

THE SOL WITH THREE BRAINS.

(H. Holden, April 2020).

Introduction:

First there was the movie: “The Thing with Two Heads” from 1972. Then the movie: “The Man with Two Brains” from 1983 with Steve Martin. Now we have a SOL-20 computer with “Three Brains”. This project shows how two of PT’s ROM modules can be added to the computer.

The “brain” in Processor Technology’s (PT) SOL-20 computer is the “ROM module” which contains 2048 bytes of firmware. This program handles I/O routines for “Pseudo-ports” including Tape I/O functions. It was fairly aptly named because it does give the SOL-20 computer a personality of sorts. With this firmware the user can interface to the outside world via the serial, parallel or tape ports and also interface with the keyboard and video monitor. Also dump memory contents at will. It is a simple and elegant mini-operating platform.

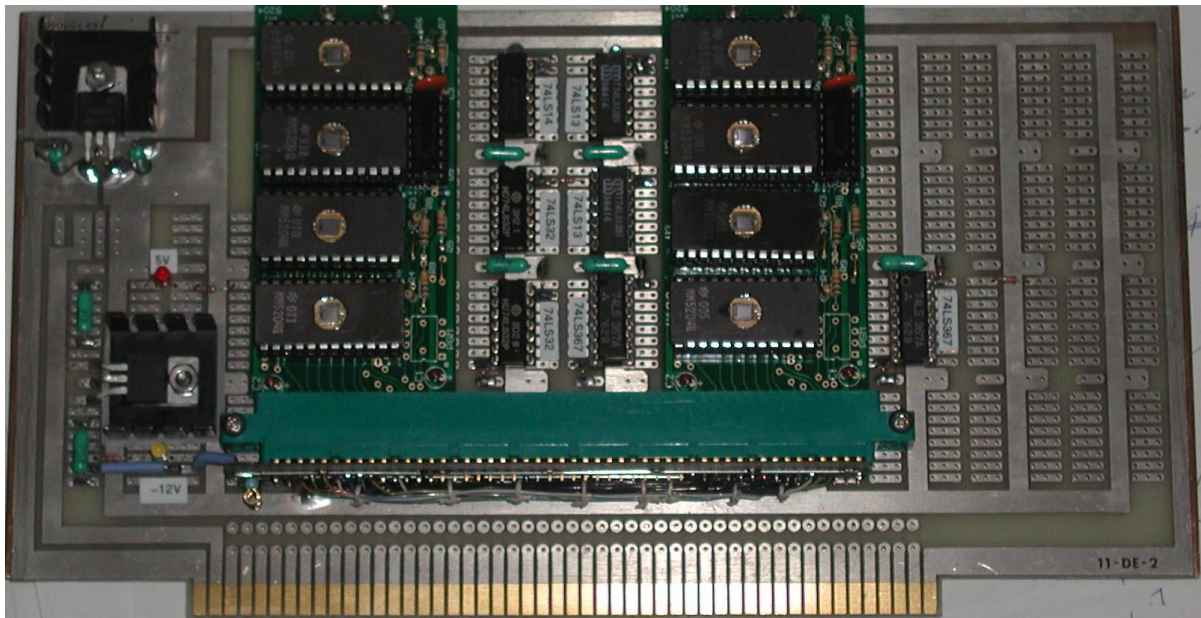
In a recent project I built some replicas of PT’s module using the four MM5204 UVeproms which were programmed with the aid of the Eberhard programmer:



Two replica modules were manufactured and added to an S-100 prototype pcb. My Sol has 48k of general RAM and runs 48k CP/M. I did not want to use up any of the existing RAM space which also includes the 1k of video RAM and the 1k of general RAM on the main board for small programs, so I considered some extra ROM space would be nice.

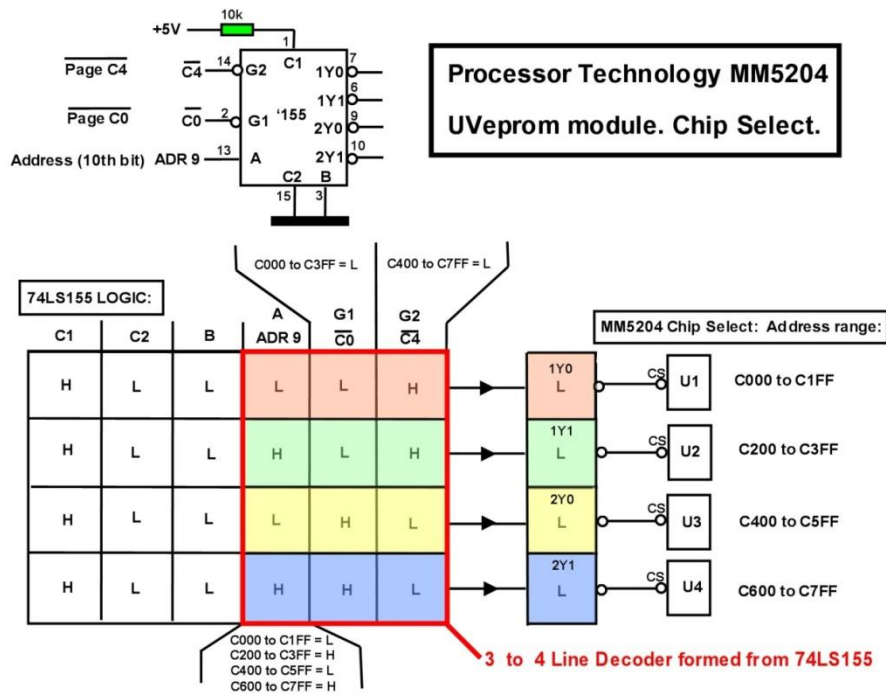
The plan was to use the address area from D000h to DFFFh or 4096 bytes of memory space currently not used in my Sol. This also avoids an area at E800h used by my North Star disk controller card.

