPROM COPY >>

1 2 3 4

DESTINATION DESTINATION DESTINATION Not Blank

TYPE = 27256 (12.5V)

Please insert blank EPROM and
Strike any key when ready

Press ESC to abort !

Checksum = 1FF7

Fig. 10-3-1

9. Screen shows: Fig. 10-4
with "WRITING" flashing
All LEDs will light up in writing.

<< EPROM COPY >>

1 2 3 4

DESTINATION DESTINATION DESTINATION DESTINATION

TYPE = 27256 (12.5V)

Insert destination EPROMs in all sockets.
Strike any key when ready

WRITING

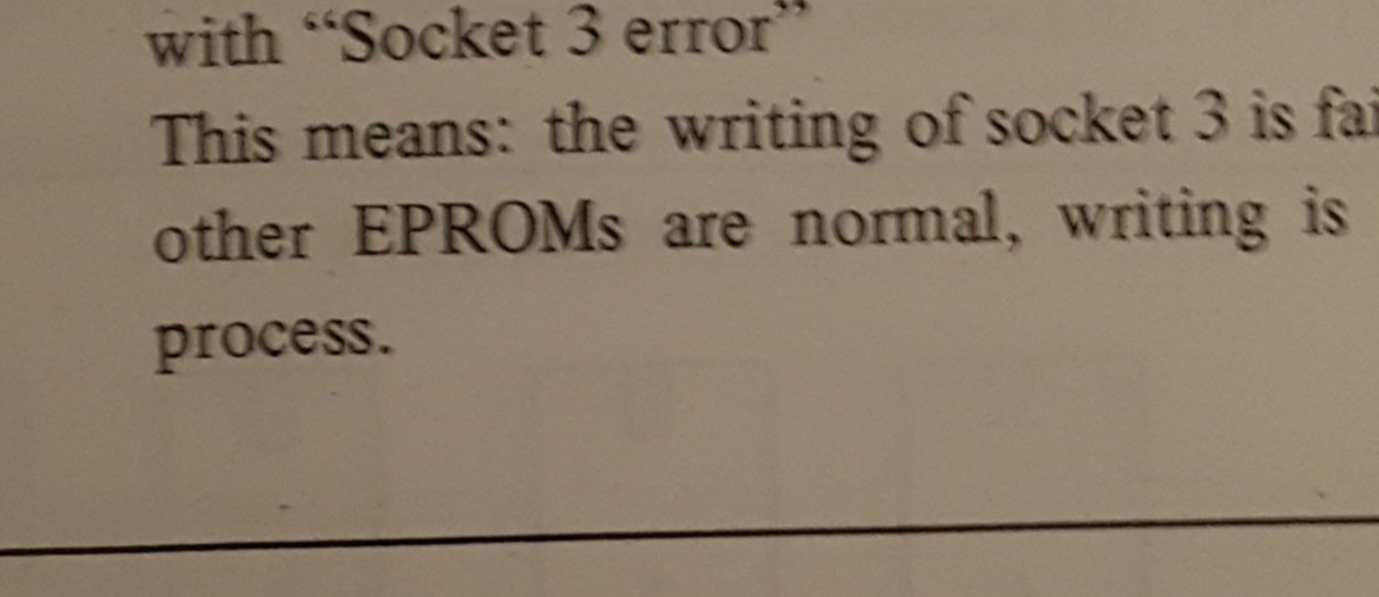
Press ESC to abort !

Fig. 10-4

If: Screen shows: Fig. 10-4
This means: Writing is in good condition

If: Screen shows: Fig. 10-4-1
with "Socket 3 error"
This means: the writing of socket 3 is failed, of other EPROMs are normal, writing is still in process.

