

NOVA 1200

诊断程序

上海市电报局 1974年9月翻印

NOVA 1200 逻辑测试

程序

NOVA 1200 逻辑测试

纸带

二进制: 095-000036-00

摘要

NOVA 1200 逻辑测试是设计来作为 NOVA 1200 中央处理单元的维护程序。它用 NOVA 1200 指令系统来实现逻辑上的逐门测试。这套测试不含任何输入——输出或是配套设备。

097-000017-00

1. 摘要

NOVA 1200 逻辑测试是设计来作为 NOVA 1200 中央处理单元的维护程序，它用 NOVA 1200 指令系统来实现逻辑上的逐门测试。这套测试不含任何输入——输出或是配套设备。

2. 机口要求

2.1 NOVA 1200 中央处理机

2.2 4K 读存 存储

3. 置数开关

3.1 起始地址 400

4. 操作过程

4.1 用二进制数 000400

4.2 打开下按钮

4.3 按下中央处理机应该行止

5. 示错说明

5.1 用行止指令来指示错误

5.1 当折查表、钩筑、行可以重新形成错误的程序，折查逻辑。

6. 程序说明

这一程序是许多小程序循环的总合，每一循环都设计成能测试处理机逻辑的一部分，每一循环都设计成测试逻辑中的最少组成单元。

7. 附录

此程序通过的一遍的时间是极短的。

000000 .Loc4
 00000 063077 灯机
 00001 100000 (2)0
 00002 063077 灯机
 00003
 00004
 00005
 00006
 00007
 000121 .Loc121

000121 000000 PASS: 0 重复计数四
 000122 000000 KΦ: 0
 000123 177777 M1: -1
 000124 177776 M2: -2
 000125 177775 M3: -3
 000126 177774 M5: -4
 000127 177767 M9: 177767
 000130 100000 K10K: 100000
 000131 040000 K40K: 40000
 000132 020000 K20K: 20000
 000133 010000 K10K: 10000
 000134 00377 K377: 377

000135 000001 K1: 1
 000136 000003 K3: 3
 000137 000007 K7: 7
 000140 000017 K17: 17
 000141 000004 K4: 4
 000142 000010 K10: 10
 000143 000002 K2: 2
 000144 000020 K20: 20
 000145 0000200 K200: 200
 000146 167356
 000147 156735
 000150 135673
 000151 073567
 000152 010421
 000153 021042
 000154 042104
 000155 104210
 000156 001234
 000157 000420
 000160 010020
 000161 000123
 000162

在400开始, 不在这里。

00163	00421	061700	A06:	DIB P0, φ	到跳越触发口的与/或门。
00164	00422	063500		SKPBZ φ	DIB指令跳越, 检查 DAITB 的错
00165	00423	063077		HALT	误译码, IO 跳越应该不确定。
00166	00424	062700	A07:	DICP φ, φ	DIC指令跳越, 检查 PATIC 的错
00167	00425	063500		SKPBZ φ	误译码 IO 跳越应该不确定。
	00426	063077		HALT	
00400	00427	102400	A08:	SUB φ, φ	这些 "DOU" 指令, 不应该禁止 CPU
00401	00430	063076		DOC φ, 76	指令电平, 如果他们这样做了,
00402	00431	063075		DOC φ, 75	计算机将其主一条行机指令和
	00432	063073		DOC φ, 73	MBC1 φ — MBC15 触发口的与门。
	00433	063067		DOC φ, 67	
	00434	063057		DOC φ, 57	
	00435	063037		DOC φ, 37	
00403	00436	062677	A09:	IORST	ION 触发口应该清除 (见控制台上
00404	00437	063577		SKPBZ CPU	的灯); 应该跳越行机指令折查该
	00440	063077		HALT	脉冲的译码, CPU 指令应该
					阻止 (ION 禁止) 电平。
00405	00444	060377	A11:	NIOP CPU	送到运控的 "P" 脉冲不应该禁止
00406	00445	063577		SKPBZ CPU	ION 触发口, 检查脉冲译码
	00446	063077		HALT	口。
					"C" 脉冲不应该禁止 ION 触发口
	00447	060277	A12:	NIOP CPU	检查脉冲译码口
	00450	063577		SKPBZ CPU	
	00451	063077		HALT	
00411	00452	102040	A13:	APCO φ, φ	"ADC" 指令跳越, 检查通向跳越
00412	00453	063500		SKPBZ 0	触发口的异门 MBC15 位应为 "0",
00413	00454	063077		HALT	
00414	00455	102041	A14:	ADCO φ, φ, SKP	"ADC" 指令应该跳越, 折查通
00415	00456	063077		HALT	向跳越触发口的异门和与/或门
					等。
00416	00457	102021	A15:	ADCZ φ, φ, SKP	"ADC" 指令应该跳越, 折查与
00417	00460	063077		HALT	或门 (MBC14) (RYSZT.SAVE)
00420					MBC14 的输入端应该不是 "1" 见
					"跳越" 逻辑。

00520	103420	A26=	ANDZ, 0, 0, 0 ANDC, 0, 0, 5 NC HALT	进位应置位(0) 进位应补或1, 见前巨测试
00521	103463			
00522	063077			
00523	102620	A27=	SUBZR, 0, 0, 0 ANDZ, 0, 0, 5 ZC HALT	将AC中为1的位 清除并测试进位, 将查与5门 输入(SHL, ADDRESS)到形成 置CY的5/或门。
00524	103422			
00525	063077			
00526	102520	A28=	SUBZL, 0, 0, 0 ANDZ, 0, 0, 5 ZC HALT	置AC中为0的位 清除测试进位, 将查到产生 CRY SET的5/或门的5门 输入端(SHR, ACB11)
00527	103422			
00530	063077			
00531	103420	A29=	ANDZ, 0, 0, 0 MOV, 0, 0, 5 ZC HALT	进位应经过AND置位"0" 在MOV指令进行逻辑0, 将查在进位逻辑中通过5门 的5门(CRY OUT, AND)
00532	101002			
00533	063077			
00534	103421	A30=	ANDZ, 0, 0, 5 SKP MOV, 0, 0, 0 AND, 0, 0, 5 ZC HALT	AND指令清除进位有跳 越的MOV指令不应置位将查 输入回收进位的4输入5门(A LC, SKIP)
00535	101040			
00536	103412			
00537	063077			
00540	103440	A31=	ANDZ, 0, 0, 0 MOVZ, #21, 0, MOV, 0, 0, 5 NC HALT	由AND指令建立进位, 在MOV Z指令中MBC12应阻止进位 清除将查进位触发的C输入 端的4个输入5门在"MOVZ" 时ACRY应该不建立。
00541	101030			
00542	101003			
00543	063077			
00544	102000	A32=	ADC, 0, 0, 0 ANDZ, 0, 0, 5 ZC HALT	AC置位(-1)清除进位并交换位 果进位进位又被置"1"将查ZHL 或SHR的错误译码。
00545	103722			
00546	063077			
00547	034402	A32A=	LPA3, +2 MOV, 0, 0, 5 SKP HALT	LPA指令改变PC, 将查了SR 令的错误译码5门(CFA, HALT)
00550	101001			
00551	063077			
00552		A32B=	LPA3, K0 LDA2, K0 LDA1, K0	
00553	03022			
00554	02412			

