ATARI.RSC
The Atari Developer's Resource

Vol. IV, Issue 3
June-July 1991

Professional Systems Group Introduces AEGIS Strategic Partner Program

The Professional Systems Group has launched the renaissance of Atari's North American Computing Market with the AEGIS Strategic Partner program.

The AEGIS Strategic Partner program is designed to create a unique relationship between Atari and its key resellers and developers. The relationship nurtures the development of both vertical-market and cross-discipline solutions, resulting in a strong and effective national sales organization.

A key part of the AEGIS Strategic Partner program is the AEGIS Symposium, a clearinghouse for the ideas and information that create powerful vertical-market solutions. The AEGIS Symposium is a two-day product training event with participation from both key resellers and key developers.

On August 5, 1991, the AEGIS Strategic Partner program was kicked off with the very first AEGIS Symposium, held at the Wyndham Garden Hotel, in Sunnyvale, California, and at the Cogswell Polytechnical College in Cupertino, California.

The symposium began at the Wyndham Garden Hotel with an overview of the organization of Atari Computer, given by General Manager Greg Pratt. This was followed by an introduction of the AEGIS Strategic Partner program and its goals and methods, given by Atari Computer V.P. of Sales, Don Mandell. Next came a discussion of the Independent Atari Dealer's Association, given by Bill Rehbock and an overview of the Atari Computer product line, given by Art Morgan.

A training session on the system hardware and software was jointly given by Bill Rehbock and Art Morgan, designed to insure that all of the dealers were familiar with all aspects of the basic operation of Atari ST/TT computers, especially the newer machines.

The afternoon of the first day of the symposium was devoted to a number of seminars covering a variety of topics, including Unix on the TT030, Retail Sales Techniques, Seminar Sales Techniques, the Atari Sales Administration, and Atari Field Service.

Scattered on each of the tables throughout the meeting room were a number of TT030 computers. At the end of the afternoon, a product exposition was given. The participating developers installed their software on the machines and spent the next several hours giving hands-on demonstrations to the dealers.

Day Two of the AEGIS Symposium took place at two different locations. At the Wyndham Garden Hotel, Computer resellers were treated to a number of special training seminars given by the participating developers, including Lexicor, Soft-Logik Publishing, Gribnif Software, ISD, and Goldleaf Publishing.

At Cogswell College, in nearby Cupertino, Music resellers were also treated to a number of special training seminars given by participating developers, but the focus here was on music software, from such developers as Steinberg/Jones, Roland, Dr. T's, C-LAB, Interval, Scorpion, and Hybrid Arts.

Some developers, such as Codehead Software, Step-Ahead, Roland, and Thinkware, gave seminars at both locations.

Computer and Music resellers from all over the country attended the symposium, although the majority were from the west coast region. Future AEGIS Symposiums will be at different locations across the country, giving more dealers and developers a better chance to participate.

If you are a developer andyou would like to become part of future AEGIS Symposiums, please contact Bill Rehbock at (408) 745-2082.

Inside This Issue

ATARI.RSC
The Resource File

PRESIDENT
ATARI COMPUTER CORP.
Greg Pratt (408) 745-2146

TECHNICAL DIRECTOR
Bill Rehbock (408) 745-2082

DEVELOPER TECHNICAL SUPPORT
J. Patton (408) 745-2135
GEnie: ATARIDEV
CIS: 70007,1072

Mike Fulton (408) 745-8821
GEnie: MIKE-FULTON
CIS: 75300,1141

Fax: (408) 745-2094

DEVELOPER ADMINISTRATOR
Gail Johnson (408) 745-2022

SOFTSOURCE ADMINISTRATOR
Dan McNamee (408) 745-2024

Atari Computer Corp.
1196 Borregas Ave.
Sunnyvale, CA 94089-1302

Atari, the Atari logo, TT, and MEGA are trademarks of Atari Corporation.

ATARI.RSC edited by Mike Fulton

ATARI.RSC was created using Pagestream v2.1 on the Atari TT030 computer with Atari TTM194 monochrome monitor, and was printed using Ultrascript on the Atari SLM804 Laser Printer.

Developer Report

Gail Johnson

Don't Be Blue Effective 08/01/91 all developer orders will be shipped U.P.S. Ground Label Service rather than Blue. If you require U.P.S. Blue (2nd day delivery) or U.P.S. Red (next day delivery), contact Gail Johnson for additional cost information before you send in your prepaid order. Please note that the delivery service you select does not affect normal order processing time for Atari. Our processing time is still approximately five days for in-stock merchandise.

