

DJS — 100 — 30 电子数字计算机

程序系統

第 一 冊

内存檢查程序

北京无线电三厂編印

一九七五年五月

前 言

本资料是根据编者学习 NOVA 1200 机内存检查程序 I、II、III 的读书笔记整理而成。供内存调机使用，也作为编制 DJS-130 机的内存检查程序时参考。

内存检查 II 和 III 都是采用在读出线上产生最坏布局下的干扰条件，以对存储器进行检测。在内存检查 II 的基础上增加了测内存大小、浮动干扰程序、检测地址寄存器的每一位等内容之后而形成的内存检查 III，包括了内存检查 II 的全部功能。因此，对于框图部分，本资料仅提供内存检查 I 和 III 的框图。读者在学习内存检查 II 时可参阅检查 III 的框图部分。

由于时间仓促，水平所限，未免有错，敬希读者提出宝贵意见。

北京无线电三厂 DJS-130 软件组

一九七五年五月三日

内存检查程序目录

一、内存检查 I

1. 摘要及操作过程说明	1
2. 程序部分	2
3. 框图部分	3

二、内存检查 II

1. 程序说明及操作过程	4
2. 程序部分	6

三、内存检查 III

1. 简要说明及操作过程	11
2. 程序部分	12
3. 框图部分	27

四、对内存最坏布局干扰示意图	39
----------------------	----

内存检查 I —— 地址检查程序

地址检测是被用来检测存储
地址选择逻辑的维护子程序

§1. 摘要

存储地址检测是被用来发觉在存储地址选择逻辑中错误操作的子程序。程序籍地址模型而行满存储地址，这模型的成功读回证明所有地址的存在。

§2. 机要要件

§2.1. DJS-130机的中央处理机逻辑。

§2.2. 读/写存储地址，参看第7节

§3. 开关设置

启动地址 = 0

§4. 操作步骤

§4.1 由二进制引入程序送入程序

§4.2 置开关到 000000

§4.3 按启动

§4.4 程序将运行，直到发现一个错误或手动停机为止，

§5. 程序输出/错误描述

§5.1 当错误被发觉时，程序将中止，致察 AC3 就得到出错的地址，致察 AC2 就得到从 AC3 指定的那个地址内读出数据。

§5.2 发生停机时进位链被置，存储地址出错在刚写进去的那个单元的被读出上。这暗示在出错单元处电流有问题。如果发生停机时进位链复位，单元出错是在已成功读了一次之后发生的，这暗示着其它地址可能影响着出错的地址。

§ 5.3 记录下每次打机的地址和数据，按“继续”试图去形成错误的类型，是否所有错误在一个 x 或 y 线上？出错的单元是否包含另一个地址的数据。

§ 6. 程序描述

程序包括比较简单的存貯和校对存貯四程序。从启动后第一次过程时，模型单元的地址被存入该模型单元，于是这单元被读出并校对其有否错误。程序继续下一个单元一直到模型的末尾。在过程的末尾，进位将取反。程序重新以读取和校对数据表扫描存貯四，但不存貯。