00463	102177	A17=	APCC, #0, 0, 5 ON MOV, #0, 0, 0 SKPBZ CPU HALT	APC指令置ION触发, 折 查5门IO, E, AND, ZNAB, PULSZ ZNAB) IO, 关电平当 ADC指令时不应置真值
00464	101010			
00465	063577			
00466	063077			
00467	102400	A18=	SUB, 0, 0, 0 ANDZ, 0, 0, 0 ANDZ, 0, 0, 5 ZC HALT	测试0进位, 如果进位灯不亮 折查进位触发四输入端的5 门, 如进位灯亮折查进位触发 四输入。
00470	103420			
00471	103422			
00472	063077			
00473	103423	A19=	ANDZ, 0, 0, 5 NC MOV, 0, 0, 5 SKP HALT	进位为0, 这指令应该不跳越 折查在跳逻辑中MBC15的 5门。
00474	101001			
00475	063077			
00476	103440	A20=	ANDZ, 0, 0, 0 ANDZ, 0, 0, 5 ZC HALT	进位应该从1到0折查进位 灯折查产生CRY ZNAB(答 许进位)的5或门(CARRY, MBC11, CARRY MBC10) 到5门。
00477	103422			
00500	063077			
00501	103420	A21=	ANDZ, 0, 0, 0 ANDZ, 0, 0, 5 NC HALT	进位应该建立, 见控制台灯, 如灯亮折查在跳逻辑中的 5或门, 如果进位灯不亮折 查到产生CRY ZNAB的 XUR门与或门(进位, MBC11, 进位 MBC10)
00502	103443			
00503	063077			
00504	103440	A22=	ANDZ, 0, 0, 0 ANDZ, 0, 0, 5 NC HALT	进位进位, 见前巨对可能形成 错误的测试。
00505	103445			
00506	063077			
00507	103420	A23=	ANDZ, 0, 0, 0 AND, 0, 0, 5 ZC HALT	进位触发应该能进位进位 已被清除, 折查进位触发四 的输入端和或5门(进位 MBC11, 进位 MBC10)
00510	103402			
00511	063077			
00512	103440	A24=	ANDZ, 0, 0, 0 ANDZ, 0, 0, 5 NC HALT	进位触发应该能进位进位 已建立见前巨测试。
00513	103403			
00514	063077			
00515	103440	A25=	ANDZ, 0, 0, 0 ANDZ, 0, 0, 5 ZC HALT	进位应置位(1) 进位应补或0, 折查或5门 (进位, MBC11, 进位, MBC10)
00516	103462			
00517	063077			

清除A_C中和进位ADC指令置进位或在ADC指令的(PTG=中, T53)错误地加1, 折查加/逻辑的或5逻辑(ACC, IR7, IR7, KZY)

"ADC" 不应产生从加信田的进位如果在A_C中的结果是177776则ADC指令不做加补码而做加反码折查产生到加信田的(SZ)电平(ACC, IR6)

一条M₁₀指令没有启动脉冲, 不应该放动ION触发田。

一条倒CPU的着余用叫错误着(除ION)触发田折查(CLR ION)电平的译码和到ION触发田的CLR ION的输入端。

量和测试ION触发田如控制台ION灯不亮折查(SZT ION)使手的正确译码通向ION触发田的或门的累成态和清除除端如灯亮折查通向跳越逻辑(CPU INST, ION) OCIO和在跳越逻辑MBC 9位是白

当ION置位时"SKPBZ"指令跳越折查在跳越逻辑中MBC9的实现。

00555 020122
00556 103420
00557 102002
00560 063077

A33= LDA, 4, Kφ
AND, 4, φ
ADC, 4, 5, SEC
HALT

00561 020123
00562 024123
00563 030123
00564 034123
00565 102022
00566 063077

A34= LDA, 4, M1
LDA1, M1
LDA2, M1
LDA3, M1
ADC, 4, 4, SEC
HALT

00567 062077
00570 060377
00571 063577
00572 063077

A35= IORST
M10P CPU
SKPBZ
HALT

00573 060277
00574 063577
00575 063077

A36= M10 CPU
SKPBZ CPU
HALT

00576 060077
00577 063577
00600 063077

A37= M10 CPU
SKPBZ CPU
HALT

00601 060177
00602 060277
00603 063577
00604 063077

A38= M10S CPU
M10C CPU
SKPBZ CPU
HALT

00605 060177
00606 063477
00607 063077
00610 060277

A39= M10S CPU
SKPBZ CPU
HALT
M10C CPU

00611 060177
00612 063577
00613 103401
00614 063077

A40= M10S CPU
SKPBZ CPU
AND, 4, SKP
HALT

00615 060277
00616 060177
00617 063500
00620 063077

A41= M10C CPU
M10S CPU
SKPBZ, φ,
HALT

00621 060277
00622 060177
00623 063077
00624 063077

A42= M10C CPU
M10S CPU
SKPBZ CPU
HALT

00625 060277
00626 024122
00627 102022
00630 063077

A43= M10C CPU
LDA, 1, Kφ
ADC, 2, 4, 5, SEC
HALT

00631 030122
00632 102022
00633 063077

A44= LDA, 2, Kφ
ADC, 2, 4, 4, SEC
HALT

00634 020122
00635 024123
00636 030124
00637 034125
00640 106022
00641 063077

A45= LDA, 4, Kφ
LDA, 1, M1
LDA, 2, M2
LDA, 3, M3
ADC, 2, 1, 1, 2, 3, 4
HALT

00642 152022
00643 063077

A46= ADC, 2, 3, SEC
HALT

00644 126005
00645 063077

A47= ADC, 3, 3, SNR
HALT

00646 176400
00647 175043
00650 063077

A48= SUB, 3, 3,
M0V, 0, 3, 3, SNC
HALT

00651 175000
00652 175005
00653 063077

A49= ADC, 3, 3, 3,
MOV, 3, 3, SNR
HALT

车跳越指令测试SZLB线不是ION触发田, 当CPU指令未出现时ION(CPU, INST, ION)将SZLB线接地。

ION处于置位不应影响的SZLB线的测试或5门(SZLB, MBC8, SZLD)

ACS应该指令(-1) ACI为0, 如果"ADC"指令选择的反ACI不是零形成一宁进位折查ACS 2选择电平尾AD选择等。

ACS指令(-1)除了ACI和AC2, 如果ALC指令选择的反ACI不是0, 它可以形成一宁进位折查ACS 1选择电平, 反AC选择等。

ACI = 0 所有其余(-1), 如果ADC指令选择反ACI代替ACI 反会形成一宁进位折查ACS 1选择电平(在ALC指令)也折查反AC存储。

ACS SZL电平失效以前测试。

折查加反的及码, 应能产生一宁进位跳越逻辑与或输入和MBC 3。

0 传是不应影响进位跳越逻辑的折查, 折查到跳越逻辑的5或0。

ADC指令在A3中产生177777, 如果"MOV"中IR 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000

ACφ 应送入 0 并且按 +1 如
果送入 A01 指令也送入 0, 则
INC 指令将置进位折查 ACSZ
SZL 电平 SYNC SCφZ
ON LDA 和 PACK

如果 ACZ 指令将 AC 代入 0, 则
INC 指令将置进位在第二条
"LDA" 指令折查 ACS1 选择
电平 SYNC SCφZ 在 LDA
和 PACK

如果送入 ACφ 实际上送入
AC1, INC 指令将置进位折查
折查 LDA 的表 AC 选择。

如果送入 ACφ 实际上送入
AC2, INC 指令将置进位折查
LDA 的表 AC 选择。

ACφ 应包含 100000 MOVZ
指令应该跳越折查 0 跳越遇
辑中移位 0 输入到 4 输入。
折查移位位在 0 跳越逻辑中移
位 1 输入到 4 输入或门。
折查在 0 跳越逻辑中移位 2
输入到 4 输入或门。
折查在 0 跳越逻辑中移位 3
输入到 4 输入或门。
0 存储触发 0 应记住 AC 差
不全 0 这样 AOVZ 应该跳
越折查 0 存储置位到跳越遇
辑力一端的与门 11

ACφ 应该读输入 0 并且加到 #
折查在 INC 指令中在 PFG
TS3 时保证 +1.

