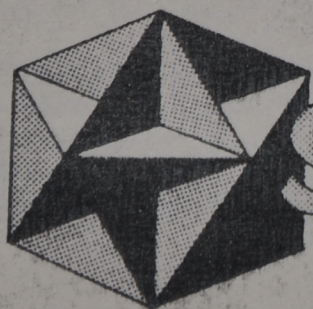


# The Pascal Speedup System

Documentation  
Operation Guide

For Apple II



**STELLATION  
TWO**

P.O. BOX 2342  
SANTA BARBARA, CA. 93120  
(805) 966-1140



THE PASCAL SPEED-UP SYSTEM  
Installation Instructions

Ingredients: 3 diskettes

- 1) the enclosed diskette "RELEASE:",
- 2) your existing Pascal system diskette, containing your original interpreter file SYSTEM.APPLE (version 1.1),
- 3) a blank diskette (not supplied) to use for "backup".

Procedure

- 1) Turn off your Apple and insert THE MILL into one of the peripheral card slots. Then turn power on and boot up your Pascal System.
- 2) Use the Pascal System Filer to make a copy of your existing SYSTEM.APPLE file onto your blank backup diskette. Store this backup in a safe place.
- 3) Insert the RELEASE: diskette and execute the program RELEASE:INSTALL. (If you have a one disk system, you will of course have to remove your system diskette.)
- 4) The program will ask you to specify
  - (a) which slot THE MILL is in, and
  - (b) whether you want an interpreter that will flash "6809" or "6502" in the upper right hand corner of the forty column screen, to indicate dynamically which processor is in control.
- 5) The program will modify the Pascal interpreter in the Language Card, and will inform you that you are now running under THE MILL.
- 6) The program will ask you if you wish to have your interpreter file SYSTEM.APPLE overwritten with the new Mill/Pascal Interpreter. If you do NOT confirm this by typing INSTALL, the program will finish, leaving you running under THE MILL; but the next time you "boot" your system, you will be running under the standard Apple 1.1 interpreter and will want to run this program again to again install the Mill/Pascal interpreter.

If you DO confirm by typing INSTALL, the program will overwrite your file SYSTEM.APPLE so that from now on you will directly "boot" THE MILL. (If you have a one disk system, the program asks you to re-insert your system disk so that this can be done.)



Notes on the Installation Procedure

**HARD DISKS:** the Speedup System works admirably in conjunction with "hard" disks such as the Corvus. However, some of the hard disks use a special "protected" area of the disk to access the Pascal interpreter when booting up; in this case, modifying SYSTEM.APPLE will not guarantee subsequent automatic installation of the Speedup System. You may thus find it most convenient not to confirm "INSTALL" above, but instead to make this short run of the INSTALL program a part of your normal machine boot-up procedure.

**THE FLASHER:** If you have an eighty column board installed, you will not see the "flasher" on your screen, since the Speedup System uses the Apple's display memory directly to save time in flipping the flasher from "6502" to "6809" and back. Please note that even with this technique, selecting the Flasher option degrades performance by some three percent. We recommend you run without the Flasher under normal conditions.

**SLOT PROBLEMS:** the only problem we've seen with installation of the Speedup System is in making a good electrical contact between THE MILL board and your Apple slot. If the installation program displays a message saying that THE MILL is not properly installed in the slot, you should first verify that you typed the correct slot number; if not, just rerun INSTALL. If so, we suggest that you simply turn off the machine and re-try the procedure, installing THE MILL in a different slot. (Although the Speedup System will work with any slot, every slot's metal contacts grip a little differently.)

-----

That's all. You will notice an immediate, overall improvement in computing performance, on all your Pascal programs as well as the compiler, the editor, the filer, and the entire Pascal System (though not extending to input-output). This is because your 6809 processor is doing all the interpretation of Pcodes, including the "floating point" operations associated with Pascal REAL variables; while the 6502 processor continues doing input and output. That's all you need to know, since, apart from speed, your MILL-enhanced Apple Pascal System runs exactly like your original system !

[Copyright 1982 SBProgramming]

["Apple" is a registered trademark of Apple Computer]



THE MILL/PASCAL SPOOLERFeatures

- runs with THE MILL under the Pascal Speedup System
- full use of Pascal System while printing
- works with virtually any printer and interface
- prints any disk file, no size limit
- uses less than 1K bytes memory space
- printing task runs in "background" mode
- preserves full MILL speed on compilations, etc.
- printing task may be monitored, killed
- printing may be started anywhere in text file
- survives user interrupt/system reinitialization

1. LAUNCHING THE SPOOLER

It's as simple as A, B, C:

A. Specifying the input file

From the main Pascal System command level, X(ecute SPOOLER. The program will read in the file SPOOLER.DATA, which must be present, on any disk unit. It will then ask you

Print what text file?

and you should type the name of the file you wish to print from. If you leave off the suffix .TEXT, it will be supplied, unless you specifically terminate your entry with a period. If you just type "return", the program will pick up your workfile.

B. Previewing the text

The program now asks

Preview [filename] ?

If you are sure you have specified the correct file, and you want to start printing right from the beginning of it,



type N or just "return", and proceed to step C.

If you type Y, the program displays a blockful (512 characters) of the text from the designated file. At the top of the screen is the prompt

```
[2,0]: B(lock C(har P(rint A(bort =)
```

If the displayed text is indeed what you want to print, just type P(rint and proceed to step C.

If on the other hand you decide that this is the wrong text file, you can type A(bort and exit back to the Pascal command level.

If this is the correct file, but you want to start printing somewhere other than the beginning of the text file, you can use the B(lock and C(haracter inputs to specify where.

The [2,0] in the upper left corner of the screen indicates your current position in the file: block 2, character 0. (The Pascal System puts special information in blocks 0 and 1 of a text file.) To skip to another block, type B and you get the prompt

```
Block (2..n) ?
```

where the n given is the maximum block number in your file. Type the block number you wish to try, and "return", and you will see what your printing will be like if you decide to start there.

For "fine tuning" within a block, use the C(haracter input; you'll get the prompt

```
Character (0..511) ?
```

and can try various values until you see exactly the text you want to start printing.

The B and C commands are useful for picking up where you left off if you "ditch" your Pascal System and have to re-boot. See section 4 below.

When you finally get positioned to the point where you would like to start printing, just type P and the Spooler will start from there.



### C. Specifying the output unit

Now the prompt

Printer unit number [1,2,6,7,8] ?

appears. TURN ON YOUR PRINTER.

If your printer is installed as the normal Pascal unit number 6 (printer interface card in Apple slot 1) you can just hit "return". Otherwise enter the unit number.

## 2. THE SPOOLER TASK: RESOURCE USAGE

Now if your printer is turned on, you will see it begin to print. But look back to your screen: you see the Pascal main command prompt again! And you will discover that your keyboard is still "alive" to all the standard inputs, and you can C(ompile, E(dit, F(ile, and run your own programs, while the printing is being done.

You will also notice that if you start a C(ompile, the printer will stop printing. Be of good courage: printing will resume at the end of compilation. The Spooler always gives maximum priority to your keyboard commands, so that you won't have to wait any longer than usual for your compilation to finish. Similarly, the printing may stop temporarily while certain other requests you make -- via the Editor, Filer, and your application programs -- are being promptly fulfilled.

In fact, the only "overhead" of the Spooler task is about 900 bytes of memory. It's unlikely that you will notice the loss, out of 65536!

There are however two responsibilities you have as the operator of a dual-tasking machine. For obvious reasons, the Spooler needs two special resources: the printer for output and the specified disk file for input. You must regard these resources as reserved by the Spooler for its duration.

THE PRINTER IS RESERVED: don't turn it off while the Spooler task is running, nor run another program which attempts to print on it. Results will be erroneous, possibly humorous.

THE DISK FILE IS RESERVED: don't remove the disk volume (diskette) containing your input text file, from its disk unit, while the Spooler task is running. If you do, the information transferred to the printer by the Spooler cannot be correct!



This does not mean however that the entire diskette is "off limits" to your continuing work with the Pascal System; your programs, compilations, editions, and so on, may continue to use the diskette containing the input text file, as well as the text file itself -- as long as they don't write other information over its original physical location on the diskette.

### 3. MONITORING THE SPOOLER TASK; KILLING IT

If you wish to see how the Spooler is doing, all you have to do is X(ecute SPOOLER once again. You get a one page display of statistics, including a rough guess as to how many pages remain to print.

At the end of this display you are given the option of typing KILL to discontinue the Spooler task. This is the "clean" way to abort the Spooler; in the following we discuss "dirtier" (and perhaps unintentional) ways.

### 4. IF YOU HAVE TO RE-BOOT

Of course, if you turn off the power to your Apple, the printing will stop. When you turn it back on, your Pascal System will greet you as usual, but your Spooler task will have been forgotten. However, you can restart it gracefully, from the precise character where it left off if desired, as per 1B above, "Previewing the text".

The same situation arises if you hit the RESET button, or type H(alt from the main Pascal System command level. Likewise, if your Pascal program gets into an infinite loop, the only way to get out of it is to hit RESET, and you will then have to restart the Spooler.

Note though that you may without restriction make use of the "break" or "control-shift-P" key to interrupt a running program; the Spooler task will survive and continue after system re-initialization. In the same way, typing I(nitalize from the main Pascal System command level does not crash the Spooler.



## 5. A WORD ABOUT YOUR HARDWARE CONFIGURATION

SPOOLER has been tested with a wide variety of Apple and non-Apple hardware and software, and with numerous printers and interfaces. While it will work on virtually any configuration, timing characteristics can make a difference in its performance. Characteristics and options of the printer interface are particularly important, and the user may benefit from a little experimentation, to get the best "mix" of machine cycles spent in transmission versus attending to the keyboard.

If you experience inadequate response to your keyboard commands while the Spooler is running, it is very likely because your interface is "locking up" too much processor time. Some "soft" printer interfaces employ "handshaking" protocols which cause the processor to loop when the printer is busy. This condition can invariably be cleared up by adjusting the "transmission speed" or "baud rate" of the interface to a value close to, or less than, the printer's actual maximum speed, and setting the printer to that same rate.

If you have an "eighty column board" installed in your Apple, and it does not support type-ahead, you have probably already been annoyed by its occasionally "losing" your keystrokes. This situation can be aggravated, and aggravating, while the Spooler task is running. If your eighty column board does have type-ahead, or if you can upgrade it to include this feature, then you should have no problem with keyboard response.

### The Apple Serial Interface Card

Good results have been obtained using the Apple Serial Interface Card with the standard "P8" PROM installed in the Card.

With the P8A PROM, intended for letter-quality printers, the user may experience poor keyboard responsiveness at certain baud rates, due to the firmware's waiting for acknowledgement from the printer. Setting the rate to something close to, or less than, the printing rate of the printer, produces better results. Thus for example with the Diablo Hyterm, whose top speed is about 43 characters per second, a transmission speed of 300 baud (30 cps) works better than 1200 baud (120 cps).



## **STELLATION TWO PRODUCTS INCLUDE:**

**THE MILL**, including OS9/BASIC #9 with daughterboard/  
documentation

The ultimate operating system. Stellation Two is your exclusive source to bring this fantastic system to the Apple II.

### **OS9/BASIC #9 SOFTWARE ONLY**

For the owners of THE MILL, includes daughterboard/documentation.

### **SPRINTER SOFTWARE**

For Apple Pascal users without our PASCAL SPEEDUP KIT, Sprinter works as a spooler, allowing your printer to run while you continue running your Pascal applications.

## **FOR THE PROGRAMMER**

### **ASSEMBLER DEVELOPMENT KIT: INCLUDING THE MILL**

A full feature Assembler, allows you to write 6809/6502 programs that are DOS 3.3 compatible. Includes complete documentation.

### **McMILL MACRO ASSEMBLER**

A full feature recursive macro, including conditional assembly designed to extend the abilities of THE ASSEMBLER DEVELOPMENT KIT.

### **MUG DEBUGGER**

The complete debugger-disassembler.

### **McMILL ADAPTOR FOR PASCAL (M.A.P.)**

For use with Pascal Speedup Kit and McMill Macro assembler, adapts McMill to run under Pascal (for full utilization of 6809 Pascal).



PLEASE: PASCAL SPEED UP  
WITH REAL NUMBERS AND  
SPOOLER  
RIGHT 1981,1982  
.B. PROGRAMMING