Thank You, ST/TT Developers! To all who responded to our requests for reregistration and current product information, we appreciate your cards and letters. Some of you even updated me twice! There are many excellent products out there. Keep up the good work!

Notice to Canadian Developers: If you haven't yet registered with me personally, please do so before September 30,1991. Unclaimed developers will be removed from the database after that date. Contact me at (408) 745-2022 if you did not receive the reregistration materials sent out to you in July. Thank you for your cooperation. ❑

Portfolio Briefs

J. Patton

For Portfolio users in the field there is a keyboard cover from Compucover for the Portfolio. These covers fit over the keyboard and protect the machine from dirt and spills reaching the internals through the keyboard. Covers come in clear and anti-glare versions. Contact Ellie Darden at (800) 874-6391 for orders only within the U.S., or at (904) 862-4448 for information or orders outside the U.S..

Since we can't possibly cover all the industry magazines which may feature articles/reviews on the Portfolio with your software/hardware, we would like to invite you to contact by mail, or fax articles to, Atari Corporation so that we may include this in our file of review materials for potential customers doing research. Mail them to the attention of the Portfolio Developer Coordinator or fax to (408) 745-2088. Those contact points are good for brochures as well.

Any VARs with quantity needs of Portfolios take note. If you can order twenty four per shot, we are offering grade B Portfolios at a price of $150.00. Grade B Portfolios are returns (in many cases unopened) which may have minor scratches on the case. All will be QA'd before leaving Atari and will carry the normal 90 day warranty. This is a limited offer, contact Adam Rabbino at (408) 745-2007. ❑

Documentation Change

There is an error in the documentation for the objc_edit() function as described in the GEM AES manual and in the Rainbow TOS Release Notes.

First, if your documentation refers to an `ob_ednewidx` parameter, cross out all references to it. Second, the `ob_edidx` parameter is supposed to be a pointer to the index to the next character in the raw text string, not the index itself. In the sample call, change it to "`&ob_edidx`". Upon exit from objc_edit(), the index will be updated to reflect the cursor position after whatever editing was done. For those doing their own bindings, int_out[1] will contain this returned value. ❑

Atari Developer News

ATARI.RSC Staff

FSMGDOS Developer's Package

Registered Developers who are interested in programming for FSMGDOS and taking advantage of all the new features it provides will be glad to hear that FSMGDOS programming materials are now available to upon request.

Besides the outline font capabilities of FSMGDOS, support for bezier curve functions is also included.

If you want to receive the FSMGDOS Developer's information, please contact Gail Johnson.

STBook/STylus Documentation

Documentation on the expansion bus on the soon-to-be released STBook and STylus computers is now available upon request. This document discusses how the new expansion bus is different from the expansion connectors on older machines. Please contact Gail Johnson to obtain this document.

New Revision of Pexec Cookbook

A new revision of the ***Pexec Cookbook*** document is now available. If you don't have this document, or the one you have is dated earlier than March 19, 1991, then you may want to get the update.

Besides extensive information on the Pexec() GEMDOS function, there is also valuable info about executable file formats, program flags, and memory allocation. To obtain the new version, contact Gail Johnson.

NEW ACSI DMA Document

The ***Atari ACSI/DMA Integration Guide*** is now available to developers who are working on devices for the ACSI port. This document details all of the ins and outs of using the ACSI DMA port found on all ST models to date. Be warned, however, that this is an intensely technical, hardware-oriented document, and not for the weak of heart, but if you're doing something for ACSI, it's probably just what you've always wanted.

If you are in need of this document, contact Gail Johnson. (Do you notice a trend here, or what?)

New Version of Atari Debugger

A new version of DB is now available. Version 3 of DB features infix notation (meaning that it now understands 2 + 2 instead of having to use + 2 2), the ability for a program to communicate with and send commands to the debugger through imbedded XBIOS functions, and more, including support for GST/Lattice format symbol tables.

If you haven't tried this debugger before now, you're really missing out. The new version is available in the developer sections on GEnie and Compuserve.

Announcements

If you have a new product or information you would like to announce to other developers for the ATARI.RSC newsletter, please mail or fax the information to Mike Fulton at (408) 745-2094. Wherever possible, be sure to include such items as price, availability, and where to call or write.

There is no charge for printing these announcements, but please keep in mind that they are used on a space-available, first-come, first-served basis.

Universal Network System

Application and Design Software announces v1.1 of the Universal NETwork. Universal NETwork v1.1 is a hardware independent software solution that can work with all Atari ST/TT network devices currently on the market.

With Universal NETwork you can access a file server across the network from the GEM Desktop or from within applications just like it was another drive on the system. You can have one file server, or several. Users can share hard disks, printers, and more.

Universal NETwork uses the LAN port on the TT030 and Mega STE, and third party adapters are available that allow you to link up older machines as well.

Universal NETwork is now shipping, at the low price of $219.00 for a two-node setup. (Hardware not included) Additional nodes are $95.00 each.

For additional information contact:

Application Design Software
280 Peach Street
Merlin, OR 97532
(503) 476-0071.

Using the GEM AES Scrap Library, TAKE TWO

Mike Fulton

Since last issue's article about using the AES scrap functions appeared, discussions with various Atari developers has brought forth new considerations which warrant certain changes and enhancements to that article and the standards which it describes. Therefore, that article has been revised, and the revised article is included below. In order to avoid further confusion, the entire article is included, not just those parts which have been changed. The changes mainly concern three things: the format of the specification used with scrp_read() and scrp_write(); the deletion of existing scrap files in the clipboard directory when a new scrap file is written; and where to put scrap files when no clipboard directory is defined. For further details, please read the article.

The GEM AES Scrap Library provides a standard method of allowing data interchange between applications via a clipboard mechanism. At the user's request, the current application will place data from its own document onto the clipboard. Later, from the same application or a completely different one, when the user requests the data from the clipboard, it can be read back into the current document. There are two GEM AES scrap library functions, scrp_read() and scrp_write().

The Clipboard Manager XCONTROL panel module allows the user to set the system's clipboard folder to whatever they want, so that this folder will be specified when the system is booted. The Clipboard Manager CPX is documented separately from this article.

Writing a Clipboard Item

```
WORD sc_wreturn =
scrp_write(sc_wpscrap);
```

The scrp_write() call establishes the directory used to contain clipboard scrap files. A normal sequence of events for an application to put something in the clipboard would be:

1) Do a scrp_read() call to determine the location of the system's clipboard folder. If no existing clipboard folder exists, then it is recommended that your application create a folder named "CLIPBRD" in the root directory of the system's boot drive, and then construct a string containing this path specification.

As far as determining the system's boot drive is concerned, if the system drive map returned by the BIOS call Drvmap() indicates that drive C is available, then you should create the "CLIPBRD" folder on that drive. Otherwise, create the folder on drive A or ask the user where to put it.

2) Do not make assumptions about the format of the string you get from scrp_read(). Always examine the string and insure that you have a valid path specification like "C:\CLIPBRD\", without a filename at the end, and ending with a backslash. Do whatever manipulations are required to get the string into that format. For example, if there is a file specification at the end of the string like "C:\CLIPBRD\SCRAP.TXT", then find the first character following the last backslash in the string and change it to a NULL. If you get a string like "C:\CLIPBRD" without a trailing backslash, then do an Fsfirst() call to check if "CLIPBRD" is a subdirectory, and then append the backslash to the end.

3) An application should delete any existing scrap files in the clipboard directory before writing its own. This is done in order to prevent any possibility of an old scrap file being grabbed by the next application. A application may create its own scrp_clear() function for this purpose. A basic example of such a function might be:

```
void
scrp_clear( scrappath )
char *scrappath;
{
short err
char scrapfile[200];

  strcpy( scrapfile, scrappath );
  strcat( scrapfile, "SCRAP.*" );

  err = Fsfirst( scrappath, 2 );
  while( ! err )
  {
    Fdelete( scrapfile );
    err = Fsfirst( scrapfile, 2 );
  }
}
```

This function would be passed a pointer to a string which contains the current clipboard folder specification, as specified and determined in steps 1 & 2 above.

4) Create and write your clipboard file, using at least one of the standard file formats given below. Clipboard files should be of the form "SCRAP.*" where the filename extension specifies the type of data contained within the file.

The following filenames are reserved for the following file formats. Additional file formats may be added to the list as they appear.