从 +1 加到 +2
从 +3 加到 +4
从 +7 加到 +10
从 +17 加到 +20 加的结果是
+φ, 不是 +2φ, 折查 5 进 (PFG
= φ, 7SZφ, PFG = φ, TS3 串
行) 在加 1 逻辑中, 也折查串
行 CRY 触发 0.

(-1) 将加 (-1) 的生成 (-2) 和
一进位折查 AND 是 (AND, CRY
CUT) 到产生 CRY ZNAB 的
置 10.

折查产生 CRY-ZNAB 的置
10, 见以前的测试。
折查 5 进 (AND, CRY OUT) 进
位应该是 0.

ACφ = -1 其变 ACS = φ 折查
保证用的 AC 在 MOV 被选

00654	021022	A50=	LDA, φ, Kφ LDA, 1, M1 INCZ, φ, φ, SZC HALT
00655	024123		
00656	101422		
00657	063077		
00660	020122	A51=	LDA, φ, Kφ LDA, 2, M1. INCZ, φ, φ, SZC HALT
00661	030123		
00662	101422		
00663	063077		
00664	024122	A52=	LDA, 1, Kφ, LDA, φ, M1, INCZ, 1, SZC HALT
00665	020123		
00666	101422		
00667	063077		
00670	030122	A53=	LDA, 2, Kφ LDA, φ, M1 INCZ, 2, SZC HALT
00671	020123		
00672	151422		
00673	063077		
00674	020130	A54=	LDA, φ, K1φφK MOV, φ, φ, SNR HALT
00675	101005		
00676	063077		
00677	020131	A55=	LDA, φ, K4, φK MOV, φ, φ, SNR HALT
00700	101005		
00701	063077		
00702	021032	A56=	LDA, φ, K2φK MOV, φ, φ, SNR HALT
00703	101005		
00704	063077		
00705	020133	A57=	LDA, φ, K1φK MOV, φ, φ, SNR HALT
00706	101005		
00707	063077		
00710	020134	A58=	LDA, φ, K377 MOV, φ, φ, SNR HALT
00711	101005		
00712	063077		
00713	020122	A59=	LDA, φ, Kφ, INC, φ, φ, SNR HALT
00714	101405		
00715	063077		
00716	020135	A60=	LDA, φ, K1 INC, φ, φ, SNR HALT
00717	101405		
00720	063077		
00721	020135	A61=	LDA, φ, K3 INC, φ, φ, SNR HALT
00722	101405		
00723	063077		
00724	020137	A62=	LDA, φ, K7 INC, φ, φ, SNR HALT
00725	101405		
00726	063077		
00727	020140	A63=	LDA, φ, K17 INC, φ, φ, SNR HALT
00730	101405		
00731	063077		
00732	020123	A64=	LDA, φ, M1 LDA, 1, M1 LDA, 2, M1 LDA, 3, M1 ADDZ, φ, φ, SZC HALT
00733	024123		
00734	030123		
00735	034123		
00736	103033		
00737	063077		
00740	103052	A65=	ADDZ, φ, φ, SZC HALT
00741	063077		
00742	103432	A66=	ANDZ, φ, φ, SZC HALT
00743	063077		
00744	024122	A67=	LDA, 1, Kφ LDA, 2, Kφ LDA, 3, Kφ LDA, φ, M1 MOV, φ, 1 MOV, #, 1, SNR HALT
00745	00745		
00746	00746		
00747	00747		
00750	00750		
00751	00751		
00752	00752		

检查保证在第一套 NoV 指令保
证目的 A02 被选。

A68: MOV #, 2
MOV #2, SNR
HALT

测试保证 ADD 用目的 AC, 折查
在 DISABLERD 多路开关的与效
应 (A'L.C, I.R.S)

A69: LDA #, K #
LDA I, M1
ADD #, I, SNR
HALT

$\phi + 1 = 1$,
折查目的 AC
缓冲十多路传输)

A70: LDA #, K #,
LDA I, K1
ADD #, I, SNR
HALT

见前
 $\phi + 2 = 2$

A71: LDA #, K #
LDA I, K2
ADD #, I, SNR
HALT

见前
 $\phi + 4 = 4$

A72: LDA #, K #
LDA I, K4
ADD #, I, SNR
HALT

见前
 $\phi + 1 \phi = 1 \phi$

A73: LDA #, K #
LDA I, K1 #
ADD #, I, SNR
HALT

把 (-1) 加到 (+) 不应该产生世
信或让 ADC 输出在 TS 3 失效而
且 MAC 而不是目的 AC 拍入。

A74: LDA #, M1,
LDA I, K #
ADD #, I, SEC
HALT

ADC 置 AC # 到 (-1), 这 "SUB" 应
该不改变 CCAC #, 因为空已被
跳越 LOP CRY 应该阻止 PACK 和
AC 缓冲, 折查 PACK 触发输入。

A75: ABC #, #, SKP
SUB #, #,
MOV #, #, SNR
HALT

折查通过 ADC 指令折查进入
AC #

A76: LDA #, K #
ADC #, #,
MOV #, #, SNR
HALT

折查用 LDA 指令折查进入 AC #
折查到 PACK 触发四的或门。

A77: SUB #, #,
LDA #, M1
MOV #, #, SNR
HALT

任意做减去本身应产生 # 如果
CCAC # = # 折查到跳越逻辑的
与或门 (MBC13, 0) 如 CCAC #
PS 不 = 0, 扫阔 ACB 寄存器, 加信
四等。

A78: LDA #, K #
SUB #, #, SER
HALT

见前已测试

A79: LDA #, M1
SUB #, #, SER
HALT

有一条 "SUB" 应该不跳越 # = 条
应该

A80: LDA #, M1,
SUB #, #, SNR
SUB #, #, SER
HALT

ADC 将各位置 "1" COM 则拍其变
补, 然后皆由 0

A81: ADC #, #,
COM #, #, SER
HALT

折查 MOV 会拨动各位

A82: SUB #, #,
MOV #, #,
MOV #, #, SER
HALT

折查全 0 从 AC # 移到 AC2,

A83: SUB #, #,
LDA I, M1,
LDA #, M1,
LDA #, M1
MOV #, #,
APP #, #, SER
HALT

折查全 0 从 AC # 到 AC3 # 的 AC
能否保持全 0

A84: SUB #, #,
MOV #, #,
APP #, #, SER

01055	063077	HALT	折查全0从Ac4移到Ac中,目的AC 转变保持全0.
01056	102400	ASB= SUB φ, φ	
01057	101000	MOV φ, φ	
01060	103004	ADD φ, φ, SZR	
01061	063077	HALT	
01062	102400	ASB= SUB φ, φ	
01063	105000	MOV φ, 1	
01064	107004	ADD 1, 1, SZR	
01065	063077	HALT	
01066	020122	A87= LDA φ, K φ	测试1 的从AC3到AC中的移位
01067	024122	LDA 1, K φ	折查目的AC 能保持1.
01070	030122	LDA 2, K φ	
01071	176000	ADC 3, 3	
01072	161000	MOV 3, 3	
01073	116404	SUB φ, 3, SZR	
01074	063077	HALT	
01075	176000	A88= ADC 3, 3	见前页的测试
01076	165000	MOV 3, 1	
01077	126404	SUB 1, 3, SZR	
01100	063077	HALT	
01101	176000	A89= ADC 3, 3	见前页的测试
01102	171000	MOV 3, 2	
01103	156404	SUB 2, 3, SZR	
01104	063077	HALT	
01105	102000	A90= ADC φ, φ	见前页的测试
01106	115000	MOV φ, 3	
01107	162404	SUB 3, φ, SZR	
01110	063077	HALT	
01111	102000	A91= ABC φ, φ	任意加法及其所得(-1) (-1)
01112	100004	COM φ, φ, SZR	
01113	063077	HALT	
01114	126000	A92= ADC 1, 1	见上页的测试
01115	124004	COM 1, 1, SZR	
01116	063077	HALT	

