H89-SBC SSD

Background

The H89-SBC SSD is a peripheral board created by Norberto Collado to be installed into his re-implementation of the cherished H89 Z80 based computer. It is based on a concept that originally attempted in the late 1980s by your author, Dan Emrick.

The basic idea was to have a solid state “disk drive” on the H89 that would be fast and silent. There are some large HDOS programs that can take a considerable amount of time to load from the H17 floppy. In addition, programs like a C compiler are not only large but the source code files can also become quite large, thus the desire for fast peripheral storage.

Acknowledgments

The project team for development included Glenn Roberts, Ken Owen, Stanley Web, Norberto Collado and Dan Emrick. In addition, Terry Gulczynski's MTR90 ROM code provided a means for booting from the new SSD. Success of the project was possible only through the insight, experience and ideas of the team members. Thanks to all.

General Description

The H89-SBC SSD uses up to 8 AS6C-4008 static ram chips each with 512 kilobytes of 8-bit storage. These chips occupy a contiguous eight megabyte address space arranged in four banks of two for drive implementation. While not all memory chip sockets must be populated, chips must be used in pairs. A pair of chips provide the storage for one megabyte of random access data – one SSD drive.

The drive implementation consists of four drives of one megabyte each. The base port address for this device is 270Q (0B8H). This is the I/O data port. Address ports are the next three ports. The address ports are write only and are latched to facilitate access to the SSD RAM chip contents.

There is a battery backup on the board that preserves the SSD contents between boots and/or power cycles. To date the operational life of this function has not been determined. This memory maintenance circuit can be powered from either the on-board battery or from an external source.

The board with all logic and storage circuits plugs onto a “USER I/O BOARDS” position on the H89-SBC backplane and can be held in place by an H89-SBC peripheral board bracket.

Initialization and Operation

Currently only HDOS implementation has been completed. As with all mountable storage devices under HDOS, the SSD is handled by a device driver. That driver is called SD.DVD, thus the SSD is accessed by the user through device SD: using SD0:, SD1:, SD2: or SD3:.

For first time use or for complete re-initialization of the four SSD drives, a program called INITS.ABS is run. This creates four drives of one megabyte each and writes a boot track and the required RGT.SYS, GRT.SYS and DIRECT.SYS on each. This initialization program is SSD specific and DOES NOT FOLLOW the normal INIT process. There is no INIT parameter preamble for the SD.DVD driver. The parameters for the SSD drives are contained within the program.

Getting Started

If not already done, insert the static RAM chips into the board in pairs. A minimum of two chips is required. Each "drive" requires a pair of RAM chips.

Use the test program TESTSSD.ABS to verify that the RAM chips are properly decoded and that they are functional. The test does destructive read/write cycles, so run it before any other data or programs are written to the drives. If run after initialization any data will be lost and, the SSD drive(s) must be re-initialized.

Following the basic test of the SSD and immediately after runing TESTSSD.ABS, power down the system and disconnect all power. Wait a few minutes and then power up and reboot. Before attempting any other action, run TESTBAT.ABS. This program expects a specific pattern left behind by TESTSSD.ABS to be present in the SSD memory chips. Successful completion of TESTBAT.ABS without an error indicates that the battery powered SSD memory saving circuit is working.

If data loss is encountered, measure the voltage across the battery terminals. It should be at least 3.6V. If the voltage is low, leave the system powered up and wait until the battery charges to 3.6V on a fully populated board; all RAMs inserted. Then re-run TESTSSD.ABS, power down and up again and re-run TESTBAT.ABS. If problems persist help may be necessary from Norberto or others.

Once the hardware tests are successful, copy SD.DVD and INITS.ABS to the boot floppy and **reboot.** Then run INITS.ABS. It will initialize the four SSD drives (or however many there are chips to support). Because it writes a new RGT.SYS, GRT.SYS and DIRECT.SYS, anything previously written to the device(s) is lost.

Once INITS.ABS is executed, the TESTSSD.ABS and the TESTBAT.ABS should not be run again, unless the battery voltage is below 1.5V.

After the INITS step, the devices may be mounted as normal with a command like:

MOUNT SD0:

The mounted device can be used just like any other mass storage device on an HDOS system, it's just fast.

Boot From SSD

It is may be that the monitor ROM in your H89-SBC does not know about the SSD device. Norberto has a modified version of the H89-SBC MTR90 ROM that is SSD aware. It allows the H89-SBC to boot from a properly configured SSD drive.

Swap the original MTR90 ROM in the H89-SBC for the one modified to recognize the SSD.

Boot from the "normal" boot disk that has SD.DVD on it.

Mount the initialized SSD device. Ex. MOUNT SD0:.

Run SYSGEN.ABS from your boot drive. Follow the SYSGEN instructions to make one of the SSD devices bootable.

Copy SD.DVD to the SSD. Ex. COPY SD0:*.*=SY0:SD.DVD

Reset the SBC.

Type B(oot) S(D)0 and the system should boot from SD0: very quickly.

Note that with this boot from a secondary device, the physical device that was SDx: is now SYx:, as with other secondary boot processes. SD0: is mounted as SY0:. The other SSD drives can be mounted with:

MOUNT SY1:
MOUNT SY2:, etc.

It is useful to note the volume number and label at mount time so that you know which drive has the files you need.

If your "normal" boot device is a floppy and you want to access to floppies, they can be mounted with:

MOUNT SD0: for what would have been SY0:

Setting the CPU clock to a higher speed results in very fast loading and execution of programs. (Of course the 2 MHz clock may be required for programs that count CPU cycles for timing.)

At higher CPU speeds, a modified SY.DVD is required to access floppies without manually setting the speed back to 2 MHz.

Utility Programs

In addition to SD.DVD and INITSSD.ABS and the test programs TESTSSD.ABS and TESTBAT.ABS, there is a program called LABELSSD.ABS. This program can be used to change either an SSD volume number or an SSD label or both. While volume number and labels can be changed at any time, it is probably best to use this program from the "normal" boot device (as is done with INITS.ABS) before booting from an SSD device.

TESTSSD.ABS by Stanley Webb is a very important hardware verification utility. Using it to be sure that the hardware, particularly the RAM section, is functional can save lots of confusion and frustration during setup and first use. Using TESTBAT.ABS will verify that the SSD contents will be preserved across power cycles – an extremely useful feature.