This manual has been prepared using our Editor, Scate.

APULIB™

©Redding Group Inc./30 Side Cut Road/West Redding, CT 06896/(203) 938-3202/Telex 643351

INTRODUCTION

APULIB is a FORTRAN library for use with the Microsoft FORTRAN-80 compiler and Altos computers equipped with an Am9511 arithmetic processing unit (the APU).

APULIB is compatible with all versions of FORTRAN-80, up to and including version 4.42. APULIB is also compatible with LYNX, Redding Group's overlay linker.

APULIB allows use of the APU, the arithmetic processing unit, to speed up arithmetic calculations.

The actual speed increase can be in excess of a factor of ten, however the improvement depends both upon the type of arithmetic operations being performed and upon the ratio of arithmetic to non-arithmetic operations.

There is a significant overhead in loading data into the APU so short operations such as floating point addition are not improved by the same ratio as longer operations such as SQRT.

Some operations, such as input, output, subroutine linkage and byte manipulation, make no use of the APU, so no speed improvement is possible.

A typical FORTRAN program, performing extensive numerical calculations, should run about four to six times faster with APULIB than with Microsoft's FORLIB in single precision and two to three times faster in double precision.

It is assumed that the reader is familiar with the operation of the ALTOS computer, with Microsoft FORTRAN-80 and with the linker, L80, as well as with the CP/M operating system.

DIFFERENT ALTOS MACHINES

Two different versions of APULIB and associated programs are provided due to diffences between ALTOS machines.

The files that end in S, e.g. APULIBS.REL, will operate on ACS8000-1 through ACS8000-4 machines.

The files that end in D, e.g. APULIBD.REL, will operate on all Altos machines except the ACS8000-1.

Notice there is an overlap. A program that uses APULIB that is run on an inappropriate machine will "hang" the machine and one or more resets will be necessary to reboot.

In the discussion that follows the suffixes "S" or "D" will be assumed.

USING APULIB

APULIB is a complete FORTRAN library. It may be used in two ways.

METHOD 1

APULIB may be searched after linking the other program files.

EXAMPLE

```
A>L80 PROG1,PROG2,APULIB/S
  */E  or  /G
```

This will cause two files, PROG1.REL and PROG2.REL, to be linked. The file APULIB.REL will be searched for any unresolved references.

METHOD 2

The file, APULIB.REL, may be renamed "FORLIB.REL" and copied to a disk containing L80.REL. APULIB will then automatically be searched for unresolved references.

EXAMPLE

```
A>L80 PROG1,PROG2/E  or  /G
```

USING APUTEST

The APULIB package includes a file APUTEST.COM which performs a series of arithmetic calculations that can be verified for correct operation of the Am9511. If this program "hangs" or gives incorrect results, then there probably is a fault in the Am9511 or in the associated interface circuitry.

USING TIMING

The file TIMING can be used to determine the speed of most of the operations supported by the Am9511.

The results give an upper limit of the time-per-operation, since some time is used by the routine for looping and displaying the results.

The operation of TIMING is largely self-explanatory. Each operation is executed the number of times indicated and the interval should be timed with a watch.

DIFFERENCES BETWEEN APULIB AND THE MICROSOFT LIBRARY

The principal difference that will be observed by users of APULIB is a reduction of the dynamic range of single precision variables. The magnitude of single precision variables must lie in the range

2.7E-20 to 9.2E18.

Attempts to compute a value greater than the maximum value will result in a diagnostic message and program termination. If the absolute value of a result is less than the minimum, it is replaced by zero.

Double precision variables have the same dynamic range in APULIB as they have in the Microsoft library.

With the exceptions noted below, the precision of APULIB is similar to that of the Microsoft library. Differences will be found, however, in the least significant binary digits for some operations.

The double precision multiply and divide algorithms maintain approximately fifty two bits of precision in the mantissa, so a reduction in precision may be found in comparison with the standard Microsoft library.

Since most double precision library routines, such as DEXP, DLOG, etc., make use of multiply and divide instructions, their precision will also be slightly reduced. This is a result of a decision to trade a slight reduction in precision for a significant increase in speed.

APULIB gives the user the option of reducing the precision of double precision division to about 48 bits in the mantissa by calling the subroutine QIKDIV with a TRUE argument (i.e. CALL QIKDIV(.TRUE.)). This further reduction in precision will result in an even greater gain in speed. A call to QIKDIV with a FALSE argument will restore the more precise division algorithm.

ERROR HANDLING

APULIB will detect certain arithmetic errors. An error message will be generated if the error flag, ERRFLG, is enabled.