01117	152000	A93= ADC 2, 2	见上页的测试
01120	150004	COM 2, 2, SZR	
01121	063077	HALT	
01122	176000	A94= ADC 3, 3	见前页的测试
01123	174004	COM 3, 3, SZR	
01124	063077	HALT	
01125	176000	A95= ADC 3, 3	将AC3置成(-1) 然在将它在
01126	161404	INC 3, φ, SZR	(+φ), 折查加进位。
01127	063077	HALT	
01130	176000	A96= ADC 3, 3	(-) 增加应该产生(+0)和进
01131	161404	INC 3, φ, SMC	位。
01132	063077	HALT	
01133	176000	A97= ADC 3, 3	置C(AC3)=17777
01134	102400	SUB φ, φ	C(ACφ)=φφφφφφ
01135	105400	INC φ, 1	φ+1=φφφφφφ
01136	137014	ADD #1, 3, SZR	(-1)+(-1) = φφφφφφ
01137	063077	HALT	
01140	102020	A98= ADC Z φ, φ	MISC测试进位
01141	101402	INC φ, φ, SZR	
01142	063077	HALT	
01143	102700	A99= SUB 0, φ, φ	减法应该产生全0, 交换一半半
01144	101004	MOV φ, φ, SZR	位AC缓冲寄存AC中指令的每一位
01145	063077	HALT	身对于折查在减指令中送入
01146	102340	A100= ADC 0, φ, φ	AC缓冲寄存1, 交接一半半也
01147	104004	COM φ, 1, SZR	位AC缓冲寄存口接管的每一位
01150	063077	HALT	折查折查在AC缓冲寄存的各位
01151	102001	A101= ADC φ, φ, SKP	"ADC"的"MOV"应该执行, SUB指
01152	102401	SUB φ, φ, SKP	令应该不推跳越, 折查测试跳
01153	101001	MOV φ, φ, SKP	越触发口的输入。
01154	063077	HALT	

交换命令错误译成左移或是右移
这样就是位。

B00: ADC 2, 0, 0, SEC
HALT

进位的交替测试。

B01: ADC 0, 0, 0, 0
SUB 0, 0, 0, 0, SEC
HALT

0和0相加

B02: SUB 0, 0, 0, 0
AND 0, 0, 0, 0, SEC
HALT

(-1) 与 (-1) 应产生 (-1)

B03: ADC 0, 0, 0, 0
AND 0, 0, 0, 0, 0
COM 0, 0, 0, 0, SEC
HALT

(-1) 与 (0) 应产生 0

B04: ADC 0, 0, 0, 0
SUB 1, 1, 1, 1
AND 0, 0, 0, 0, SEC
HALT

(0) 与 (-1) 应产生 0

B05: ADC 0, 0, 0, 0
SUB 1, 1, 1, 1
AND 1, 0, 0, 0, SEC
HALT

0 与 0 应该产生 0

B06: SUB 0, 0, 0, 0
AND 0, 0, 0, 0, SEC
HALT

0 变量仍应为 0

B07: SUB 0, 0, 0, 0
NEG 0, 0, 0, 0, SEC
HALT

(-1) 变量应该形成 +1 在 AC1

B08: ADC 0, 0, 0, 0
NEG 0, 0, 0, 0, 1
ADD 0, 0, 0, 0, 1, SEC
HALT

测试在左移时 AC0B15 到移位 2
的输入正确结果 = 2

B09: LDA 2, K1
MOV 2, 2, 3, SNR
HALT

测试在左移时从 AC0B14 到 1 的
输入正确结果 = 4

B10: LDA 2, K2
MOV 2, 2, 2, SNR
HALT

测试在左移时从 AC0B13 移到 0
的输入。

B11: LDA 2, K4
MOV 2, 2, 2, SNR
HALT

测试在右移时从 AC0B14 移到 3
的输入正确结果 = 1

B12: LDA 2, K2
MOV 2, 2, 2, SNR
HALT

测试在右移时从 AC0B14 移到 2
的输入, 正确的结果 = 2

B13: LDA 2, K4
MOV 2, 2, 2, SNR
HALT

测试在右移时从 AC0B12 移到 1
的输入。

B14: LDA 2, K10
MOV 2, 2, 2, SNR
HALT

左移应该在进位触发 0 中置 0,
折查产生置 CRY 电平 的 5/或 0
折查 (SNR, ACB11) 输入。

B15: SUB 0, 0, 0, 0
MOV 2, 2, 2, SNR
HALT

右移应该在进位触发 0 中置 0,
折查产生置 CRY 电平 的 5/或 0
折查 (SNR, ACB11) 的输入。

B16: SUB 0, 0, 0, 0
MOV 2, 2, 2, 2, SEC
HALT

左移应该在进位触发 0 中置 0,
折查产生置 CRY 电平 的 5/或 0
SHL 不禁止 TOP 和 (CRY, ENAB,
SHR, SHL)

B17: SUB 0, 0, 0, 0
MOV 0, 0, 0, 0, SEC
HALT

右移应该在进位触发 0 中置 0,
折查产生置 CRY 电平 的 5/或 0
SHR 电平不禁止 TOP 和 (CRY,
ENAB, SHL, SHR)

B18: SUB 0, 0, 0, 0
MOV 0, 0, 0, 0, SEC
HALT

测试在右移时从 AC0B15 到移位 2
折查产生置 CRY 的 5/或 0 输入
(SHL, AND, OR, 输入)

