SWTPC 6800 Short Memory Address Convergence MEMCON-1

This Memory Convergence diagnostic is one designed to check for and locate address convergence problems in the SWTPC 6800 Computer System memory boards, MP-M/MP-MX. The program itself uses 56 words and is meant to be loaded within the 128 word RAM used by the MIKBUG operating system on the MP-A Microprocessor/System board; making the program independent of the MP-M RAM memory. The diagnostic may be loaded from either tape of from the terminal instruction by instruction using MIKBUG starting from address A014$_{16}$ thru A034$_{16}$ and then from address A048$_{16}$ thru MEW The program must be loaded in two parts to avoid interfering with the system's push-down stack. The section of memory to be tested is set by loading the most significant byte of the lower memory address into A002$_{16}$, the least significant byte of the lower memory address into A003$_{16}$, the most significant byte of the upper memory address into A004$_{16}$, and the least significant byte of the per memory address into A005$_{16}$ using MIKBUG just as is done for MIKBUG punch routine. The lower and upper addresses are inclusive and may be any addresses between 0000$_{16}$ and FFFF$_{16}$ with the only requirement that the lower address be less than or equal to the upper address. Since addresses A05F$_{16}$ thru A07F$_{16}$ of the MIKBUG RAM are still available for program use, the diagnostic may run on these locations just to make sure the diagnostic itself is functioning correctly. Since the program counter is set when the program is initially loaded, the routine is initiated after loading according to the "Go To User's Program" section of the Engineering Note 100 in the Operating System section of this notebook. Once initiated, the program can be stopped only by depressing the "RESET" button. The program may then be re-started after setting the program counter to A015$_{16}$ at A048 and A049 as described in the Display Contents of MPU Registers Function" section of the Engineering Note 100.

The test sequence starts by loading a continuous stream of 256 sequential binary numbers from the low memory address to the high memory address, inclusive. It then goes back and sequentially reads the data in each of the locations and compares it to what actually should be there. If it finds any discrepancies within the memory cycle, one X is printed and the cycle is re-started, otherwise a # is printed to indicate successful cycle completion. Since the actual location of any detected errors does not point to the source of the problem, no provision is made for indicating the addresses of detected errors. It must also be noted that the program is not 1001 effective. It would be possible to set bits in multiple locations that coincidentally would have been set anyway. However, each cycle puts different data in each memory location, so the changes of a missed problem are reduced. The program loops forever and may b exited when desired by depressing the "RESET" switch which loads the MIKBUG h control program.

If you wish to eliminate the cyclic printout of the "#" sign you can do so by changing the data in address locations A059, A05A and A05B to NOP instructions (01$_{16}$) using MIKBUG. This way you only get a printout of the error cycles, if any.

SWTPC Short Memory Address Convergence Diagnostic MEMCON-1

```
A002              LOMEM       Starting Address MSB
A003                          Starting Address LSB
A004              HIMEM       Ending Address MSB
A005                          Ending Address LSB

                              Start Loading Program at A014

A014     00       BSTORE
A015     F7       START       STA B BSTORE
A016     A0
A017     14
A018     FE                   LDX LOMEM
A019     A0
A01A     02
A01B     E7       LOOP1       STAB 0,X
A01C     00
A01D     BC                   CPX HIMEM
A01E     A0
A01F     04
A020     27                   BEQ CHECK
A021     04
A022     08                   INX
A023     5C                   INC B
A024     20                   BRA LOOP1
A025     F5
A026     F6       CHECK       LDA B BSTORE
A027     A0
A028     14
A029     FE                   LDX LOMEM
A02A     A0
A02B     02
A02C     E1       LOOP2       CMP B 0,X
A02D     00
A02E     26                   BNE ERROR
A02F     20
A030     BC                   CPX HIMEM
A031     A0
A032     04
A033     20                   BRA JUMP
A034     15

                              Continue Loading Program at A048

A048     A0
A049     15
A04A     27       JUMP        BEQ CYCLE
A04B     0B
A04C     08                   INX
```

```
A04D     5C                      INC B
A04E     20                      BRA LOQP2
A04F     DC
A050     86        ERROR         LDA A #'X
A051     58
A052     BD                      JSR OUTEEE
A053     E1
A054     D1
A055     20                      BRA START
A056     BE
A057     86        CYCLE         LDA A #’#
A058     23
A059     BD                      JSR OUTEEE
A05A     E1
A05B     D1
A05C     5A                      DEC B
A05D     20                      BRA START
A05E     B6
                                 END
```

Memory Address Assignment Table (Hex)

Board #	Memory Quadrant (K of memory)	Starting Addr.	Ending Addr.
	1	0000	03FF
	2	0400	07FF
	3	0800	0BFF
	4	0C00	0FFF
1	1	1000	13FF
	2	1400	17FF
	3	1800	1BFF
	4	1C00	1FFF
2	1	2000	23FF
	2	2400	27FF
	3	2800	2BFF
	4	2C00	2FFF
3	1	3000	33FF
	2	3400	37FF
	3	3800	3BFF
	4	3C00	3FFF
4	1	4000	43FF
	2	4400	47FF
	3	4800	4BFF
	4	4C00	4FFF
5	1	5000	53FF
	2	5400	57FF
	3	5800	5BFF
	4	5C00	5FFF
6	1	6000	63FF
	2	6400	67FF
	3	6800	6BFF
	4	6C00	6FFF
7	1	7000	73FF
	2	7400	77FF
	3	7800	7BFF
	4	7C00	7FFF

MP-M/MP-MX Memory IC Assignment Map

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Quadrant 1 (1K)	IC15	IC13	IC11	IC9	IC7	IC5	IC3	IC1
Quadrant 2 (2K)	IC16	IC14	IC12	IC10	IC8	IC6	IC4	IC2
Quadrant 3 (3K)	IC40	IC38	IC36	IC34	IC32	IC30	IC28	IC26
Quadrant 4 (4K)	IC39	IC37	IC35	IC33	IC31	IC29	IC27	IC25

00 hex = 0000 0000	binary	08 hex = 0000 1000	binary
01 hex = 0000 0001	binary	10 hex = 0001 0000	binary
02 hex = 0000 0010	binary	20 hex = 0010 0000	binary
04 hex = 0000 0100	binary	40 hex = 0100 0000	binary
		80 hex = 1000 0000	binary