An error message will have the form:

```
ERROR 9511xx
```

where the "xx" is given by:

xx	CAUSE
OV	overflow
AL	argument too large for inverse sine, cosine or exp
AN	argument negative for square root or log
DZ	divide by zero

The error flag, ERRFLG, is set by the subroutine call:

```
CALL APUERR(ERRFLG).
```

The value of ERRFLG will determine the behavior of APULIB in case of error such that:

ERRFLG	RESULT
0	no error messages produced
1	all errors reported except for overflow
2	all errors reported.

The default value of ERRFLG is 1.

BUG REPORTING

If you think you have found a bug in APULIB that you want to report to Redding Group, produce the smallest program that exhibits the bug. A 2000 statement program requiring several input files, that "doesn't give the right answer", is of little use to us.

The version number and serial number of APULIB must be included in any bug report or request for help. Calling the subroutine APUVER will cause the version number of the library to be printed.

Not all problems are caused by bugs. Certain numerical computations, such as matrix inversion and integration of differential equations, can result in severe loss of precision. Sometimes the use of a different algorithm or of double precision will improve matters.

APULIB - VERSION 1.62

(c) Redding Group Inc 1982

B (tm) is a FORTRAN library for use with Microsoft's FORTRAN-80
LER on Z80 computers equipped with an Am9511 or an 8231
METIC PROCESSING UNIT (THE APU).

B IS COMPATIBLE WITH ALL VERSIONS OF FORTRAN-80 UP TO AND INCLU-
VERSION 3.44. APULIB IS ALSO COMPATIBLE WITH LYNX (TM), REDDING
'S OVERLAY LINKER.

GURATION

B IS DISTRIBUTED IN A FORM THAT MUST BE CONFIGURED TO OPERATE
YOUR 9511 PORT ADDRESSES. THE CONFIGURATION PROCEDURE IS:

IKE A BACKUP COPY OF THE DISTRIBUTION DISK.

)PY TO THE DISTRIBUTION DISK THE FILES: SUBMIT.COM, XSUB.COM,
IRLIB.REL, INIT.MAC AND LIB.COM (SOMETIMES CALLED LIB80.COM).

AKE ANY MODIFICATIONS THAT MIGHT BE REQUIRED TO INIT.MAC AND
SSEMBLE INIT (SEE TECHNICAL NOTES BELOW).

LACE THE DISK IN DRIVE A AND RUN GENAPU. GENAPU WILL ASK FOR
HE ADDRESSES OF THE CONTROL AND DATA PORTS OF THE 9511. SUPPLY
ACH AS A TWO DIGIT HEXADECIMAL NUMBER FOLLOWED BY A CARRIAGE
ETURN. GENAPU GENERATES A FILE APULIB.REL WHICH WILL REPLACE
HE ARITHMETIC PART OF FORLIB. THE GENERATION PROCEDURE TAKES
EVERAL MINUTES.

JBMIT THE FILE APULIB.SUB. THIS WILL INVOKE THE LIBRARIAN TO
OMBINE APULIB.REL, INIT.REL, FORLIB.REL AND MAR.REL INTO A
OMPLETE LIBRARY, APU.REL. YOU WILL PROBABLY WANT TO RENAME
PU.REL TO FORLIB.REL SO THAT IT WILL AUTOMATICALLY BE USED BY
HE LINKER.

ING

APULIB PACKAGE INCLUDES A FILE APUTEST.FOR WHICH CAN BE LINKED
THE APULIB LIBRARY TO PROVIDE A QUICK TEST OF THE Am9511 CHIP.
LINKING CONVENTION IS:

```
L80 APUTEST/N,APUTEST/G,APU/S
```

PROGRAM EXERCISE WILL PROVIDE DETAILED VERIFICATION OF THE
ATION OF EACH ARITHMETIC OPERATION SUPPORTED BY APULIB.

NG

PROGRAM TIMING CAN BE USED TO DETERMINE THE SPEED OF MANY OF THE
ATIONS SUPPORTED BY THE Am9511.

NICAL NOTES

SHOULD BE AWARE THAT THE Am9511 CHIP CAN BE PUT INTO AN IMPROPER

E BY PARTIAL OR ILLEGAL COMMANDS. THIS SITUATION MUST BE RECTI-
BY A HARDWARE RESET OF THE 9511. IF YOUR HARDWARE PROVIDES A
WARE RESET FACILITY FOR THE 9511, A RESET SHOULD BE DONE AT THE
RT OF EACH PROGRAM. A CONVENIENT METHOD TO DO THIS IS TO PUT THE
ET CODE INTO THE MicroSoft ROUTINE INIT.MAC AND INSTALL IT IN APU