B19: ADC 0, 0, 0, 0
MOV 2, 2, 2, SNR
HALT

01254	102000	B20:	AC0, 0, 0, MOVZL, 0, 0, SNC HALT	右移应该在进位触发器中设置 "1" 折查产生置CRY的5/或门, 折查(SHR, ACB11)的输入。	01314	101120	B20:	MOVZL, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01255	101223	B21:	SUBZL, 0, 0, SNR HALT	此指令应在AC0中置(+), 折 查左移到了的输入, 5/或门, CRY ENAB, SAVE等。	01315	106414	B21:	LDA, 0, K2 MOVZL, 0, 1, ADD, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01256	063077	B22:	SUBOL, 1, 1, MOV, 1, 2, SZC HALT	SUBOL应该在AC1中置0, 如 果15位置1, 折查5/或门(CRY, ENAB, SAVE, PTG=0, TSD) 左移到了的输入。	01316	063077	B22:	LDA, 0, K4 MOVZL, 0, 1, ADD, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01257	102525	B23:	LDA, 0, M2 MOVOL, 0, 0, COM, 0, 1, SNR HALT	(移2在MOV指令从ACB15 位左端输入会拔动一位。	01317	020143	B23:	LDA, 0, K4 MOVZL, 0, 1, ADD, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01260	063077	B24:	LPA, 1, M3 MOVOL, 1, 1, COM, 1, 2, SNR HALT	(移1在MOV指令从ACB14位左 端输入会拔动一位。	01320	105120	B24:	LDA, 0, K4 MOVZL, 0, 1, ADD, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01274	020126	B25:	LDA, 2, M5 MOVOL, 2, 2, COM, 2, 3, SNR HALT	(移0)在MOV指令ACB13位左 端输入会拔动一位。	01321	103000	B25:	LDA, 0, K4 MOVZL, 0, 1, ADD, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01275	151140	B26:	LDA, 3, K1 MOVZL, 3, 3, SNR HALT	10左移应该是20, 而不是 0, 折查左移到了的输入。	01322	106414	B26:	LDA, 0, K4 MOVZL, 0, 1, ADD, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01276	154005	B27:	SUBOL, 0, 0, SZR HALT	AC0应该是0折查到移位地 平的左输入端。	01323	063077	B27:	LDA, 0, K4 MOVZL, 0, 1, ADD, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01277	063077	B28:	ADD, 3, 3, SUBZL, 0, 0, ADD #0, 3, SZR HALT	SUBZL指令应该在AC0中 产生+1对-1加1应该等于0。	01331	020142	B28:	LDA, 0, K4 MOVZL, 0, 1, ADD, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01303	102544	B29:	LDA, 0, K1 LDA, 1, K1 ADD, 1, 1, ADD, 1, 1,	ADD有效地完成在移折查在 MOV指令的左端。	01332	105120	B29:	LDA, 0, K4 MOVZL, 0, 1, ADD, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01304	063077	B30:	LDA, 3, K2 MOVZL, 3, 3, SNR HALT	折查(CAC0)=0, 折查AC0包含 15/0/0折查5/或门(CRY ENAB, END, 0, 0, LE)和(移0)右输入。	01333	103000	B30:	LDA, 0, K4 MOVZL, 0, 1, ADD, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01305	176000	B31:	LDA, 3, K2 MOVZL, 3, 3, SNR HALT	折查(CAC0)=0, 折查AC0包含 15/0/0折查5/或门(CRY ENAB, END, 0, 0, LE)和(移0)右输入。	01334	106414	B31:	LDA, 0, K4 MOVZL, 0, 1, ADD, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01307	117014	B32:	LDA, 3, K2 MOVZL, 3, 3, SNR HALT	折查(CAC0)=0, 折查AC0包含 15/0/0折查5/或门(CRY ENAB, END, 0, 0, LE)和(移0)右输入。	01335	063077	B32:	LDA, 0, K4 MOVZL, 0, 1, ADD, 0, 0, SUB #0, 1, SZR HALT	见上面的测试
01310	063077	B33:	SUBZL, 0, 0, SZR HALT	折查(CAC0)=1, 5/0/0/0, 折查0, 右输入的和/或门。	01336	102400	B33:	SUB, 0, 0, NZGS, 0, 0, SZR HALT	交错的对试来看进位是 是0/或门。
01311	020155	B34:	LDA, 3, K2 MOVZL, 3, 3, SNR HALT	折查(CAC0)=0, 折查AC0包含 15/0/0折查5/或门(CRY ENAB, END, 0, 0, LE)和(移0)右输入。	01337	100704	B34:	SUBZL, 2, 2, MOV, 2, 3, SUB, 2, 3, SNR HALT	测试MOVE交换实际引起一 次交换, C(AC2)应该等于+1 C(AC3)应该等于+0/0。
01312	020135	B35:	LDA, 3, K2 MOVZL, 3, 3, SNR HALT	折查(CAC0)=0, 折查AC0包含 15/0/0折查5/或门(CRY ENAB, END, 0, 0, LE)和(移0)右输入。	01340	063077	B35:	SUBZL, 0, 0, SZR HALT	折查(CAC0)=1, 5/0/0/0, 折查0, 右输入的和/或门。
01313	127000	B36:	LDA, 3, K2 MOVZL, 3, 3, SNR HALT	折查(CAC0)=0, 折查AC0包含 15/0/0折查5/或门(CRY ENAB, END, 0, 0, LE)和(移0)右输入。	01341	152520	B36:	SUBZL, 0, 0, SZR HALT	折查(CAC0)=0, 折查AC0包含 15/0/0折查5/或门(CRY ENAB, END, 0, 0, LE)和(移0)右输入。