§ 7. 局限/附录

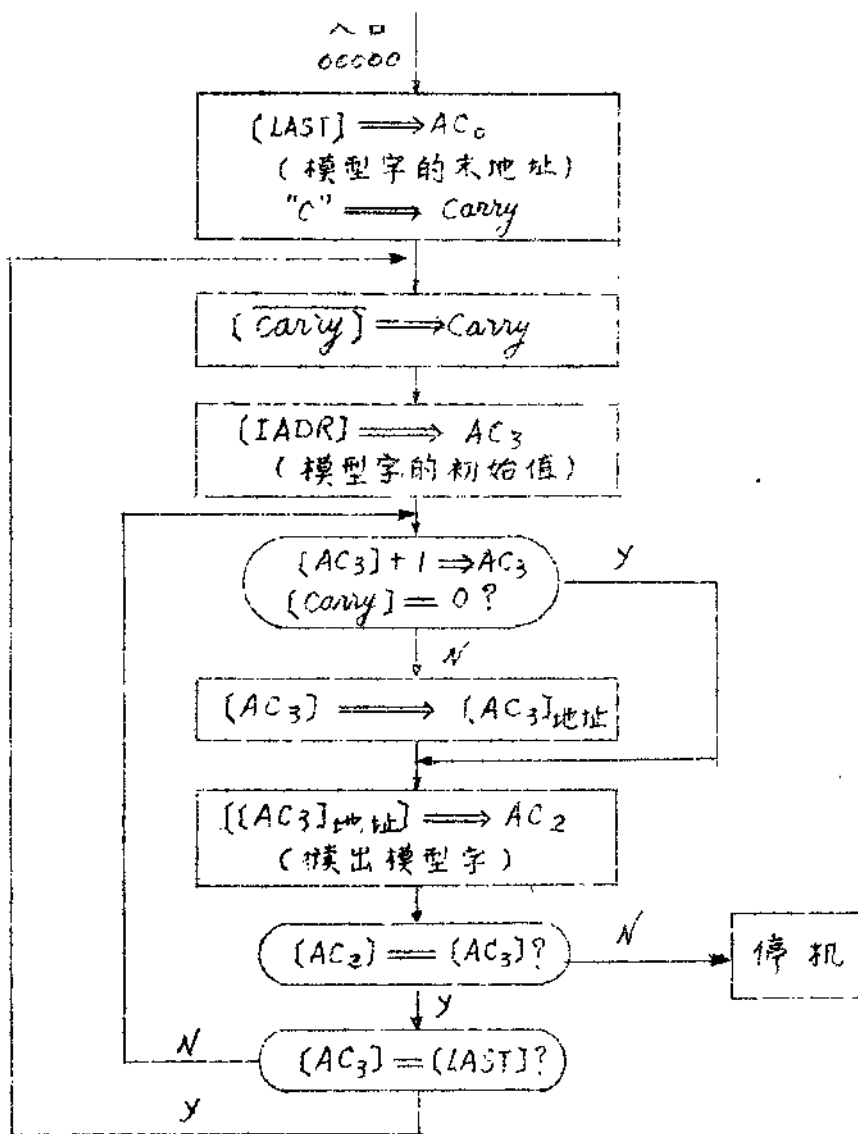
检测存貯四大小可以变更程序单元 14 号的数值来实现。

```

000000 • LOC
00000 020014  A:  LDA 中, LAST ; 这里启动
00001 101020      MOVE 中, 中
00002 101060      MOVC 中, 中
00003 034015      LDA 3, IADR ; AC3 = 初始型式
00004 175402  B:  INC 3, 3 SEC ; C(CARRY) = 1, 如果运
                                行游/校验
00005 055400      STA 3, 中, 3 ; C = 0, 如果只校验
00006 031400      LDA 2 中, 3
00007 156414      SUB #2, 3 SER ; AC2 = 出错字符
00010 063077      HALT          AC3 = 出错地址
00011 116414      SUB #中, 3, SER ; 检测是否达程末尾。
00012 000004      JMP B
00013 000002      JMP A+2
00014 007577  LAST: 7577          ; 型式字的末尾
00015 000015  IADR: •            ; 型式字的开头
                                • END
    
```

内存检查 I

地址检查程序框图部分



说明: 1. IADR 为模型字初始值的标志单元, 它必须取 $(IADR) \geq 15$ 。

2. LAST 为模型字末地址的标志单元, 必须取 $x7577$ 。

对于 4K 内存 $x=0$

8K 内存 $x=1$

⋮

32K 内存 $x=7$

内存检查II 程序说明及操作过程

一、简要说明：

该内存检查程序是采用在读出线上产生最坏布置下的干扰条件，以对存储器进行检测，本程序的正确运行，可以保证读出放大，禁止驱动和存取电流的正常操作。

二、所需的机口设备

1. DJS-130机的中央处理机
2. 4K 磁心存储器。

三、开关位置的设置：

1. 启动地址为：000002
2. 数据开关第0位为1，表示产生1024次读/写干扰。
3. 数据开关第15位为1，表示制止错误行机。

示意图如下

1		15
0 1 2		15
产生 读 写 干扰		制止 错误 行机

四、操作步骤：

1. 用二进制引导程序将本程序送入内存。
2. 置开关为000002
3. 按启动按钮，程序将执行。
4. 置开关第0位则产生1024次读/写干扰，置开关第15位则制止错误行机。

