SWTPC 6800 Rotating Bit RAM Memory Diagnostic ROBIT-1

This rotating bit memory diagnostic is designed to check for and locate memory retaining problems in the SWTPC 6800 Computer System memory boards, MP-M/MP-MX. The program itself uses 85_{10} 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. This makes the program independent of the MP-M/ MP-MX RAM memory. The diagnostic may be loaded from either tape or from the terminal instruction by instruction using MIKBUG starting from address $A014_{16}$ thru $A07A_{16}$. The program must be loaded in two parts to avoid interfering with the sytstem's push-down stack. The contiguous 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 upper 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 $A07B_{16}$ thru $A07F_{16}$ of the MIKBUG RAM are still available for program use, the diagnostic may be 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 may then be re-started after setting the program counter to $A018_{16}$ at A048 and A049 as described in the "Display Contents of MPU Registers Function" section of the Engineering Note 100.

The test sequence starts from the lower address and loads that address with a binary 0000 0001 or 01_{16}. The data in this location is then read and verified. If accurate the "one" bit is shifted left to form a binary 1000 0010 or 02_{16} and is then again tested. This shift left sequence continues until a binary 1000 0000 or 80_{16} has been loaded and verified, at which time the entire sequence is repeated at the next sequential memory address. This sequence continues until the selected upper memory address is reached. The program then prints a "+" on the control terminal to indicate cycle completion and proceeds to repeat itself. The program loops forever and may be exited when desired by depressing the "RESET" switch which loads the MIKBUG control program. When an error is detected, the memory address followed by what data should have been followed by what the memory data was, are printed out on the control terminal in hexadecimal (base 16) form. Example:

```
*0110 02 00
```

When converted to binary this means that when address 0110, which is located in the first 1,024 words of RAM memory, was loaded with a binary 0000 0010 it was read back as containing a binary 0000 0000 which indicates a possible problem in the 2^1 bit memory chip in the lower 1,024 words of memory or a possible problem in the 2^1 bit of the memory board data transceiver or a variety of other possibilities. The best way to tell for sure is to look for a pattern in the indicated errors. Take note that once one bit error has been located at a specific memory address, the one error is printed in the form shown above and the program increments to the next address without searching for more errors in the already defective address.

If you wish to eliminate the cyclic printout of the "+" sign you can do so by changing the data in address locations A076, A077 and A078 to NOP instructions (01_{16}) using MIKBUG). This way you only get a printout of the error locations; that is if there are any. The running time of this program is very fast. It will cycle thru 2,048 words of memory in less than one second.

SWTPC Rotating Bit Memory Diagnostic ROBIT-1

```
A002              LOTEMP   Starting Address MSB
A003                       Starting Address LSB
A004              HITEMP   Ending Address MSB
A005                       Ending Address LSB

                  Start Loading Program at A014

A014     00       INXMSB
A015     00       INXLSB
A016     00       ACCA
A017     2B       FLAG
A018     FE       START    LDX LOTEMP
A019     A0
A01A     02
A01B     86       LODREG   LDA A #1
A01C     01
A01D     A7                STA A 0,X
A01E     00
A01F     A1                CMP A 0,X
A020     00
A021     26                BNE ERRPNT
A022     0D
A023     48       LOOP1    ASL A
A024     68                ASL 0,X
A025     00
A026     A1                CMP A 0,X
A027     00
A028     26                BNE ERRPNT
A029     06
A02A     81                CMP A #$80
A02B     80
A02C     26                BNE LOOP1
A02D     F5
A02E     20                BRA INCR1
A02F     3B
A030     FF       ERRPNT   STX INXMSB
A031     A0
A032     14
A033     CE                LDX #MCL
A034     E1
A035     9D
A036     20                BRA SKIP1
A037     12

                  Continue Loading Program at A048

A048     A0                Program Counter MSB
A049     18                Program Counter LSB
A04A     B7       SKIP1    STA A ACCA
```

```
A04B     A0
A04C     16
A04D     BD                JSR PDATAl
A04E     EO
A04F     7E
AD50     CE                LDX #INXMSB
A051     AO
A052     14
A053     BD                JSR OUT4HS
A054     EO
A055     C8
A056     CE                LDX #ACCA
A057     AO
A058     16
A059     BD                JSR OUT2HS
A05A     EO
A05B     CA
A05C     FE                LDX INXMSB
A05D     AO
A05E     14
A05F     BD                JSR  OUT2HS
A060     EO
A061     CA
A062     CE                LDX #MCL
A063     E1
A064     9D
A065     BD                JSR PDATA1
A066     EO
A067     7E
A068     FE                LDX INXMSB
A069     AO
A06A     14
A06B     BC       INCR1    CPX HITEMP
A06C     AO
A06D     04
A06E     27                BEQ FINISH
A06F     03
A070     08                INX
A071     20                BRA LODREG
A072     A8
A073     B6       FINISH   LDA A FLAG
A074     AO
A075     17
A076     BD                JSR OUTEEE
A077     E1
A078     D1
A079     20                BRA START
A07A     9D
                           END
```