01357	01360	01361	B39	152600	153014	063077	SUBZER 2, 2, ADD # 2, 2, SZR HALT	置 C(AC2)=100000 为 1 位应该 形成 0 和进位	01403	063077	HALT	触发四逻辑中(I.O, E, AND, ENAB, IR7) 与门。	
01362	01363	01364	B40	620130	105000	107000	LDA #K10, #K MOV #1, ADDR 1, 1, SUB #1, 1, SZR HALT	测试 AND 及右移加法实际上是左 移, 太移命令恢复瓦值。	01424	102000	B47	ABC #, #, DIB #, #, COM # #, #, SZR HALT	DIB 指令应该交换 AC 中的内 容, PACK 错置路由与门(I.O, E, AND, ENAB, IR7, (IRSCPU))
01365	01366	01367	B41	024131	131000	153220	LDA 1, K4 #K MOV 1, 2, ADDR 2, 2, SUB #1, 2, SZR HALT	见前页的测试	01430	102000	B48	ADC #, #, DIB #, #, CPU COM # #, #, SZR HALT	INTA(DIB-CPU) 错接 C(AC#) 折查在 PACK 触发四逻辑中的 与门(IRSCPU, INST) 如果配合 它的与门阻止 PACK 置零。
01371	01372	01373	B42	030132	155000	172000	LDA 2, K2 #K MOV 2, 3, ADDR 3, 3, SUB #2, 3, SZR HALT	见前页的测试	01434	102000	B49	ADC #, #, DIB #, #, COM # #, #, SZR HALT	DIC 指令应置 PACK 触发四差 且存入 AC 中, 与门(IRSCPU, INST) 没有(CPU INST) 电率配 合, 就禁止 PACK
01374	01375	01376	B43	034133	161000	183220	LDA 3, K1 #K MOV 3, # ADDR #, #, SUB # #, #, 3, SZR HALT	见前页的测试	01440	102000	B50	ADC #, #, NIO #, COM # #, #, SZR HALT	NIC 指令不应交换 C(AC#) 因 为 IR7 不是 1 折查通向 PACK 触发四的 4 输入端与门(I.O, E, AND, ENAB, IR7 等)
01400	01401	01402	B44	034403	171120	145200	LDA 3, #3, MOV 2, 1, SUB #1, 3, SZR HALT	交错移位测试	01444	102000	B51	ABC #, #, NEG #, #, #, COM # #, #, SZR HALT	"NEG" 指令不应交换 C(AC#) 折查到 PACK 触发四的输入 4 输入端与门(I.O, E, AND, ENAB, IR7 等) I.O, E 不在则此与门不 应组合。
01403	01404	01405	B45	102500	101000	105000	LDA 3, #3, MOV 2, 1, SUB #1, #, SZR HALT	交错移位测试	01445	100410	B52	SUBZL #, #, MOV #, #, MOV #, 1, SUB #1, #, SZR HALT	交错的折查 AC# = +1 AC1 应 = +1 减信的结果应设为 0
01406	01407	01410	B46	102500	176000	063600	SUBZL #, #, ADC 3, 3, SKPDN, #, ADD # #, #, SZR HALT	将 AC 0 置 +1 AC 3 减 -1, 7% 跳越指 令不应该改变 AC, 因为 AND, ENAB 阻止 PACK 触发四置位 见 PACK 触发四逻辑 4 输入与 门。	01450	102500	B53	LDA #, #, MOV #, 1, LDA #, #, SUB # #, #, SZR HALT	LDA 指令是否送入同样数据。
01411	01412	01413	B47	102500	176000	063600	SUBZL #, #, ADC 3, 3, SKPDN, #, ADD # #, #, SZR HALT	将 AC 0 置 +1 AC 3 减 -1, 7% 跳越指 令不应该改变 AC, 因为 AND, ENAB 阻止 PACK 触发四置位 见 PACK 触发四逻辑 4 输入与 门。	01455	105000	B54	LDA #, #, MOV #, #, LDA #, #, SUB # #, #, SZR HALT	二条 LDA 指令应该送入同样的 数据, 如果 ENAB 触发四置位 21
01414	01415	01416	B48	102500	176000	063600	SUBZL #, #, ADC 3, 3, SKPDN, #, ADD # #, #, SZR HALT	将 AC 0 置 +1 AC 3 减 -1, 7% 跳越指 令不应该改变 AC, 因为 AND, ENAB 阻止 PACK 触发四置位 见 PACK 触发四逻辑 4 输入与 门。	01460	106414	B55	LDA #, #, MOV #, 1, LDA #, #, SUB # #, #, SZR HALT	二条 LDA 指令应该送入同样的 数据, 如果 ENAB 触发四置位 21
01417	01420	01421	B49	102500	176000	063600	SUBZL #, #, ADC 3, 3, SKPDN, #, ADD # #, #, SZR HALT	将 AC 0 置 +1 AC 3 减 -1, 7% 跳越指 令不应该改变 AC, 因为 AND, ENAB 阻止 PACK 触发四置位 见 PACK 触发四逻辑 4 输入与 门。	01462	020000	B56	LDA #, #, MOV #, #, LDA #, #, SUB # #, #, SZR HALT	二条 LDA 指令应该送入同样的 数据, 如果 ENAB 触发四置位 21
01422	01423	01424	B50	102500	176000	063600	SUBZL #, #, ADC 3, 3, SKPDN, #, ADD # #, #, SZR HALT	将 AC 0 置 +1 AC 3 减 -1, 7% 跳越指 令不应该改变 AC, 因为 AND, ENAB 阻止 PACK 触发四置位 见 PACK 触发四逻辑 4 输入与 门。	01463	024000	B57	LDA #, #, MOV #, #, LDA #, #, SUB # #, #, SZR HALT	二条 LDA 指令应该送入同样的 数据, 如果 ENAB 触发四置位 21

01414 106414 SUB#0,1, SZR
01465 063077 HALT

01466 176400 B55= SUB, 3, 3,
01467 152400 SUB, 2, 2,
01470 020401 LDA, 1, 1,
01471 020401 LDA, #, +1,
01472 106415 SUB#0, 1,
01473 063077 HALT

01474 176400 B56= ADC 3, 3,
01475 152400 ADC 2, 2,
01476 020401 LDA 1, 1,
01477 021001 LDA, #, 1, 2,
01500 106415 SUB#0,1,SNR
01501 063077 HALT

01502 102400 B57= SUB #, #,
01503 126400 SUB 1, 1,
01504 152400 SUB 2, 2,
01505 176400 SUB 3, 3,
01506 020402 LDA #, +2,
01507 025001 LDA, 1, 2,
01510 106415 SUB#0,1,SNR
01511 063077 HALT

01512 102400 B58= SUB #, #,
01513 126400 SUB, 1, 1,
01514 152520 SUBZL 2, 2,
01515 176520 SUBZL 3, 3,
01516 025400 LDA 1, #, 3,
01517 020110 LDA #, #,
01520 106415 SUB#0,1,SNR
01521 063077 HALT

01522 102400 B59= SUB #, #,
01523 126400 SUB 1, 1,
01524 152400 SUB 2, 2,
01525 176500 SUBZL 3, 3,
01526 021400 LDA, #, #, 3,
01527 021600 LDA, 1, #,
01530 106415 SUB#0,1,SNR

当EFA时间接地那送入的假将是
LDA指令本身折查与门(EFA, IR6,
IR7)到(ED15 ABZLEP / 100LT)

测试 LDA 指令的有效地址。如果在
第 2 条 LDA 指令的 EFA 时间 (D15
ABZLE P 0 M ULT) 是 +, 应该从位置
1 而不是现行地址 +1 折查与/或门
(EFA, IR6, IR7) IR7 的输出。不
应该禁止与门。

第 2 条 LDA 指令从位置 1 取假, 不
是从 C(Ac2) 用 +1 代替。见前已的
测试。

第 1 条 LDA 指令的变址在 MA (程序
计数) 是 2。如果在 AC2 所有的 ACS
都将是折查与门。如果用了 MA, 此测试
(EFA, IR6) 产生 (AC0 017) 在 EFA
的 SYNC scope

第 1 条 LDA 指令将 AC3 用作变址
地址。第 2 条是 AC1, 折查与门。第 3 条
是折查与门。第 4 条是 LDA 指令在 EFA
射 (ACD3 SEL) 电平

第 1 条 LDA 指令将 AC2 用作变址
地址。第 2 条是 AC3, 折查与门。第 3 条
是折查与门。第 4 条是 LDA 指令在
(ACD4 SEL) 的与/或门。

01531 063077

01532 102400 B60= SUB #, #,
01533 126400 SUB 1, 1,
01534 152520 SUBZL 2, 2,
01535 176400 SUB 3, 3,
01536 021000 LDA #, #, #,
01537 020400 LDA 1, #,
01540 106415 SUB #, #, 1, SNR
01541 063077 HALT

01542 152400 B61= SUB 2, 3,
01543 176400 SUB 3, 3,
01544 021000 LDA #, #, 2,
01545 025400 LDA 1, #, 3,
01546 106414 SUB#0,1, SZR
01547 063077 HALT

01550 020122 B62= LDA #, #, #,
01551 105604 MOV #, #, #, SZR
01552 063077 HALT

01553 020123 B63= LDA #, #, #,
01554 104004 COM #, #, #, SZR
01555 063077 HALT

01556 020135 B64= LDA #, #, #,
01557 024103 LDA 1, #, #,
01560 107014 ADD #, #, #, SZR
01561 063077 HALT

01562 020402 B65= LDA #, #, #,
01563 024401 LDA 1, #, #,
01564 106414 SUB #, #, #, SZR
01565 063077 HALT

01566 020402 B66= LDA #, #, #,
01567 101001 MOV #, #, #, SZR
01570 000000 #
01571 101004 MOV #, #, #, SZR
01572 063077 HALT

一条 LDA 指令在 EFA 时, 折查与
地址的是 AC3 而不折查与门。
第 2 条 LDA 指令在 EFA 时, 折查与
地址的是 AC3 而不折查与门。
AC2 作为变址和果 (ACD, 4, 5)
AC2 作变址和果 (ACD, 4, 5)

AC2 和 AC3 假被置 0, 一条用假
变址的 LDA 指令与用 AC2 作
变址的指令得到同样的结果。
查在 JS3 时间经过 IR7 或/5
错误地如 1,

测试保证 LDA 指令在 0 页
址,

见前测试

见前测试

送入 AC # 和 AC1 相对于 PC 变址
该送入同样的假。

测试变址在 PC 中送入, 折
查地址计数失败

見上面

01573 020402 LDA #1, +2
 01574 101001 MOV #0, #, SKP
 01575 177777 COM #0, #, #, SZR
 01576 100014 HALT
 01577 063077

B67: 020402 LDA #, +2
 101001 MOV #, #, SKP
 177777 123456
 024777 LDA 1, ., -1
 106414 SUB #0, 1, SZR
 063077 HALT

对应于 PC ACC 和 (A) 输入第 =
 各 LDA 指令应在 16 位上加 -1
 折查位过 5 位 (EFA, MBC 等)
 S 中波察发。

01606 020200 LDA #, #, #, #
 01607 030145 LDA #, #, #, #
 01610 025000 LDA 1, #, #, #
 01611 106414 SUB #0, 1, SZR
 01612 063077 HALT

直接送入地址 2 中经过变址
 送入 2 中折查第一套 LDA 指
 令在 EFA 的 S 中电平

01613 020177 LDA #, #, #, #
 01614 034145 LDA #, #, #, #
 01615 025377 LDA 1, #, #, #
 01616 106414 SUB #0, 1, SZR
 01617 063077 HALT

見前面的波成

01620 102300 ADCS #, #, #
 01621 104004 COM #, #, #, #, SZR
 01622 063077 HALT

交错的 ACC

01623 152000 ADC #, #, #, #
 01624 020000 LDA #, #, #, #
 01625 025001 LDA 1, #, #, #
 01626 106414 SUB #0, 1, SZR
 01627 063077 HALT

测试 ACC 变址
 送入地址中
 送入地址中 (-1+1)=0

01630 176000 ADC #, #, #, #
 01631 020000 LDA #, #, #, #
 01632 025401 LDA 1, #, #, #
 01633 106414 SUB #0, 1, SZR
 01634 063077 HALT

見前面的波成

01635 176401 SUB #3, #, #, #, SKP
 01636 104001 JSR. +1
 01637 175004 MOV #3, #, #, #, SZR

JSR 指令应该不执行因为
 (IR # + SKIP) 应该阻止 (T
 MP + JSR) (F+B) 的译码。 25

01640 063077 HALT

01641 020000 LDA #, #, #, #
 01642 101001 MOV #, #, #, #, SKP
 01643 010000 ISZ #
 01644 024000 LDA 1, #, #, #
 01645 106414 SUB #0, 1, SZR
 01646 063077 HALT

15 位指令应该不执行, 折查 I, R 译
 码。

01647 020000 LDA #, #, #, #
 01650 126001 ADC 1, #, #, #, SKP
 01651 044000 STA 1, #, #, #
 01652 024000 LDA 1, #, #, #
 01653 106414 SUB #0, 1, SZR
 01654 063077 HALT

STA 指令应该不执行, 折查 I, R 译
 码。

01655 102001 ADC #, #, #, #, SKP
 01656 060477 BIA #, #, #, #, CPU
 01657 100014 COM # #, #, #, #, SZR
 01660 063077 HALT

DIA 指令不应该执行。

01661 102001 ADC #, #, #, #, SKP
 01662 020000 LDA #, #, #, #
 01663 100014 COM # #, #, #, #, SZR
 01664 063077 HALT

LDA 指令不应该执行。

01665 102000 ADC #, #, #, #
 01666 040000 STA #, #, #, #
 01667 024000 LDA 1, #, #, #
 01670 125015 MOV #1, #, #, #, SNR
 01671 063077 HALT

折查存道指令

01672 022024 LDA #, #, #, #, MZ
 01673 040000 STA #, #, #, #
 01674 024000 LDA 1, #, #, #
 01675 124015 COM #1, #, #, #, SNR
 01676 063077 HALT

STA 失败折查从 ACC3 到
 MVL73 位提上, BIT PICKED
 UP

01677 020125 LDA #, #, #, #, MZ
 01700 040000 STA #, #, #, #
 01701 024000 LDA 1, #, #, #
 01702 124015 COM #1, #, #, #, SNR
 01703 063077 HALT

STA 失败折查从 ACC2 到 MVL2
 位提上, BIT PICKED UP

B82= LDA φ, M5
 STA φ, φ,
 LDA 1, φ,
 COM #1, 1, SNR
 HALT

STA 失败折返从 ACD 到 MUL
 T1 的输入

01704 020126
 01705 040000
 01706 024000
 01707 124015
 01710 063077

STA φ, φ,
 LDA 1, φ,
 SUB #φ, 1, SZR
 HALT

B83= LDA φ, M9
 STA φ, φ,
 LDA 1, φ,
 COM #1, 1, SNR
 HALT

STA 失败折返从 ACD 到
 MULT 的输入,
 在提与 BIT, PICKER UP

01711 020127
 01712 040000
 01713 024000
 01714 124015
 01715 063077

B90= LDA φ, K15673 折返 "STA"
 STA φ, φ,
 LDA 1, φ,
 SUB #φ, 1, SZR
 HALT

B84= LDA φ, K1
 STA φ, φ,
 LDA 1, φ,
 MOV 1, 1, SNR
 HALT

STA 失败折返从 ACD 到
 MULT 的输入
 位落下

01716 020135
 01717 040000
 01720 024000
 01701 125005
 01722 063077

B91= LDA φ, Kφ73567 测试 STA
 STA φ, φ,
 LDA 1, φ,
 SUB #φ, 1, SZR
 HALT

B85= LDA φ, K2
 STA φ, φ,
 LDA 1, φ,
 MOV 1, 1, SNR
 HALT

STA 失败折返从 ACD 到
 MULT 的输入
 位落下

01723 020143
 01724 040000
 01725 024000
 01726 125005
 01727 063077

B92= LDA φ, Kφ44.1 测试 STA 相对于 PC
 MOV φ, φ, SKP
 STA φ, .-1
 LDA 1, .-2
 SUB #φ, 1, SZR
 HALT

B86= LDA φ, K4
 STA φ, φ,
 LDA 1, φ,
 MOV 1, 1, SNR
 HALT

STA 失败折返从 ACD 到
 MULT 的输入
 位落下

01730 020141
 01731 040000
 01732 024000
 01733 125005
 01734 063077

B93= LDA 1, Kφ21042
 MOV φ, φ, SKP
 STA 1, .-1
 LDA 2, .-2
 SUB 2, 1, SZR
 HALT

B87= LDA φ, K1φ
 STA φ, φ,
 LDA 1, φ,
 MOV 1, 1, SNR
 HALT

STA 失败折返从 ACD 到
 MULT 的输入
 位落下

01735 020142
 01736 040000
 01737 024000
 01740 125005
 01741 063077

B94= LDA 3, Kφ421φ4 测试 STA 相对于 PC
 MOV φ, φ, SKP
 STA 2, .-1
 LDA 3, .-2
 SUB #2, 3, SZR
 HALT

B88= LDA φ, K167356 折返 "STA"
 STA φ, φ,
 LDA 1, φ,
 SUB #φ, 1, SZR
 HALT

STA 失败折返从 ACD 到
 MULT 的输入
 位落下

01742 020146
 01743 040000
 01744 024000
 01745 106414
 01746 063077

B94= SUB φ, φ,
 STA φ, φ,
 STA 3, .-2
 LDA 3, .-2
 SUB #2, 3, SZR
 HALT

B89= LDA φ, K156735 折返 "STA"

