65K RAM Memory Modification For The Sol-20

by Jim Spann

Don't give your Sol to the trash man; a savior is here! This simple modification gives the Sol Terminal Computer* a new lease on life by moving the VDM* and Solos* memory to the Sol's internal data bus (where it belonged anyway), so it no longer interferes with the S-100 memory address space. Now you can run all those big memory programs without having to buy a new computer, and still have access to all the Solos utility routines. And all old programs will execute properly.

A minor wiring change and the addition of two integrated circuits is required to extend the usability of the Sol Terminal Computer. This feat is accomplished by the use of a memory management flip-flop, controlled by a unused output port in the Sol. The parts required are a SN74LS74 (flip-flop) and SN7406 (open collector inverter).

The required circuit changes can be made without cutting any printed circuit board traces. The technique of hanging the IC pin to be changed outside of the socket and soldering a wire to it can save much wear and tear on the circuit board (Figure 1). The two chip memory management control circuit can be assembled on a small vector board and mounted inside the Sol under the keyboard as shown in Figure 2.

The following steps refer to the Sol schematics and drawings in the Sol manual. *Be sure to unplug and remove any S-100 boards during these steps.*

•Step 1. Build the custom memory management control circuit as shown in figure 3 on a small vector board. Set this board to one side. It will be used in a later step.

•Step 2. (This step moves the display memory data output signals from the S-100 bus to the Sol's internal bus.) Lift all the output pins of the tri-state I.C.s (see drawing 4) U29 and U89 and tie to the internal bus signal INT 0—INT 7, (see drawing 1). The internal bus runs all over the Sol mother board; use any handy INT 0—INT 7 signals to connect to. Be sure to mark-up changes and keep a accurate set of prints of your computing system.

PIN 13 of U89 (74LS367) to "INT 0" PIN 10 of U79 (74LS253)
PIN 11 of U89 (74LS367) to "INT 1" PIN 6 of U79 (74LS253)
PIN 13 of U29 (74LS367) to "INT 2" PIN 10 of U65 (74LS253)
PIN 9 of U29 (74LS367) to "INT 3" PIN 6 of U65 (74LS253)
PIN 7 of U29 (74LS367) to "INT 4" PIN 10 of U78 (74LS253)
PIN 11 of U29 (74LS367) to "INT 5" PIN 6 of U78 (74LS253)
PIN 3 of U29 (74LS367) to "INT 6" PIN 10 of U66 (74LS253)
PIN 5 of U29 (74LS367) to "INT 7" PIN 6 of U66 (74LS253)

•Step 3. (This step modifies the control of the internal/external multiplexer (U66, U78, U65, U79—see drawing 1) to allow the data from the display to get to the processor.)
Lift PIN 2 of U44 (74LS00) and tie it to PIN 1 of U44.

•Step 4. (This step moves the MWRITE signal of the internal RAM, so that it may be controlled by the memory management circuit.)
Lift PIN 9 of U44 (74LS00) and tie to PIN 14 of U46 (8T30). See drawing 4.
Lift PIN 13 of U24 (74LS04) and tie to PIN 14 of U46 (8T30). See drawing 2.

•Step 5. At this point the Sol computer should operate normally. Plug it in and try some programs that use the display; TARGET is a good test program. If the system does not work there is a wiring error, so double check everything and try again.

•Step 6. In step 6 the connection of the memory management circuit board is installed. Mount the memory management board and connect the circuit to VCC (+5 Vdc) and ground. This power comes from the Sol mother board. Connect to the following signals to the Sol. See figure 2 and drawing 2.

Jim Spann, Box 2061, Livermore, CA 94550.
* trademark names of Processor Technology Inc.