The photo below shows the hand crafted prototype card. It receives two of PT's ROM modules:



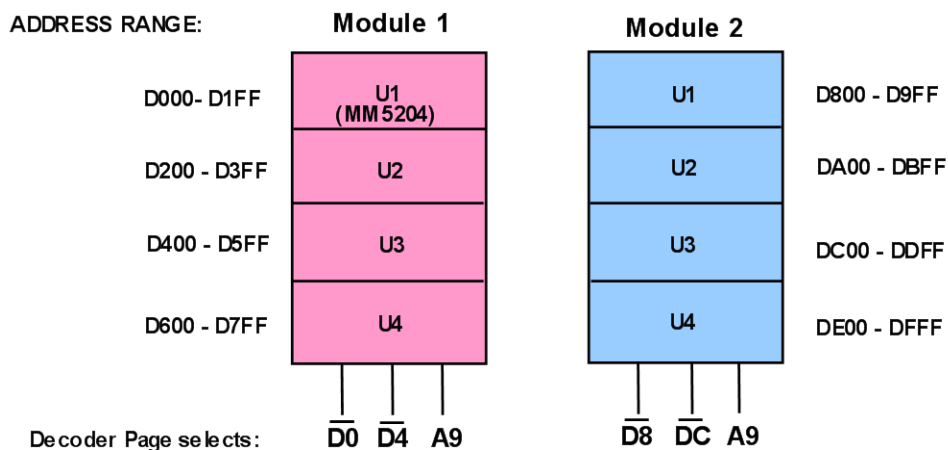
The main trick with this project was interfacing the PT modules to the S-100 bus. This is so they respond over the correct address range and can interface with the S-100 data bus. It turned out in the Sol that the original module does not interface directly with the S-100 data bus. It has its own interface built onto the Sol motherboard. Therefore the interface for the two additional modules needed to be custom designed, so that the modules themselves could remain original according to PT's design for them.

The original ROM chip select circuitry on the module (which was not altered) is shown below. It makes use of a 74LS155 IC wired as a three line to a four line encoder:



As can be seen it works on an address “page select” system and address A9 to select the ROMs over the range C000h to C7FFh. Each MM5204 ROM, U1 to U4 holds 512 Bytes each.

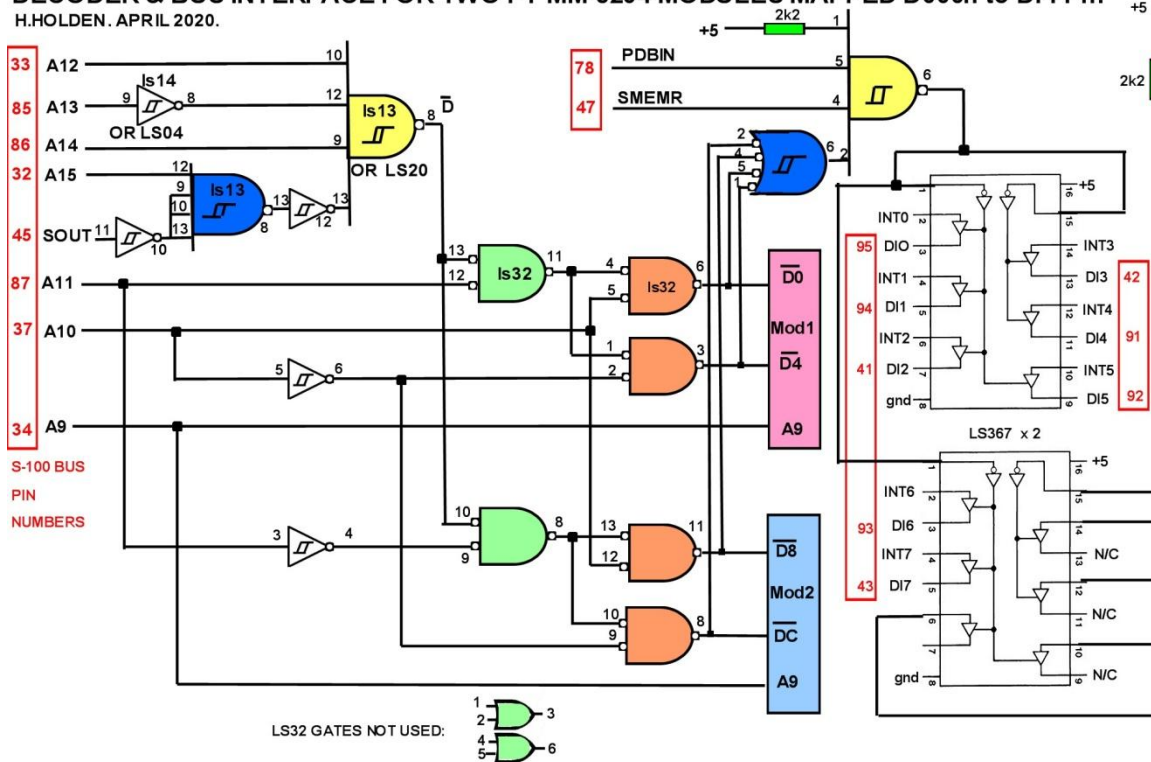
Making this operate on other pages requires the arrangement shown below, to select two modules in sequence starting at page D0 instead:



This is achieved with the circuit below. The connections INT0 to INT7 are connections on the module’s edge connector.

DECODER & BUS INTERFACE FOR TWO PT MM-5204 MODULES MAPPED D000h to DFFFh.

H.HOLDEN, APRIL 2020.



I found it was not necessary to use any wait states to reliably transfer the data in the ROM's to the data bus. (The Schmitt trigger versions of the LS IC's are not required, I used them because I was running low on stocks of the others).

Now, what to load into the ROMs ? Firstly the most frequently used program is the MEMTEST program. It always pays to check the memory after start up, so I placed this in U1 of module 1. It is now invoked by typing EX-D000. To get this to work required re-assembling the program in the Sol to org at 0D000h. Then transferring the resultant .HEX file (using PCPUT) to my modern computer running Tera Term and communicating with the Eberhard programmer to program the MM5204.

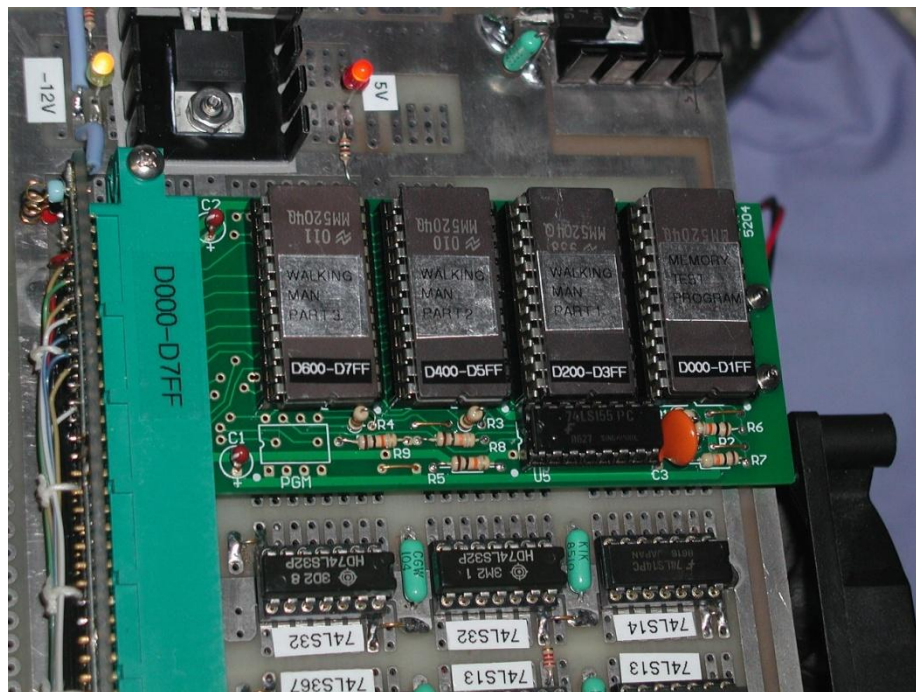
For the 3 remaining MM5204 ROMS on module 1, I loaded my "Walking Man" program, which produces a walking man on the SOL's Text screen using the "\$" symbol. (His name is Steve) a You-tube video of him can be seen here:

<https://www.youtube.com/watch?v=D8PEJsoEWUc>

The file for the Walking Man is quite large for what seems like a simple "graphic" on a text screen, it uses up about 1270 Bytes, I'm not an experienced programmer, so it is probably inefficient. The program is now invoked with EX D200.

The four ROM's on module 2 remain to be programmed.

Once I have programmed a ROM, I occlude the window with sticky aluminium foil to totally occlude light. Then a clear sticker is added over that with black writing (from a Brother Label Machine) to say what the program is. Also, the address range is on a sticker on the ROM. That way it will never be forgotten what the ROM is supposed to be doing and the address range over which the program running when it is operating.



The photo below shows a side view of how the modules are mounted on the original holes. Also, at the connector end there is a blank pcb inserted into the connector between the modules, so they cannot move out of alignment with the socket pins.



Due to this "S-100" card being taller than usual, I run it in the top slot of the SOL-20 card stack:



The photo of the hand wired board from below:

