Apple II Technical Notes

Developer Technical Support

UniDisk 3.5
#5: Architectural Differences Between 3.5” Drives

Revised by: Matt Deatherage
November 1988
Written by: Cameron Birse & Mike Askins
October 1986

This Technical Note provides information of interest to those developers writing low-level software for the UniDisk 3.5 and Apple 3.5 disk drives.

Definition of Drives

It is important to understand the differences between Apple’s 3.5” drives if you are considering writing low-level software for use on the Apple II family drives.

UniDisk 3.5 is an intelligent drive, meaning that it has a microprocessor-based controller inside the drive enclosure that communicates with the host computer in an intelligent fashion through the IWM port. The host sends commands to the intelligent controller in the drive and the controller manipulates the drive hardware to read or write, and sends the data back to the host in a “packet” format.

Apple 3.5 Drive is an unintelligent drive that depends on the host computer to manipulate the drive hardware to read and write data to and from the drive. Apple IIGS low-level routines for this drive will be essentially the same as those downloaded to the UniDisk 3.5 controller RAM, except they will reside in the host computer’s memory. New device-specific control calls must be used for the Apple 3.5 Drive.

Tips for Low-Level Drive Access

The following calls are not guaranteed to be compatible in the future; for the highest level of compatibility, avoid disk access at this level.

- **Identifying the drives:** The drives can be identified by first searching for a device that has the SmartPort firmware. After determining that there is a SmartPort device in the machine, perform a `STATUS` call with the `statcode` = $03 (return Device Information Block (DIB)). In the DIB there is a type byte and a subtype byte. The UniDisk 3.5 has a value of $01 for the type byte and $00 for the subtype byte. The Apple 3.5 Drive also has a value of $01 for the type byte, but its subtype byte value is $C0. Be sure to make device-specific calls to ensure drive identification. See SmartPort Technical Note #7, SmartPort Subtype Codes for more details.

- **Special routines:** In the UniDisk 3.5, there is extra RAM space in the controller's memory map for custom read, write and ID routines. These routines can be downloaded to the controller from the host and executed via the SmartPort. With the Apple 3.5 Drive, these special routines reside in the host memory. Equivalent mark and hook tables for the Apple 3.5 Drive, set by control calls through the SmartPort, are supported on the Apple IIGS , but are not guaranteed for all drives and CPUs.
- **IWM hardware differences:** On the UniDisk 3.5, the IWM registers are located in the drive's controller memory starting at $0A00. On the Apple 3.5 Drive, the IWM registers are located in host memory starting at $C0E0 (slot 6 I/O space).
- **Speed differences:** Downloaded code in the UniDisk 3.5 controller runs at slightly under 2 MHz, and the cycle times are regular. The Apple IIGS running at 1 MHz also has regular cycles, however, when running at 2.8 MHz, the timing is complicated by RAM refresh and I/O synchronization times. It is best to avoid timing critical solutions, or be sure to run at 1 MHz for the Apple 3.5 Drive.

As always, in order to promote compatibility between your software and future Apple II systems and to avoid writing utilities which will only work on one kind of drive, you should avoid low-level calls that are specific to a particular device or CPU.

Further Reference

- *Apple IIGS Firmware Reference*