<< EPROM COPY >>



DESTINATION DESTINATION DESTINATION DESTINATION

TYPE = 27256 (12.5V)

Insert destination EPROMs in all sockets.
Strike any key when ready

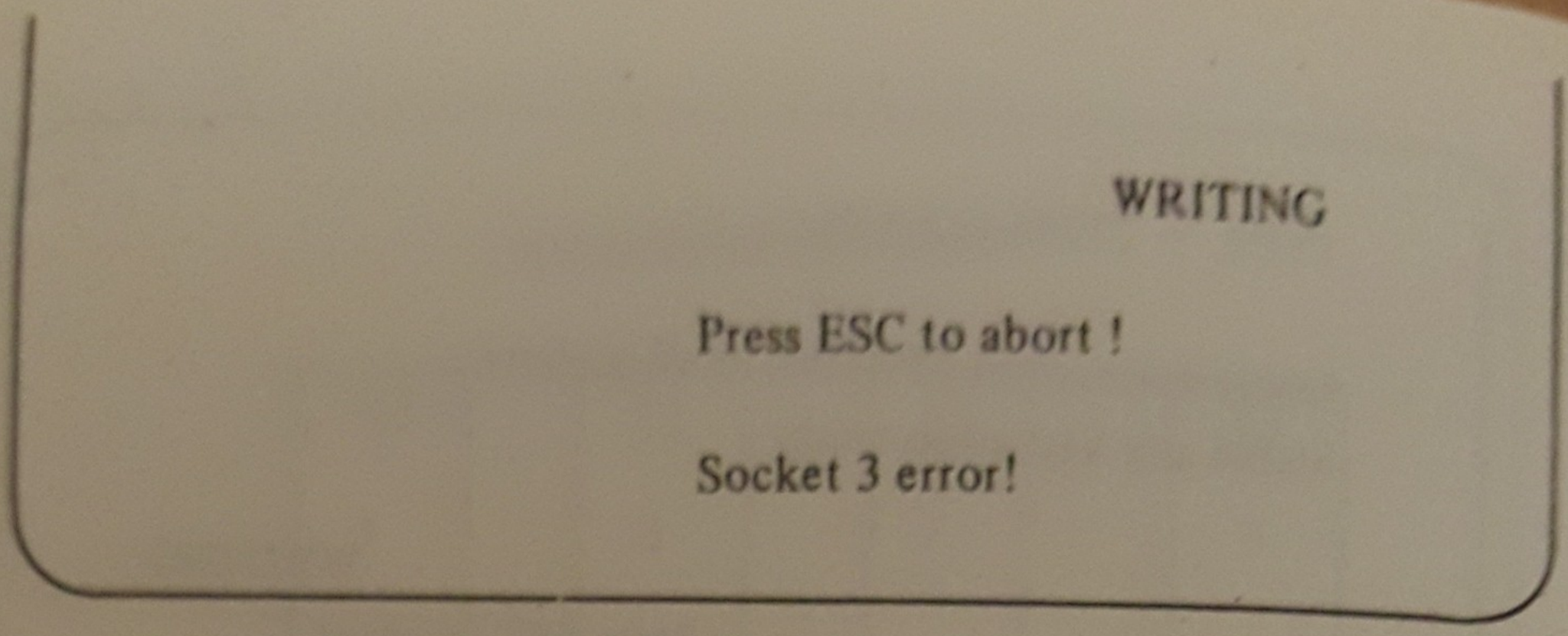


Fig. 10-4-1

10. Screen shows: Fig. 10-5
    Writing is completed, with checksum showed.
11. If there are any other EPROMs are for copy,
    Press "Y", and practice all procedure accordingly.
    If you do not want to copy anymore,
    Press "N", and screen back to Fig. 8-1