SCRAP.TXT – ASCII only text, with a CR/LF at the end of each line

SCRAP.ASC – ASCII only text, with a CR/LF at the end of each paragraph

SCRAP.RTF – ASCII only text with formatting specified through the Rich Text Format defined by Microsoft

SCRAP.1WP – First Word Plus formatted text

SCRAP.WP – Word Perfect formatted text

SCRAP.GEM – Standard GEM Metafile Graphics Image

SCRAP.IMG – Standard GEM Bitmapped Graphics Image

SCRAP.TIF – Tiff (Tagged Interchange File Format) Graphics Image

SCRAP.EPS – Encapsulated Postscript File

SCRAP.CVG – Calamus Vector Graphic

SCRAP.DIF – Data Interchange Format - Spreadsheet/Database data

These are not the only types of clipboard files that may be written. It is expected that many applications will want to write their own specific file format. In this case, it is hoped that the details of the file format are made available so that it can be supported by other applications. In any case, please notify Atari regarding the file format(s) and filename extension(s) used by your application.

At a bare minimum, a text-oriented program should always write SCRAP.TXT. A vector graphics-oriented program should always write SCRAP.GEM. And a bitmapped graphics-oriented program should always write SCRAP.IMG. However, where possible, your application should write as many formats as it can.

For example, let's say you have a word processor named WRITEIT that wants to create SCRAP.WIT clipboard files containing formatted text in its own format. Besides writing the SCRAP.WIT file for its own format, it might also write SCRAP.TXT, SCRAP.ASC, SCRAP.RTF, SCRAP.1WP and any other text formats that it understands. Likewise, a graphics program would write clipboard files for all of the graphics formats that it understands. A vector graphics program might even write out a bitmapped image for a non-vector graphics program to read.

In some cases, depending on the context, it may be appropriate to write both text and graphics formats. For example, if you are in a graphics program and the user has selected a text object to copy to the clipboard, then the application may want to write a SCRAP.TXT file containing the text from that text object. Or if the user is currently editing a text object, and chooses to paste from the clipboard, then it would be valid for the graphics program to look for a text clipboard file so that it can grab text to put into the text object.

5) If you got a valid clipboard directory from scrp_read() in step 1 and no manipulations were necessary in step 2 in order to get the string into the correct format, then you are done.

If you did not get a directory from scrp_read() and so created the "CLIPBRD" folder yourself, or if you had to manipulate the string in step 2, then you now need to do a scrp_write() call using the string obtained at the end of step 2.

Reading A Clipboard Item

```
WORD sc_rreturn =
scrp_read( sc_rpscrap );
```

The scrp_read() function returns a string containing the clipboard directory specification. The sequence of events for an application to read an item from the clipboard would be:

1) Do a scrp_read() call to get the clipboard directory. If scrp_read() returns zero, then the clipboard folder has not been set since the computer was last reset. If the return value is non-zero, then the clipboard directory specification is returned to you.

Do not skip step 1 because you've already done it once. All steps must be done each time you want to grab something off the clipboard. Do not assume the clipboard directory will not change while your application is running.

The scrp_read() function is defined as possibly returning an error code of zero. However, this function currently always returns a value of 1. Since this may change in future versions of GEM AES, applications should handle the case of the error code even if the programmer does not expect to see it.

2) If scrp_read() returns an error code of zero, then the application should act as though nothing is available on the clipboard.

3) If you get a valid string from scrp_read() then you need to make sure it is in the correct format as described in Step 2 of "Writing a Clipboard Item".

4) Now you should have a path specification like "C:\CLIPBRD\". Now search the clipboard directory for a "SCRAP.xxx" file your application understands. You should first look for the formats you understand that provide the most information and then move to more simple formats if you don't find that. That is, a word processor would want to look for SCRAP.1WP before looking for SCRAP.TXT, because the SCRAP.1WP file would specify information about formatting and text styles that the SCRAP.TXT file would not. ❑

ATARI COMPUTER

Atari ST/TT Q & A

Mike Fulton

Q: What is the value of the _MCH cookie for the Mega STE?

A: For the Mega STE, the _MCH cookie value is 0x00010010. The 0x0001 in the high word means "STE" The low word value of 0x0010 means, "Specifically a Mega STE."

Remember that in most cases, the _MCH cookie is not what you want to look for. If you want to know if a machine supports DMA sound, you should check for the _SND cookie. If you want to know what video modes are available, you should check for the _VDO cookie.

Check the _MCH cookie only for things that aren't represented by other cookies or version numbers. If you want to know if additional serial ports are present, for example, check the _MCH cookie.

Q: How is the 68882 math coprocessor on the TT030 accessed?