5. 程序允许修改如下单元。

(1) 模型起始地址 $C(3) = ADR$

注意：规定必须使 $C(3) \geq CEND$ ，否则将打机。

(2) 修改单元 $C(5) = INH$ 可以确定被屏蔽的二进位。

五. 出错显示：

1. AC_1 中存放错误字。

2. AC_2 中存放错误地址。

3. 如果电传机不忙碌，且开关第15位为0，则程序在“ER”位置打机。

4. 如果调整电流，可以置开关第15位，且按“继续”开关。

5. 观察同步脉冲：

模型字在存贮时发出P脉冲。

检测全“1”模型字发出S脉冲。

检测全“0”模型字发出C脉冲。

内存检查Ⅱ 程序部分

```

00002      • LOC 2

00002      000024      JMP BEGIN

3      007577      ADR:      CEND+1      ; 模型起始地址。
4      007577      FINAL:      7577      ; 模型终止地址。
5      177777      INH:      -1      ; 对禁止二进位屏蔽。
6      000000      PATI:      0      ; 模型字符。
7      000000      ERET:      0

00010      000017      C17:      17

1      000400      C400:      400
2      000077      C77:      77
3      001777      C777:      7777
4      000207      C207:      207
5      000101      C101:      101
6      060200      CNIOC:      NIOC 0      ; 发"C"脉冲指令常数。
7      060100      CNIOS:      NIOS 0      ; 发"S"脉冲指令常数

00020      070000      C70000      70000

1      000000      MODUAL:      0      ; 模存貯四单元
2      000000      EDIST:      0      ; 远极数存貯单元
3      063077      HALT      ; 模型起始地址在原程
                        序保护区则停机。
4      034003      BEGIN:      LDA 3, ADR ; 取一个模型初始
                        地址。
5      030020      LDA 2, C70000
6      020004      LDA 0, FINAL

00027      143400      AND 2,0
00030      040022      STA 0, EDIST
1      173400      AND 3,2
2      050021      STA 2 MODUAL ; 按4096取模存
                        入模存貯四:

```

```

00033 024156      LDA 1, CEND
      4 136033      ADCZ# 1, 3 SNC ;
      5 000023      JMP BEGIN-1 ; 模型起始地址校验错
                                误则停机
      6 034012      IPAT: LDA 3, C77 ; 模型打环开始。
      7 030003      LDA 2, ADR

00040 024011      LDA 1, C400
      1 020006      LDA 0, PATT ; 先给一个模型字。
      2 147404      AND 2, 1 SZR
      3 100000      IPAT1: COM 0, 0
      4 024005      LDA 1, INH
      5 123400      AND 1, 0 ; 屏蔽禁止二进制位。
      6 024010      LDA 1, C17
      7 060300      FILL: NIOP 0 ; 发"P"脉冲, 在062同步。
00050 041000      STA 0, 00, 2 ; 用模型字充满内存。
      1 151400      INC 2, 3 ; 模型地址加1。
      2 133414      AND# 1, 2 SZR ; 每16次跳。
      3 000047      JMP FILL
      4 157414      AND# 2, 3 SZR ; 每64次跳
      5 000043      JMP IPAT1
      6 020004      LDA 0, FINAL ; 测试本地址。
      7 142432      SUBZ# 2, 0 SEC ; 将4K内存, 按每64分组

00060 000040      JMP IPAT+2 ; 填满时间=100MS
      1 030021      DISTURB: LDA 2, MODVAL ; MODULE 选择干扰。
      2 020013      LDA 0, C7777 ; 单元干扰为0101
      3 024022      LDA 1, EDIST ; 0202, 0303---等
      4 123000      ADD 1, 0

