Seattle Computer Products' 8086 System

by Bill Machrone

Considerable attention has been generated by some of the recent entries into the 16-bit arena for the S-100 bus. Some manufacturers are still talking about it, others are doing something about it and a few are already old hands at it. Seattle Computer Products (SCP) has been manufacturing IEEE-696 compatible 8086 processors and 16-bit wide memories for more than two years. Additionally, they offer a system support board and a serial I/O board, all compatible with the 8086 or any other processor that follows the IEEE-696 Standard.

All the well-intentioned hardware in the world is worthless without software to make it go, and Seattle has pioneered here as well. Long before CP/M-86 was released, Seattle's 86-DOS was a reality. Below, we'll take a look at the available products and give an evaluation of just how fast it is and how useful it could be in your system.

Hardware

The processor board itself contains an 8 MHz 8086 which can be switch-selected to run at 4 MHz. The board produces or responds to all the standard S-100 signals, including SXTRQ* and SIXTN*. This means that the board can address memories that are either eight or sixteen bits wide and, in accordance with the IEEE-696 Standard, permits intermixing them in the same system. The memory cards must support 24-bit extended addressing. The processor handles memory and I/O references as either eight bit transfers, sixteen bit transfers or "double eight bit transfers" where memory is incapable of a sixteen bit transfer. There is a provision for an Imsai-style front panel, but a small modification is necessary to make it work. Examine and deposit functions are inoperative with the 8086.

The CPU Support Board has all the goodies necessary to make the system functional, including a monitor/bootstrap EPROM, two 8259A interrupt controllers, two 16-bit counter/timers, a 24 hour clock (more timers, actually) with provision for battery backup, a serial port, a parallel port and a sense switch input port. Strangely enough, the parallel port is configured as a separate parallel input port and a parallel output port, each with its own cable header on top of the board. This may be advantegeous for some applications, but doesn't permit a full handshaking bidirectional configuration. The bootstrap EPROM has a full 8086 monitor program which allows memory inspection, tracing, debugging and booting the disk controller. At this writing, Seattle does not manufacture a floppy disk controller, so you can request an EPROM which boots one of several popular disk controllers, such as the Tarbell double density controller or the Cromemco 4FDC.

The timers are implemented with the versatile AMD 9513, which provides five timers—one intended as a baud rate generator, two general purpose and two which can be configured as a time of day clock with 0.01 second resolution, or which can also be used as general purpose timers. It has settable alarm registers, which can generate interrupts. Much has already been written about the 8259A interrupt controllers, and their power and versatility is well known. They are configured in a master/slave relationship on the CPU Support Board. Further slave controllers or interrupt sources can be added via the S-100 vectored interrupt lines. Most of the board's options can be selected by dipswitch, and there are several pin jumpers for other options.

The boards were subjected to all the normal abuses, such as fast clock rates and high ambient temperatures, and performed flawlessly.

The decision to spread the CPU and system management functions over two boards is a sound one. Both are uncluttered, easy to configure and run cool. The two-board approach also gives the user some flexibility in upgrading existing systems. The CPU Support Board could be used by any processor, although it might duplicate one or more of the functions found on the popular 8-bit CPU cards. It's also possible to use someone else's support card with the 8086 card, such as Godbout's new System Support 1.

Bill Machrone, 595 West Hartsdale Ave., White Plains, NY 10607.

Seattle has been producing rock-solid memory boards for as long as they have been in business. The 8/16 RAM follows in that tradition, providing a sixteen bit data path for fastest performance in an 8086 environment. As the name implies, it can be used as an 8-bit memory as well. It appears as either 16K of 8-bit memory or 8K of 16-bit memory. Each card can be addressed anywhere in the 16 Megabyte S-100 address space and can be set to respond to PHANTOM*. They are fully static and use the standard 4044 memory chip.

As with the boards mentioned previously, these boards are models of spacious layout and clean design. The boards provided for the review were subjected to all the normal abuses, such as fast clock rates and high ambient temperatures, and performed flawlessly. The 6 MHz Z-80B actually places more demands on them during instruction fetch cycles than the 8086 does at 8 MHz in any operation mode. They proved to be a match for the worst conditions I could provide in several system environments.

Software

Over the months that I've had the Seattle system for review, the software has been a living, growing thing. I received an early copy of 86-DOS and have received several updates. Then there was a long delay while we waited for Microsoft to modify stand-alone Basic-86 to run under 86-DOS. The conversion was finally done by Seattle Computer Products, with help from Microsoft.

86-DOS is similar enough to CP/M to make you feel at home, but different enough to get you into trouble if you assume that it's really the same. It is conceptually similar, but the differences could be considered departures or enhancements, depending on your point of view. The fact that there are so many good ideas within a framework familiar to the user shows that SCP has some good software people with minicomputer exposure, as well as talented hardware designers.

Typical of the enhancements is the line editor built into the command line interpreter. It uses the DEC VT52 function key escape sequences to permit the last line entered to be edited and resubmitted—just the thing when you make dumb typographical errors and you really don't feel like re-entering the entire line. It's also handy when the next command you are going to enter differs by only a few characters from the last command entered. Also, the file copy utility is memory-resident, which saves you the time required to load PIP. The control characters have essentially the same effect except that a control-N is required to un-toggle the printer after a control-P has started it.