74LS74 PIN 1 (reset) to (S-100 signal POC) PIN 12 of U77.
74LS74 PIN 3 (clock) to (OUTPUT FCH) PIN 11 of U35.
74LS74 PIN 2 (data) to (S-100 signal D0) PIN 2 of U80.
7406 PIN 8 to PIN 3 of U34.
7406 PIN 1 and PIN 13 to PIN 6 of U23.
7406 PIN 2 to (S-100 signal MWRITE) PIN 11 of U50.
7406 PIN 12 to (S-100 signal FRDY) PIN 1 of U49.

This completes the modification of the Sol.

*Step 7. The system should operate normally; retest as in step 5. If there are any problems check the memory management flip-flop to make sure the Power On Clear (POC) resets it to a low level at PIN 5.

Theory of Operation

When the computer is first turned on the memory control flip-flop is cleared via the Power-On-Clear signal (POC). This signal is also generated when a restart is performed (holding both the upper case and repeat keys down). The Sol will operate normally with the Solos/display RAM/ROM memory block enabled.

The memory control flip-flop controls accesses (reads/writes) to the C000—CFFF hex memory block. This block 4K Solos/display RAM/ROM or a 4K RAM (can be part of a larger memory plane) memory on the S-100 bus. In other words the memory control flip-flop switches in the internal Solos memory or the external S-100 memory.

Operation

Software control of the memory management flip-flop is accomplished via the output instruction OUT FC and bit 0. If bit 0 is set to a zero (0) then this is normal Sol operation. If bit 1 is set to a one this enables the memory on the S-100 bus.

The programming example illustrates how to have a full 65K RAM system and use the Solos utilities with CP/M . The cold boot switches off the internal memory and turns on all RAM external memory.

Software Rules

These rules should be kept in mind when using this system.

1) Do not switch to the internal memory (Solos) if the STACK is in the C000—CFFF address area. Save the stack first, or the program will not be able to find its way back.

2) Do not switch to the internal memory from insided the C000—CFFF address area. ■

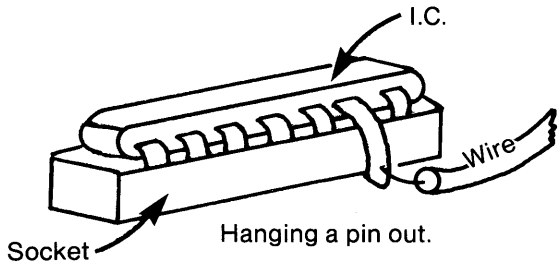

Figure 1.

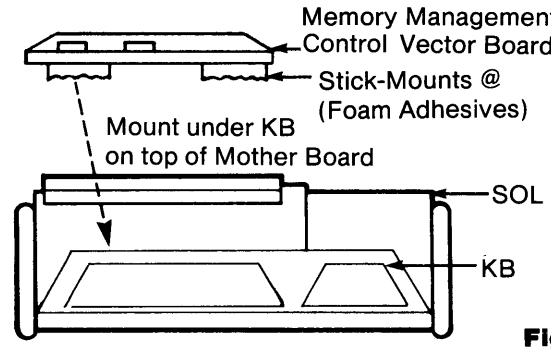

Figure 2.