```

00065	024015	LDA 1, C101 ; 存貯四内除对南线外
6	123000	ADD 1.2 ; 其它任何地址是最小干扰
7	074477	READS 3 ; 取开关数确定是否产生
00070	175112	MOVL# 3.3 SEC ; 1024次读写干扰.
1	142433	SUBZ# 2.0 SNC
2	000100	JMP ICHECK ; 干扰结束.
3	176400	SUB 3.3
4	025000	LDA 1.000, 2 ; 读存貯四.
5	175704	INCS 3.3 SZR
6	000074	JMP --2
7	000065	JMP DISTURB+4

00100	010003	ICHECK: LDA 2, ADR. ; 置检测周期起点.
1	024011	LDA 1, C400
2	020006	LDA 0, FATT ; 模型内的又线.
3	133414	AND# 1.2 SZR
4	100000	ICK: COM 0.0
5	034005	LDA 3, I11H
6	163400	AND 3.0 ; 屏蔽禁止的二进位.
7	024017	LDA 1, CNIOS ; 置"S"脉冲常数.
00110	114044	COMC 0.3 SZR ; 置"C"脉冲常数.
1	024016	LDA 1, CNIOC ; I/O 干扰信号.
2	044113	STA 1, CHECK
3	000000	CHECK: 0 ; 发出同步脉冲.
4	025000	LDA 1.000, 2 ; 信号PWV2, PWV1.
5	106414	SUB# 0, 1 SZR
6	004112	JSR ERR1
7	055000	STA 3.000, 2

00120	025000	LDA 1.000,2
1	136414	SUB# 1.3 SZR ; 信号 UV1, UV2.
2	004143	JSR ERR2
3	041000	STA 0.000,2
4	151400	INC 2,2
5	024010	LDA 1.C17 ; 计数16次
6	147414	AND# 2,1 SZR
7	000113	JMP CHECK
00130	034012	ECHECK: LDA 3.C77
1	157414	AND# 2,3 SZR ; 检测是否结束.
2	000104	JMP ICK ; 未满16次转入ICK.
3	004004	LDA 1.FINAL ; 每64次一跳.
4	146432	SUBZ# 2,1 SZR ; 对码的结束检测.
5	000101	JMP ICHECK+1
6	020006	LDA 0, PATT ; 模型字取反.
7	100000	COM 0. 0
00140	040006	STA 0, PATT
1	000024	JMP BEGIN
2	101020	ERR1: MOVZ 0,0 ; 干扰入口.
3	054007	ERR2: STA 3,ERET ; 不干扰入口.
4	101005	MOV 0,0 SNR ; C(carry)=1为多位.
5	101060	MOVC 0,0 ; C(carry)=0为丢位.
6	034014	LDA 3, C207 ; C(1)=错误字.
7	063411	SKPBN T10 ; C(2)=错误地址.
00150	075111	DOAS 3,TF0 ; 置开关为1 禁止错误停机
1	074477	READS 3,...
2	175213	MOVR# 3.3 SNC

~10~

00153	063077	ER:	HALT	；如果电传不忙碌则停机。
4	114040		COM	0.3
5	002007		JMP	@ ERET
6	000156	CEND:		

• END

内存检查Ⅲ

一、简要说明：

该内存检查程序是采用在读出/禁止线上产生最坏侮辱下的干扰条件，以对存储区进行检测，本程序的正确运行，可以保证读出放大器，禁止驱动区和存取电流的正常操作。

二、所需的机四设备

1. DJS-130机的中央处理机
2. 磁心存储区

三、开关位置的设置：

1. 启动地址为：000002
2. 数据开关第0位为1，表示产生1024次读/写干扰。
3. 数据开关第15位为1，表示制止错误行机。

四、操作步骤

1. 用二进制引导程序将本程序的目的程序送入内存。
2. 置开关为 000002
3. 按下启动按钮，则程序印出内存检查区域的最后一个单元的地址。
4. 如果出错是边缘的，置开关第0位，有助于引起错误的发生。
5. 当增大或调整电流时置开关第15位，将使错误出现时不行机，电传打字机印出“换行”符号。
6. 程序允许如下修改：