A: The 68882 on the TT030 operates as a coprocessor to the main 68030 cpu. There are certain machine language instructions which it understands, but which the main 68030 cpu does not. When one of these instructions is encountered while running a program, the 68882 takes over from the 68030 and processes the instruction. From the program's point of view, it's just like the main cpu understood the instruction all along.

To program for the 68882, you need to either have an assembler which understands the 68881/68882 instruction set, or a high-level language like C or Basic which can generate code for the 68881/68882. See the other questions below for more information.

Q: How is the 68881 math processor on the Mega STE accessed?

A: The 68000 cpu does not support the idea of a coprocessor like the 68030 does, so the 68881 on the Mega STE is used in peripheral mode rather than coprocessor mode. This is a bit slower, but still much faster than not using the chip at all.

Also, the Mega STE's 68881 is used in exactly the same way as the SFP004 add-on board for the Mega ST computer, so programs written with the SFP004 in mind will automatically take advantage of it. Some third party upgrades for the ST and Mega series include a 68881 option which works in the same way.

For full programming details on the Mega STE's 68881 or other peripheral mode 68881 add-ons, see the SFP004 Programmer's Guide. If you do not have this document, contact Gail Johnson.

Q: Will programs written for the SFP004 math processor board for the Mega ST work with the 68882 on the TT030, or vice versa?

A: Because the TT030's 68882 is used as a coprocessor, not as a peripheral, programs written specifically for the SFP004 will not recognize the 68882 in the TT030, and vice versa.

However, it is possible for a program to detect which type of math chip is being used, if any, and take advantage of either type. This uses a bit more memory, but it may be preferable to having separate versions of your program file for each math chip. See the other questions below for more information.

Q: How can a program tell if a math chip is installed?

A: The first thing to do is check the Cookie Jar for an _FPU cookie. This will indicate if a math chip is available, and if so, what kind. The low word of the cookie's value describes software-based floating point math support. Currently this is always zero, but this may change in the future. The high word describes the hardware floating point support installed in the system. This is a bitmapped flag, with meanings as follows:

```
Bit#  Description if = 1

0 -   SFP004 or other peripheral
      mode 68881

1 -   Either 68881 or 68882 fpu,
      unsure which

2 -   If bit 1 = 0, then 68881
      for sure.  If bit 1 = 1
      then 68882 for sure

3 -   68040 built-in fpu.

4-7 - reserved for future
      expansion
```

If your software requires Line-F floating point support (for a 68881 or 68882 as a true coprocessor), you should check the high word for a value equal or greater than 2. If your software requires peripheral mode 68881 floating point support, check bit 0 of the high word.

If the _FPU cookie is not present, it may be because the program is running on an older machine with no cookie jar installed. Refer to the SFP004 Developer's Kit for more information on detecting a peripheral mode 68881.

Q: What programming packages support the 68881 and 68882?

A: Hisoft has several packages which support the 68882 in the TT030, including the Devpac assembler, Hisoft BASIC 2.0, and Lattice C, which is capable of generating code for either the 68881 in the Mega STE or the 68882 in the TT030, or auto-detecting code which will use either type of math chip or software routines if neither is available.

continued on page 8

Portfolio Q & A

J.Patton

Questions from peripheral designers dominate this Q&A column. Thanks go out to Jim Tittsler of Atari Japan for his immeasurable help. Many of these questions can be deduced from the Intel specification.

Q: Does the INT/INTA pin of our 82c59 connect to the EINT/EACK of the Portfolio?

A: Yes, EINT/EACK are the correct places to connect an external interrupt controller. The 82C59 should be programmed to produce a single byte vector when EACK is asserted.

Q: If so, will DTR and DEN respond by cycling low during EACK (low) in the same manner they do during the microprocessor INTA cycle (2 pulses)?

A: Yes, DTR and DEN are exactly the processor's DT/R and DEN signals.

Q: Will EACK respond only to EINT and not to PF on-board interrupts? Any other considerations other than the 40ns delay between EACK and INTA?

A: EACK should only occur for external interrupt acknowledges.

Q: Does the IACK line respond to only on board interrupts or also to external interrupts?

A: IACK happens for ***both*** internal and external interrupt acknowledges. It is a buffered (inverted) version of INTA.

Q: Is it necessary to set up the Peripheral ID if my hardware doesn't use it and the OS doesn't need to know about it?

A: It is not necessary to setup PID's for a peripheral. The OS won't choke if its not there. It also will not take any action to initialize the peripheral on its own if it does not find one of the PIDs it knows about.

Q: How do the port address assignment for a peripherals relate to an IBM PC and what range of addresses am I free to use?

A: The port assignments are unique to DIP. The motherboard uses ports in the 8000H-806FH range. DIP's standard peripherals are in the 807xH range. There is no technical reason why you couldn't use the IBM 02xx-03xx range, but you MUST include the most significant address bits of the I/O port in the decoding, which is something PC peripherals do not typically do.

It would be a good idea to choose places that would be available on a "normal" PC as well.

Address 807Fh should NOT be used, because the OS tries to read the PID there.

The ONLY way port 807Fh should be used is to return a Peripheral Identifier (PID). Any other use of this port is bound to lead to problems. Your peripheral should obviously not return an already assigned PID unless it mimics an existing peripheral.

You should NOT use 807Ch, since it is used for controlling external memory cards and peripherals with on-board ROMs.

8070h-807Bh would be a better choice, with some unassigned PID returned on reads of 807Fh.

Q: When an interrupt occurs, the interrupt vector should be placed on the bus by the peripheral.

A: Correct. When EACK is asserted, the peripheral should drive its desired interrupt vector onto the data bus. Not driving the data bus will result in an undefined vector which can't be relied on.

Q: Can the line IINT be used instead of the line EINT?

A: No. IINT is really only an output.

Q: Regarding vectors, since reserved general use interrupts on the PC are 60H through 66H, then 62H through 66H should be available on the Portfolio, right?

A: The only problem with the 6xH vectors is there is no real mechanism for claiming one... so everybody just assumes the one(s) they happen to want is (are) free. DIP took 60H and 61H as examples. 62H and up is probably safe.

Another possibility if you just need four vectors is to consider using the low ones where PC hardware interrupts are normally mapped. The PC/XT range shouldn't be touched, but the PC/AT range of 70H-77H would seem especially good candidates since it is unlikely for Portfolio class machines to have a second interrupt controller.

In any event, you should use a latch so that software can change the most significant bits of the vector, rather than hard-wiring it into your circuit. ❑

New Online For the Portfolio

EDHOOKS.TXT - Documentation which allows manipulation of the editor functions with your program.

VTLISP.ZIP - Very Tiny Lisp v2.0 for the Portfolio. Modified by B.J. Gleason. ❑

Calendar of Upcoming Events

September 14-15
Southern California Atari Faire, v5.0
Glendale Civic Auditorium
Glendale, CA
For details, contact
John King Tarpinian at
(818) 246-7286

September 21-22
West L.A. Music Expo
Los Angeles Airport Hilton
Los Angeles, CA
For details, contact Don Griffin or
Than Silverlight at (213) 477-1945

October 2-4
Seybold Conference
San Jose, CA
For details contact Bill Rehbock

October 4-5
Audio Engineering Society Expo
New York City
For details, call (212) 661-8528

October 12-13
WAACE Show
For details, leave GEnie Email to
J.D.BARNES, or mail request to:
WACCE Vendor Coordinator
C/O John D. Barnes
7710 Chatham Rd.
Chevy Chase, MD 20815

October 21-25
Fall Comdex '91
Las Vegas Convention Center
& Sands Convention Center
Las Vegas, Nevada
For details, contact Bill Rehbock

November 15-17
Cyber Arts International
Pasadena, CA
For details contact:
CyberArts International
20085 Stevens Creek Blvd.
Cupertino, CA 95014
(408) 446-1105
(408) 446-1088 - fax

November 23-24
Chicaco Atari Computer Show
Chicago, IL
For details, contact:
Larry Grauzas
P.O. Box 8788
Waukegan, IL 60079-8788
(708) 566-0671

Atari ST/TT Q & A
(Continued from page 6)

Laser C, from Megamax, also features support for the 68881. The GNU assembler and GCC C compiler also supports 68881/68882 code generation.

Q: Where can I get programming information on the 68030, 68881, and 68882?

A: I suggest the book "Programming the 68030" by Steve Williams, from Addison-Wesley Publishing. It covers both the 68030 and 68882, but it also notes which parts do not also apply to the 68000 and 68881, making it a good all-around reference. Also highly recommended are the original Motorola programming guides, which should be available at most technical bookstores or direct from Motorola. ❑

Atari Computer Corporation
1196 Borregas Ave.
Sunnyvale, CA 94089-1302

ATARI.RSC
The Developer's Resource