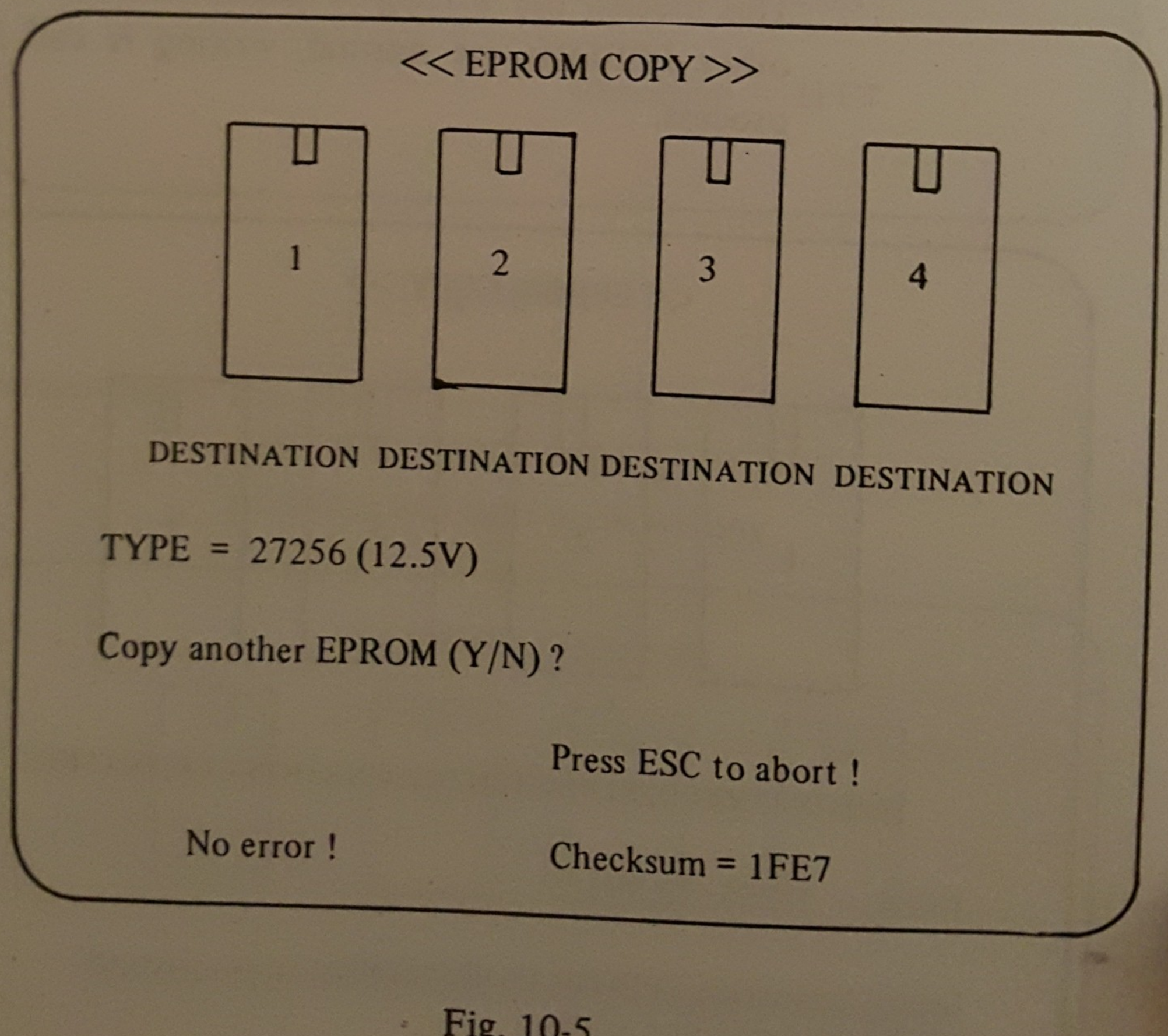


Fig. 10-5

FILE -- Save data from file to buffer
or from buffer to file

1. Screen on Fig. 8-1
2. Key in "8" (file)
3. Screen shows: Fig. 11-1
with 1. Read File
2. Write File
3. Back to previous screen

<< FILE MODE >>

(1) READ FILE (2) WRITE FILE
(3) BACK TO PREVIOUS SCREEN

KEY IN A NUMBER TO CONTINUE . . .

Fig. 11-1

Read File

1. Key in "1" (read file)
2. Screen shows: Fig. 11-2
with "FILENAME (UPPER CASE):"
This means: key in file name in capital letter
3. Key in "YL.EXE" (suppose the YL.EXE is the file name), and press "RETURN"
4. File YL.EXE was read.
5. When file reading finished, screen go back to Fig. 8-1

