The SSM VB3 Video Board

by Jon Bondy

The VB3 video board is a 'third-generation' memory-mapped character-oriented video board made by SSM Microcomputer Products. Unlike the first generation (16 by 64) and second generation (24 by 80) display boards, the VB3 offers a dense display (up to 51 lines by 80 characters) with a variety of character 'attributes' such as low intensity, reverse video, underscore, blinking, user-defined characters and graphics characters. In addition, unlike most of the previous generation boards, the VB3 provides a blank select option whereby the refresh memory can be removed from the address space of the computer when not in use (even in-between writing each character to the screen), allowing almost all of the computer's 64K byte address space to be used without interference.

The VB3 comes with a thick ($1/4$ inch) manual which covers a lot of material, including unpacking/assembly instructions, board setup, theory of operation, schematics and board layout, a short section on troubleshooting, and voluminous listings of I/O drivers. Although I did not build the kit, I did look over the assembly instructions and they appear to take the builder through the process step-by-step, with more warnings and comments than are usually found in such manuals. Board checkout is a staged process, with each section of the board being tested before the IC's for the next stage are inserted. The VB3 uses the SMC CRT5037 and CRT8002 ICs which are programmable video display chips, and the SSM manual discusses how to use their flexibility for a variety of purposes. One of the chief features of the VB3 is the access which the user gets to these programmable IC's.

The VB3 board contains sockets for 8K bytes of memory, of which only 4K bytes were installed on my board. This allowed for storage of 2048 characters (enough for an 80 by 24 character display) and 2048 attribute bytes (to determine if each character is to be displayed normally or with reverse video or with underlining or the like). If one were interested in using the VB3 to display 51 lines by 80 columns, one could install the additional 4K bytes of display memory.

Jon Bondy, Box 148, Ardmore, PA 19003.

In order to use the board properly, one first must initialize the attribute memory for normal display and then clear the character memory to all blanks. One might think that the default value of the attribute memory for 'normal' display would be zero, but SSM implemented the hardware so that the value '3' must be used instead. There are two ways to set the board up, either as a normal memory board (so that it is always in the address space of the computer) or with the bank select option enabled so that the memory on the board can be removed from or brought back to the address space of the computer on command. I started running the board in the former mode for simplicity, and then tried to use the more complex memory management feature later on.

The VB3 video board is a 'third-generation' memory-mapped character-oriented video board.

The VB3 allows the user to configure it in a variety of ways. The user can determine whether the board uses horizontal and vertical sync signals generated on the board itself, or whether it uses externally supplied signals. One can change the character width in a range from 6 to 16 dots, and one can address the VB3's on-board keyboard ports to any port address pairs in the 256 port range. One can instruct the VB3 to latch the S-100 bus address lines during each bus cycle, or to accept the bus data directly (through receiver buffers, of course) without latching, by setting some jumpers. One can set the display memory address to start at any multiple of 8K bytes, and one can set the CRT5037 control register port addresses to start at any multiple of 16. Finally, the bank select option can be enabled or disabled.

The VB3 which I reviewed was sent to MICROSYSTEMS by SSM already assembled, so I did not get a chance to inspect it as closely as I might have had I performed the assembly myself. Upon inspecting the board, however, I noticed a few jumper wires and cut wire runs on the back side, indicating that some sort of bugs in the board layout had been fixed at the factory; I found no documentation on the changes in the manual.

The first thing I tried to do with it was to plug it in and use it simply as a memory board. The board is densely packed, and there is a voltage regulator heat sink at each edge which got in the way of my card guides, making it difficult to insert into my computer; for testing purposes, I inserted the VB3 into an extender board. When I powered up my computer to test the VB3 as a RAM board, my ROM monitor would not work; when the VB3 was removed, and the monitor began to work again. The VB3 was interfering with the rest of my system in some way.

It turned out that the factory 'assembled and tested' unit had a jumper wire missing; it was clearly described in the manual, but I had assumed that the unit would work as delivered, without changing jumpers. With the jumper applied and the various switches correctly set up, I tried again. My ROM monitor worked and so did the VB3, at least as a memory board. It passed my memory tests perfectly, but nothing appeared on the screen.

After some thought I realized that the flexibility which the programmable CRT5037 IC offered was also going to be an inconvenience, since the board was not going to do a thing until I 'programmed' it; and I would have to program it each time I reset my computer. I had encountered a similar situation when I purchased and modified the S.D. Sales VB-8024 (see MICROSYSTEMS, Vol 1 No 1, p. 11), so once I realized that the VB3 was not a simple plug in board like its first- and second-generation predecessors, I began to read the I/O driver listings to see what I had to do.

I keyed in the initialization routine which I found in the manual (about 40 bytes) and ran it, and the screen filled with the garbage which was resident in the memory after it powered up. I filled the attribute memory with '3's, the value which causes the board to display normal characters, and then ran a memory test on the character storage portion of the board. Patterns began to race across the screen as the memory test progressed. Unfortunately, although the patterns were correct, the characters themselves seemed to be shimmering, as if black 'snow' were all over the picture. Deciding that the shimmering was a minor problem, I proceeded to try to use the board with its bank select feature. I never managed to solve the shimmering problem, though, and the characters remained only marginally legible.

