APPLICATIONS EXCHANGE

ALTAIR 8800 MEMORY CHECK PROGRAM

It seems that many of the members of S.C.C.S. are owners or soon to be owners of ALTAIR'S (myself included) and are in the first stages of going through the learning curve(s) of this "new" microcomputer technology. Building the hardware is only a small part of this hobby, because once everything is soldered together you're done. Software is where the real challenge and excitement come in. There is no "done" in software where you may sit back and say "I've done all the programs". There is no such point, as software can keep you occupied for years (I hope not on the same program). So to those members still in their learning curve I offer two basic programs for ALTAIR owners.

The first is a memory check program, of which there are several running around. This one is useful when you've just finished that new 4k board you bought from so-n-so and you're wondering if it's OK (memory chips do have a *small* failure rate).

The second is a cassette tape read and write program so you can dump all those programs you were playing with before you put your 8800 to bed and it forgets all it learned. This program was written to operate through the MITS Inc. Audio Cassette Interface, although any I/O parallel to serial modem operating around 2kHz should work. Any medium quality recorder can be used.

The tape I/O port is addressed for control channel - 6 and data channel - 7. Put the bootstrap in manually; it will load the tape input program, which must be the first program recorded. Make sure that the starting address of where the bootstrap is to start loading into memory is four addresses lower than where the body of the program being loaded is to start. This is because the boot is a dump type program and will load into memory the 4 preamble words that the output program places before each program it writes. After the programs are initially entered the output program can be used to write itself onto tape.

SCOTT WILCOX
405-B No. Monterey
Alhambra, CA 91801

DESCRIPTION: This program checks any block of directly addressable memory for read/write data errors. All possible bit patterns are read in and out of each memory location. The program stops (enters a futile loop) and stores the address of the "bad" memory location and the data byte that execution failed on for three situations:

1. a faulty memory location was encountered (data read did not equal data written).
2. a protected block of memory was encountered.
3. the program "stop memory check" address was set higher than the amount of memory actually in the machine.

ADDRESS	MNEMONIC		COMMENT
000	LXI H,L	041	Load stop address of memory check
001		XXX	377 Example: Shows stopping at
002		XXX	017 end of 1st 4K block.
003	LXI D,E	021	Load start address of memory check
004		XXX	060 Start test just above
005		XXX	000 this program.
006	MVI A	076	
007		377	Initial check data
010	MOV A,B	107	Save data for comparison to memory read data
011	STAX D,E	022	
012	STA	062	Store data in case memory fails
013		056	
014		000	
015	LDAX D,E	032	Read data for comparison
016	CMP B	270	compare read/write data
017	JZ	312	Loop if no memory error
020		031	
021		000	

022	XCHG	353	
023	SHLD	042	Store address of bad memory
024		054	
025		000	
026	JMP	303	end of program futile loop
027		023	
030		000	
031	SUI	326	decrement check data
032		001	
033	JNC	322	loop if not final data case (000)
034		010	
035		000	
036	INX D,E	023	increment to next memory address
037	MOV D,A	172	
040	CMP H	274	
041	JNZ	302	loop if not last memory check address
042		006	
043		000	
044	MOV E,A	173	
045	CMP L	275	
046	JNZ	302	loop if not last memory check address
047		006	
050		000	
051	JMP	303	loop to store stop address
052		023	
053		000	
054		XXX	lo address of bad memory
055		XXX	hi address of bad memory
056		XXX	data case that memory failed on

LIMITATIONS: The program requires 46 bytes of known good memory and must not be located in a protected block of memory. Running time is approximately 30 sec. for each 4k of memory being checked

RUNNING THE MEMORY TEST

When the test is running properly most of the address lights on the front of the computer will be lit.You will be able to observe that several of the lights will be brighter than the others and will be working their way across from right to left as the program proceeds through the memory. If you are testing,for example, one 4K board in your system then you would use the start and stop addresses shown in the example. When the program is finished most of the address lights will go out and locations 54 and 55 will hold the address of the bad byte of memory or the address of the next byte above the stopping address.(i.e., the stopping address that you specified in the program)

64K MEMORY CHART

The chart below identifies (in hex and octal) the high order portion of the address for any given 1/4K of memory. Each 1/4K begins with low order address 00_{16} (000_8) and ends with FF_{16} (377_8). Each K is sectioned off for easy identification. Examples:

Section:	Begins with: HX	Begins with: OCT	Ends with: HX	Ends with: OCT
The 8th K	1C00	034 000	1FFF	037 377
The 3rd quarter of all 64K	8000	200 000	BFFF	277 377
The 2nd quarter of the 2nd K	0600	006 000	06FF	006 377
The 38th & 39th K	9400	224 000	9BFF	233 377