<< FILE MODE >>

(1) READ FILE (2) WRITE FILE
(3) BACK TO PREVIOUS SCREEN

KEY IN A NUMBER TO CONTINUE 1

FILENAME (UPPER CASE) : YL.EXE

Fig. 11-2

Write File

1. Key in "2"
2. Screen shows: Fig. 11-3
   with "FILENAME (UPPER CASE): "
   This means: key in file name in capital letter
3. Key in "TEST.P" (suppose the TEST.P is the file name), and press "RETURN"
4. Screen shows: "STARTING ADDRESS (hex):"
5. Key in starting address in hexadecimal digit, press RETURN
6. Screen shows: "LENGTH (hex, unit: kbytes, max 40 (= 64 Kbytes)):
7. Key in the length of file, press RETURN
8. When file writing completed, screen go back to Fig. 8-1

```
                    << FILE MODE >>

        (1) READ FILE      (2) WRITE FILE
        (3) BACK TO PREVIOUS SCREEN


        KEY IN A NUMBER TO CONTINUE . . . .   2

        FILENAME (UPPER CASE): TEST.P

        STARTING ADDRESS (hex): 0
        LENGTH (hex. unit: Kbytes, max 40 (=640 Kbytes)) : 12
```

Fig. 11-3

APPENDIX

This IC device types that YIH LUNG IC tester & EPROM writer Version 1.0 can test are

1. 54/74 TTL series/LS

00, 01, 02, 03, 04, 05, 06, 07, 08, 09,
10, 11, 12, 13, 14, 15, 16, 17, 20, 21,
22, 23, 24, 25, 26, 27, 28, 30, 32, 33,
37, 38, 40, 41, 42, 43, 44, 45, 46, 47, 48,
50, 51, 54, 55, 64, 70, 72, 74, 75, 76, 78,
80, 82, 83, 85, 86, 89, 90, 93, 107, 109, 112,
125, 126, 128, 132, 133, 136, 137, 138, 139, 140, 141,
145, 147, 148, 151, 153, 155, 156, 157, 158, 161, 164,
165, 166, 170, 174, 175, 180, 182, 183, 184, 185, 189,
190, 192, 193, 195, 197, 240, 241, 243, 244, 245, 246,
247, 248, 251, 253, 257, 258, 259, 260, 265, 266, 273,
279, 280, 283, 298, 352, 353, 365, 366, 368, 373, 374,
375, 377, 393, 425, 534, 540, 541, 573, 645, 670, 688,
689, 795, 796, 798,

2. CMOS series

4000, 4001, 4002, 4008, 4009, 4010, 4011, 4012,
4013, 4015, 4016, 4018, 4020, 4023, 4025, 4027,
4028, 4030, 4035, 4040, 4041, 4042, 4043, 4044,
4066, 4068, 4069, 4070, 4071, 4072, 4073, 4075,
4076, 4077, 4078, 4081, 4082, 4093, 4161, 4174,
4175, 4192, 4193, 4502, 4502, 4503, 4511, 4512,
4516, 4518, 4519, 4520, 4522, 4529, 4532, 4555,
4556, 4572, 4584,

EPROM Number	Brand
2716 (25V)	FUJITSU, HITACHI, INTEL, MITSUBISHI, TI
2732 (25V)	FUJITSU, MITSUBISHI, NEC, TOSHIBA, HITACHI
2732A (21V)	FUJITSU, TI
2764 (21V)	INTEL, HITACHI, MITSUBISHI
27128 (21V)	NEC, TOSHIBA
27256 (12.5V)	HITACHI, INTEL, MITSUBISHI
27256 (21V)	FUJITSU, NEC
27512 (12.5V)	AMD

APPENDIX I

<< DRAM TEST OPERATION TIME >>
(Base on AT mode)

TYPE	SPEED (sec.)
4416	6
4464	24
4164	30
41256	114

APPENDIX II

EPROM TYPE	MEMORY BUFFER	LENGTH
2716	0000–07FF	800
2732 (A)	0000–0FFF	1000
2764 (A)	0000–1FFF	2000
27128 (A)	0000–3FFF	4000
27256 (A)	0000–7FFF	8000
27512 (A)	0000–FFFF	10000

APPENDIX III

<< EPROM WRITER OPERATION TIME >>

UNIT: second
(Base on AT mode)

TYPE	READ	WRITE (1)	COPY (4)
2716	1	6	20
2732	1	15	52
2764	1	8	14
27128	2	9	21
27256	4	23	62
27512	10	32	80

* Quick Pulse

APPENDIX IV