The tests which I had run up until now were simple ones run from my ROM monitor, but they were laborious, since I had to key them in each time I powered the computer down. I wanted to continue using the VB3 with my UCSD Pascal system, but to do that I needed all 64KB of my computer as RAM (not as a video board); I needed the bank select feature to work. Unfortunately, for quite a while I could not get that feature to work at all. The VB3 moves the on-board memory into the computer's address space when the keyboard data port is written to (since under normal circumstances this would never happen), and removes it from the address space when the keyboard status port is written to. After much muttering and peering about with my scope, it turned out that the VB3 would not accept just any old port number for the keyboard port value, but if I used a port address of 0FEH, it would work and the on-board memory could be moved into and out of my computer's memory at will. To test it, I had written a short Pascal program, which follows this article, and it finally began to work. In order to get it to do so, however, I had to play with the location where I placed the display memory, since it could neither lie on top of the Pascal program nor the p-machine interpreter. This took some finagling. Unfortunately, it did not work consistently, and it is not clear whether the problem is in the VB3 or in my computer system.

```
program ssmvb3test;
  const
    vtac = 208; { video chip control registers -- 0D0H }
    Kbddata = 254; { 0FEH -- write to remove VB3 from address space }
    Kbdstat = 255; { 0FFH -- write to put VB3 in address space }
    video = 16384; { 04000H -- address of video ram }
  var
    i : integer; ch : char;

  procedure portwrite( addr : integer; data : integer ); external;
  procedure memwrite( addr : integer; data : integer ); external;

  begin
  portwrite( vtac+14,0 ); { reset video chip }
  portwrite( vtac+10,0 );
  portwrite( vtac,103 ); { character times per line }
  portwrite( vtac+1,188 ); { interlace and synch pulse time information }
  portwrite( vtac+2,109 ); { scans per row and characters per scan }
  portwrite( vtac+3,23 ); { skew bits and data rows per frame }
  portwrite( vtac+4,6 ); { scans per frame }
  portwrite( vtac+5,49 ); { vertical margin }
  portwrite( vtac+6,23 ); { row address of last row on screen (scrolling) }
  portwrite( vtac+14,23 ); { restart timing chain }

  portwrite( Kbdstat,0 ); { place VB3 in address space }
  { fill attribute memory with '3' }
  for i := 1 to 2048 do
    memwrite( video + 4096 + i,3 );
  { fill video memory with stuff }
  for i := 1 to 2048 do
    memwrite( video + i,i );
  portwrite( Kbddata,0 ); { remove VB3 from address space }
  end.
```

SSM VB3 cont'd. . .

Despite the fact that I have not yet gotten the VB3 to work satisfactorily, I think that I can make some comments about it. My initial reaction to the board (even before I received it) was that it was unwise to put the character memory in the computer's address space, since at this point in the development of S-100 bus computers the software requires more memory resources than it did only a few years ago. SSM solved that problem by allowing the user to remove the VB3 from the address space of the computer except for the times when things are actually being moved into that memory. This allows boards like the VB3 to be used with software systems like UCSD Pascal, as long as the feature works reliably. I have every confidence that my problems with this feature were due to my computer system rather than to a problem with the SSM design.

When I agreed to review the VB3, I had figured that it was just another memory-mapped video board, and that with an afternoon of playing with it I could make some useful observations about it. Unfortunately, it's too smart. I couldn't just plug it in and test it; rather, I had to program it and coddle it in order even to see a few characters on the screen. The complexity of the board is such that one can't just use it for video output. You must initialize it carefully, and then provide a non-trivial driver program to use it.

If you have a requirement for more than 24 lines of text or for a 'terminal' with complex features which you cannot find elsewhere, you should consider the VB3.

If you compare the VB3 with the S.D. Sales VB-8024, you discover that the two boards have similar capabilities (although the VB3 has more memory, more attributes, and more flexibility). The chief differences lie in the fact that the VB-8024 has its own on-board processor and driver ROM; you turn it on and use it, simply and with no tricks or problems. The on-board processor on the VB-8024 also makes complex manipulations of screen data less of a processing burden for the main processor. The VB3 offers more flexibility and capability while at the same time requiring more care on the part of the user. The VB-8024 is a 'black box' which the user plugs in and uses, while the VB3 is a complex but capable peripheral which the user must customize carefully.

In summary, if you have a requirement for more than 24 lines of text or for a "terminal" with complex features which you cannot find elsewhere, you should consider purchasing the VB3. I think that you will find the VB3 to be more complex and troublesome than it is worth if your application is a simple 80 by 24 terminal; in this case, the VB-8024 will give you the same functions for the same price and with a lot less software headaches.

The SSM VB3 is $375 (kit) and $440 (assembled) and is manufactured by SSM Microcomputer Products, 2190 Paragon Drive, San Jose, CA 95131. □