(1) 第3号单元存放模型字的起始地址： $C(3)=ADR$

(2) 第5号单元名称为 INHIBIT 标志在被清的二进位上，模型字该位禁止检测。

五、程序出错显示

1. 在每次错误出现时，如果电传打字机不忙碌，就打印出换行字符。如果数据开关第15位为0，程序打在“ER”

~12~

位置。

2. 在出错时机时

AC₁ 中存放错误字。

AC₂ 中存放错误地址。

AC₃ 中浮动检查程序的起始地址。

3. 如果要调整电流，可以置开关第15位，和按“继续”开关。

4. 观察同步脉冲

模型字在存貯时—发出 P 脉冲。

检测全“1”模型字—发出 S 脉冲。

检测全“0”模型字—发出 C 脉冲。

内存检查Ⅲ 程序部分

000002 • LOC2

00002 000164 JMP

00003 000620 ADR: 620 模型起始地址

00004 007577 FINAL: 7577 模型终止地址

00005 177777 INH: -1 对禁止二进制屏蔽

00006 000000 PADD: 0 模型字符

00007 000000 ERET: 0

000040 • LOC40

00040 000017 C17: 17

00041 000400 C400: 400

00042 000077 C77: 77

00043 007777 C7777: 7777

00044 000207 C207: 207

00045 000101 C101: 101

00046 060200 CN10C: N10C 0

00047 060100 CN10S N10S 0

00050	070000	C070000:	070000
00051	000000	MODUAL:	0
00052	000000	EDIST:	0
00053	001000	K1000:	1000
00054	177577	M201	-201
00055	000254	CMA	MMA2
00056	177625	PSIZE	BEGIN-CEND
00057	000000	PL0C:	0
00060	000000	BPROG:	0
00061	000000	EPROG:	0
00062	012345	RANDOM:	12345
00063	177760	M20:	-20
00064	000020	C20	20
00065	000432	CBEG:	BEGIN
00066	000000	RETURN:	0
00067	004126	SEND:	JSR,RAND;取一随机数Y _i
00070	000062		RANDOM
00071	024004		LDA 1, FINAL
00072	030056		LDA 2, 2PSIZE
00073	147000		ADD 2, 1
00074	101220		MOVZR 0, 0.
00075	122422		SUBZ 1, 0, SZC
00076	000075		JMP, .-1
00077	123000		ADD 1, 0. ; ACO = Y _i 模(FINAL)
00100	024063		LDA 1, M20
00101	123400		AND 1, 0.
00102	024003		LDA 1, ADR
00103	122433		SUBZ# 1, 0 SNC; 如果≤620 则用 C(ADR) 作为起始地址.


```

00104 121000 MOV 1.0 ;

00105 040057 STA 0.PL0C
00106 034063 LDA 3.M20
00107 117000 ADD 0.3.
00110 054060 STA 3.BPROG
00111 142400 SUB 2.0.
00112 040061 STA 0.EPROG
00113 145000 MOVE: MOV 2.1.
00114 030065 LDA 2.CBEG
00115 021000 LDA 0.0.2 ; 移动检测复版到被选点
00116 041420 STA 0.20.3
00117 175400 INC 3.3
00120 151400 INC 2.2
00121 125404 INC 1.1 SZR ; 测试对最后的KEG
00122 000115 JMP MOVE+2 ; 被移动
00123 034057 LDA 3.PL0C ; 程序出口
00124 005401 ISR 1.3
00125 000212 JMP MMA

00126 054160 .RAND. STA 3.UDΦ3 ; 生成一随机数X
00127 010160 ISX.UDΦ3
00130 044156 STA 1.UDΦ1
00131 050157 STA 2.UDΦ2
00132 031400 LDA 2.0.3
00133 021000 LDA 0.0.2
00134 004143 JSR .UD50
00135 034162 LDA 3.UD 20

```