HX OCT K	HX OCT K	HX OCT K	HX OCT K	HX OCT K	HX OCT K	HX OCT K	HX OCT K
00 000 1	20 040 1	40 100 1	60 140 1	80 200 1	A0 240 1	C0 300 1	E0 340 1
01 001 2	21 041 2	41 101 2	61 141 2	81 201 2	A1 241 2	C1 301 2	E1 341 2
02 002 3	22 042 3	42 102 3	62 142 3	82 202 3	A2 242 3	C2 302 3	E2 342 3
03 003 1	23 043 9	43 103 17	63 143 25	83 203 33	A3 243 41	C3 303 49	E3 343 57
04 004 1	24 044 1	44 104 1	64 144 1	84 204 1	A4 244 1	C4 304 1	E4 344 1
05 005 2	25 045 2	45 105 2	65 145 2	85 205 2	A5 245 2	C5 305 2	E5 345 2
06 006 3	26 046 3	46 106 3	66 146 3	86 206 3	A6 246 3	C6 306 3	E6 346 3
07 007 2	27 047 10	47 107 18	67 147 26	87 207 34	A7 247 42	C7 307 50	E7 347 58
08 010 1	28 050 1	48 110 1	68 150 1	88 210 1	A8 250 1	C8 310 1	E8 350 1
09 011 2	29 051 2	49 111 2	69 151 2	89 211 2	A9 251 2	C9 311 2	E9 351 2
0A 012 3	2A 052 3	4A 112 3	6A 152 3	8A 212 3	AA 252 3	CA 312 3	EA 352 3
0B 013 3	2B 053 11	4B 113 19	6B 153 27	8B 213 35	AB 253 43	CB 313 51	EB 353 59
0C 014 1	2C 054 1	4C 114 1	6C 154 1	8C 214 1	AC 254 1	CC 314 1	EC 354 1
0D 015 2	2D 055 2	4D 115 2	6D 155 2	8D 215 2	AD 255 2	CD 315 2	ED 355 2
0E 016 3	2E 056 3	4E 116 3	6E 156 3	8E 216 3	AE 256 3	CE 316 3	EE 356 3
0F 017 4	2F 057 12	4F 117 20	6F 157 28	8F 217 36	AF 257 44	CF 317 52	EF 357 60
10 020 1	30 060 1	50 120 1	70 160 1	90 220 1	B0 260 1	D0 320 1	F0 360 1
11 021 2	31 061 2	51 121 2	71 161 2	91 221 2	B1 261 2	D1 321 2	F1 361 2
12 022 3	32 062 3	52 122 3	72 162 3	92 222 3	B2 262 3	D2 322 3	F2 362 3
13 023 5	33 063 13	53 123 21	73 163 29	93 223 37	B3 263 45	D3 323 53	F3 363 61
14 024 1	34 064 1	54 124 1	74 164 1	94 224 1	B4 264 1	D4 324 1	F4 364 1
15 025 2	35 065 2	55 125 2	75 165 2	95 225 2	B5 265 2	D5 325 2	F5 365 2
16 026 3	36 066 3	56 126 3	76 166 3	96 226 3	B6 266 3	D6 326 3	F6 366 3
17 027 6	37 067 14	57 127 22	77 167 30	97 227 38	B7 267 46	D7 327 54	F7 367 62
18 030 1	38 070 1	58 130 1	78 170 1	98 230 1	B8 270 1	D8 330 1	F8 370 1
19 031 2	39 071 2	59 131 2	79 171 2	99 231 2	B9 271 2	D9 331 2	F9 371 2
1A 032 3	3A 072 3	5A 132 3	7A 172 3	9A 232 3	BA 272 3	DA 332 3	FA 372 3
1B 033 7	3B 073 15	5B 133 23	7B 173 31	9B 233 39	BB 273 47	DB 333 55	FB 373 63
1C 034 1	3C 074 1	5C 134 1	7C 174 1	9C 234 1	BC 274 1	DC 334 1	FC 374 1
1D 035 2	3D 075 2	5D 135 2	7D 175 2	9D 235 2	BD 275 2	DD 335 2	FD 375 2
1E 036 3	3E 076 3	5E 136 3	7E 176 3	9E 236 3	BE 276 3	DE 336 3	FE 376 3
1F 037 8	3F 077 16	5F 137 24	7F 177 32	9F 237 40	BF 277 48	DF 337 56	FF 377 64