The utility software provided includes a resident 8086 assembler, a line-oriented editor, a CP/M to 86-DOS file converter, a Z-80 to 8086 source code converter and a breakpointing debugger. I did not spend much time with the assembler or line editor, but the editor is just as bad as any other line editor I have attempted to use. The Z-80 to 8086 source code converter is interesting. It does a fairly good job until it gets to special Z-80 instructions like block I/O and some of the extra register functions. At this point you have to code by hand. I cannot attest to the relative efficiency of the 8086 code generated because I'm not sufficiently conversant with its instruction set.

The debugger is as good as any of the general-purpose debuggers to be found in the 8-bit world. It loads only object files (no HEX) and, as the manual points out, it will even trace ROM. Every instruction is traced correctly, unlike most 8080 and Z80 debuggers. It doesn't do anything fancy like using a symbol table, but what good is a program like ZSID when the thing misinterprets the object code? The debugger also includes a disk read and write capability.

All of the development tools are important, but the real thing that makes a new processor go is the availability of high level languages. The 8086 languages are coming on strong and Microsoft was there first with Stand-alone Disk Basic-86. With the conversion chronicled above, SCP became the first manufacturer to offer the full hardware, operating system and high level support of an 8086 on the S-100 bus. Virtually anything that is written for the 8080 Microsoft Basic interpreter will run on the 8086 interpreter, but it will go faster because of the higher clock rate and throughput of the 16-bit machine.

The manuals are oriented toward the experienced micro computerist, particularly one who is graduating from an 8-bit processor in the S-100 world.

Documentation

Before we go on to a comparison of execution times between the 8-bit and 16-bit worlds, a few words are in order about documentation. The folks at SCP have been conscientious in keeping current owners updated with new manuals and releases. Most of the material I received from them had a "Dear User" flavor, giving no indication of preferential status as a reviewer. The manuals are complete, clear and well written, but they are definitely oriented toward the experienced microcomputerist, particularly one who is graduating from an 8-bit processor in the S-100 world. They convey enough information for an experienced person to get the system configured and running, but I think that a relative newcomer or an Apple-wizard would be somewhat bewildered. More examples and pictorials of option switch settings would be helpful.

The one manual in which pictorials are used is the 8/16 RAM manual. Unfortunately, they are a total failure. The artist selected strange trapeziodial directional indicators for the dipswitches which have confused everyone to whom I have shown them.

Comparisons

Aside from those who always have to have the best, newest or fastest computer equipment, there are a limited number of reasons why a user would select a high performance 16-bit system over a high performance 8-bit system. There is no doubt in anyone's mind that the 8086 can move data around faster than even a 6 MHz Z-80, especially when the data path is 16 bits wide. All the standard benchmarks peg the 8 MHz 8086 as having five times the throughput of a 4 MHz Z-80, so all your programs will run five times faster, right?

SCP Review, cont'd...

I wish it was that simple. The real stumbling block is software, not CPU speed. I ran "known quantity" programs that I had written in Microsoft Basic on both machines and found some interesting results. I should point out that I use Basic-80 strictly as a development tool for the Basic Compiler, which represents a plateau of efficiency for 8-bit high level languages, since only PL/I-80 (to the best of my knowledge) produces faster object code. Now, you may object and say that it's unfair to compare a Compiler and an interpreter even when the CPU is five times faster, but we're talking about reasons to buy the 16-bit machine. The software state-of-the-art is a major factor.

First, let's state the facts: Basic-80 is definitely slower than Basic-86. If throughput in executing interpreted Basic programs was the sole criterion, there would be no contest. The second fact is that the Basic Compiler does everything faster than Basic-80, and here again, there is definitely no contest. Its slowest functions, such as string concatenation, are still three or four times faster than the interpreter. Its fastest operations, such as integer arithmetic, are up to twenty times faster than the interpreter.

So when we benchmark the 8086-based interpreter against the compiler, what do we find? We find the compiler still faster in most instances. One exception is string concatenation, which was actually faster than the compiler in the tests I made. This should not be a surprise, because large portions of Basic-86 appear to be translated 8080 code. By no means does this suggest that you shouldn't consider buying an 8086-based system. Can you imagine how fast a Basic-86 compiler or PL/I-86 will be? Or how much less contention will be experienced in a multi-user environment? Once software becomes available that is well optimized for the 8086, the performance will be remarkable.

By the way, for my fellow hardware freaks, there is a switch on the CPU board which limits it to 8-bit data transfers. It allows you to demonstrate the degree of throughput gain you get with the 16-bit data transfers. What with slogging through all the code in the Basic interpreters, I noticed very little difference between the 8086 in 8-bit mode and a 4 MHz Z-80. For that matter, there was no discernible difference in the operation of Basic-80 with a 4 MHz Z-80 and a 6 MHz Z-80. The 8086 in 16-bit mode was sufficiently faster than 8-bit mode to be noticeable, but the difference was not breathtaking. Again, the quality of the software being executed has a major effect on how efficient the processor will appear.

Conclusions

The conclusion I have drawn from living with the 8086 for a number of months is that the SCP hardware is an excellent foundation upon which to build your entry into the 16-bit world. It is solid, reliable stuff and their software works. (This cannot be said of all manufacturers who create their own operating and utility software.) The availability of Basic-86 is a tremendous convenience, one that bodes well for the future. As an OEM/systems integrator, I'm sure that I will use the 8086 in a commercial system in the not-to-distant future.

First, however, I'll need WordStar-86, MDBS-86 and all the other "spoilers" which make life in the 8-bit world so enjoyable. The advent of 16-bit high level language compilers will be the crowning touch. Then, look out DEC, Hewlett Packard, et al. ■