01747 020147

STA 3, 应该改变 ACD 的内容

02015	103014	ADD #0,0,SKP HALT	02063	126520	SUBZL 1,1, LDA #0,0, SUB #1,1,SKP HALT
02016	063077		02064	020000	
02017	034155	LDA 1,3,K1#421#	02065	106414	
02020	195001	MOV 3,3,SKP #	02066	063077	
02021	006000				
02022	054777	STA 3,-1	02067	102000	C#2=
02023	030776	LDA 2,-1-2	02070	040000	ADC #0,0, STA #0,0, ISZ #0, MOV #0,0, LDA #0,0, MOV #0,0,SKP HALT
02024	156414	SUB #2,3,SKP	02071	010000	
02025	063077	HALT	02072	101010	
02026	020155	LDA #0,K1#421#	02073	020000	
02027	176000	ADC 3,3,	02074	101004	
02030	041401	STA #1,1,3,	02075	063077	
02031	024100	LDA 1,1,0,			
02032	106414	SUB #0,1,SKP	02076	152020	C#3=
02033	063077	HALT	02077	050000	ADC 2,2, STA 2,0, ISZ #0, MOV #0,0, COM #2,2,SKP MOV #0,0,SKP HALT
02034	102401	SUB #0,0,SKP	02101	101010	
02035	177777	-1	02102	150015	
02036	040771	STA #0,-1	02103	101002	
02037	029715	LDA 1,-1-2	02104	063077	
02040	125004	MOV 1,1,SKP			
02041	063077	HALT	02105	020156	C#4=
02042	102301	ADC #0,0,SKP	02106	040000	LDA #0,K1#34 STA #0,0, ISZ #0, MOV #0,0,SKP HALT
02043	000000		02107	010000	
02044	040777	STA #0,-1	02110	101001	
02045	024776	LDA 1,-1-2	02111	063077	
02046	124014	COM #1,1,SKP			
02047	063077	HALT	02112	020156	C#5=
02050	102400	SUB #0,0,	02113	040000	LDA #0,K1#34 STA #0,0, SOBZL 2,2, ISZ -1,2, MOV #0,0,SKP HALT
02051	040000	STA #0,0,	02114	152520	
02052	010000	ISZ 0	02115	011377	
02053	101010	MOV #0,0,	02116	101001	
02054	020000	LDA #0,	02117	063077	
02055	101005	MOV #0,0,SKP			
02056	063077	HALT	02120	020156	C#6=
02057	102400	SUB #0,0,	02121	040000	LDA #0,K1#34 STA #0,0, ADC 0,2,2, ISZ 1,2, MOV #0,0,SKP HALT
02060	040000	STA #0,0,	02122	152040	
02061	010000	ISZ #0	02123	011001	
02062	101010	MOV #0,0,	02124	101001	
			02125	063077	
			02126	020156	C#7=
			02127	040000	LDA #0,K1#34 STA #0,0,

测试 STA 指令

测试用 ACS 变址的 STA 指令

交错 STA 测试全“0”

交错 STA 测试全“1”

在 0 位置中存全“0”在 ISZ 指令以后 C(0) 应等于 (4) 而不是 (1#)

位置 0 置 +0, 然后添加 Acl 为 (1) 并与在添加以后的 Loc 中相比较, 检查 ISZ 指令

位置 0 置 +(-1) 然后添加到 0 检查 ISZ 指令添加的能力。

ISZ 不变该改变 ACS 的或是进位触发的状态

1234 加 1 引起 ISZ 进位, 经 EFA 使 MIBC 正确地移位。

见前面的测试

见前面的测试

见前面的测试

02130 152520 SUBZL 2, 2,
02131 157040 MOV 0, 2, 2,
02132 011371 ISZ 0, 1, 2,
02133 101011 MOV #0, 0, SKP
02134 063077 HALT

02135 102220 C18: ADC 0, 0, 0, 0,
02136 040000 STA 0, 0, 0,
02137 010000 ISZ 0,
02140 101001 MOV 0, 0, 0, SKP
02141 063077 HALT

02142 102000 C19: ADC 0, 0, 0,
02143 040000 STA 0, 0, 0,
02144 010000 ISZ 0
02145 063077 HALT

02146 102001 C10: ADC 0, 0, 0, SKP
02147 000000 0
02150 040777 STA 0, 0, -1
02151 010776 ISZ 0, -2
02152 063077 HALT

02153 102600 C11: SUBER 0, 0,
02154 040000 STA 0, 0, 0,
02155 101001 MOV 0, 0, 0, SKP
02156 022000 LDA 0, 0, 0,

02157 020020 C12: LDA 0, 0, 0, 0,
02160 101001 MOV 0, 0, 0, SKP
02161 036020 LDA 3, 2, 4
02162 024020 LDA 1, 2, 4
02163 106414 SUB 0, 1, 0, SER
02164 063077 HALT

02165 102400 C13: SUB 0, 0, 0,
02166 040000 STA 0, 0, 0,
02167 014000 DSE 0,
02170 101010 MOV #0, 0, 0,
02171 020000 LDA 0, 0, 0,
02172 101005 MOV 0, 0, 0, SNR
02173 063077 HALT

C14: SUB 0, 0, 0,
STA 0, 0, 0,
DSE 0,
MOV #0, 0, 0,
LDA 0, 0, 0,
COM #0, 0, 0, SER
HALT

C15: APC 0, 0, 0, SKP 测试 DSE 从 17777 减至 17776

STA 0, 0, -1
DSE 0, -2
LDA 1, 0, -3
ADC #1, 0, SER
HALT

C16: SUBZL 0, 0, 0, SKP 从 +1 减到 0, 折查在 DSE 时保证
发生溢出

STA 0, 0, -1
DSE 0, -2
HALT

C17: SUB 1, 1, 1, JSR 是否设置 JSR 触发口
JSR +1
MOV 1, 1, 1, SER
HALT

C18: JSR +2 JSR 不能改变 PC, 折查在 JSR 产生 (PC
HALT ENAB) 电平的方式或门 (JMP4 JSR)
SUB 3, 3, 3, JSR 不能改变 AC3 中寄存器 (JSR,
JSR +1 FFA) 电平不设置 PASC 能发电
MOV 3, 3, 3, SNR
HALT

C22: JSR +2 JSR 指令存储的有效地址不是
+1 的指令时折查 JSR, 到
LDA 0, 0, -1 ACB 寄存器 FFA 输入移位
SUB #0, 3, 3, SNR 没有禁止
HALT

C23: JSR +1 寄存器计数的位不应放置。
MOV #3, 3, SER
HALT

在 07777 上加 1 应该产生溢出
100000
ISZ 指令不溢出。

做 -1 加 1 应为 +0 并且溢出标志
试 (TEST SKIP) 能发出溢出标志
端的或 0。

折查相对于 PC 的 "ISZ" 的溢出
的能力

如果 LDA 指令指令起来溢出溢出
口不禁止 DEFFER, 折查在
DEFER 与门 (EFA, IRS, SKIP)
逻辑中, 在此测试中
DEFER 周围

在此测试中, 不应该设置 DEFER
因为 SKIP (溢出) 已被设置, 见前
面的测试。

(用 DSE) 将位置 0 从 +0 减到 -1
减而不发生折查产生溢出
的 (DSE, E, T, SER) 门