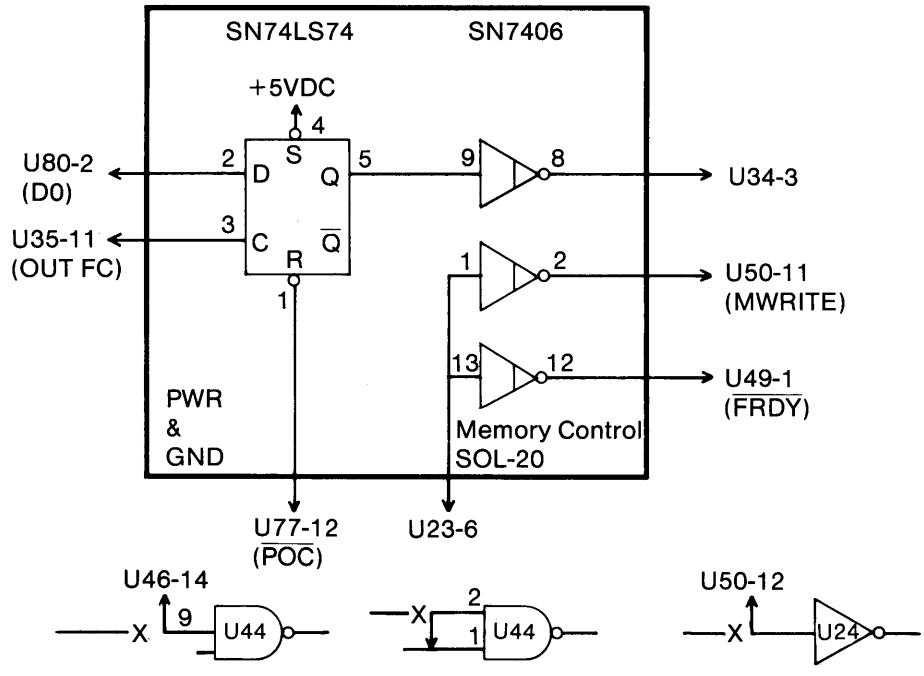

Figure 3.

The following is an example of a CP/M interface using the memory management modification.

```
;
memsw   equ     0FCH            ;output that control the memory
solon   equ     0               ;solos on, normal operation
solof   equ     not solon and 0FFH      ;enables all ram
;
sinp    equ     0C01FH          ;SOLOS entry points
sout    equ     0C019H
aout    equ     0C01CH
ainp    equ     0C022H
stat    equ     0FAH            ;Keyboard status
char    equ     0FCH            ;Keyboard data in
;
boot:   lxi     sp,stack

        mvi     a,solof         ;turns off solos rom/ram area
        out     memsw
          .
          .
          .

        jmp     cpm;
;
wboot:  lxi     sp,stack

        mvi     a,solof         ;turns off solos
        out     memsw
          .
          .
          .
        jmp     gocpm
;
;    I/O ROUTINES
;*****************************************************
;
const:  in      stat            ;Sol KB
        cma
        ani     1
        rz
rtn:    mvi     a,0ffh
        ret
;
conin:  in      stat            ;Sol KB
        ani     1
        jnz     conin
        in      char
        ora     a
        jz      conin
        ret
;
reader: mov     a,1             ;serial port
        lxi     h,ainp
        jmp     memctrl         ;send it to solos
;
punch:  mov     b,c             ;for adout
        mvi     a,1             ;serial port
        lxi     h,aout
        jmp     memctrl ;send it to solos

;       MEMCTRL
;       memory control
;       26 nov  79
; This program maps the solos area (C000 - CFFF) on & off.
; To allow for 64k byte operation of the Sol and still have
; access to the Solos software contain in the C000-CFFF area
; which includes the VDM, the following procedure is required.
;
; Power On Clear enables the Solos area so the first thing
; the boot program should do is turn off the Solos area.
;
; NOTE: THIS PROCEDURE MUST EXECUTE OUTSIDE OF THE C000-CFFF
;       ADDRESS SPACE.
;
; program calling sequence example:
;*****************************************************
conout  push    h               ;save h we need it
        lxi     h,sout          ;vector to vdm in solos
        call    memctrl
        pop     h               ;restore h
        ret
;*****************************************************
;
memsw   equ     0FCH            ;output that control the memory
solon   equ     0               ;solos on, normal operation
solof   equ     not solon and 0FFH      ;enables all ram
;
memctrl shld    vector+1        ;store vector
        lxi     h,0
        dad     sp              ;get stack
        shld    stkreg          ;save it
        lxi     sp,stack        ;get a local stack outside
                                ;C000 - CFFF range
        mvi     a,solon         ;turn on solos so we have access
        out     memsw
vector  call    0               ;get set on entry
        mvi     a,solof         ;turn off solos area
        out     memsw
        lhld    stkreg          ;recover org stack
        sphl
        ret
stkreg  dw      0
        ds      20
stack   dw      0

        end
```