

EXEC OPCOM.COD

Insert program disk & then
Enter program name
?D2:BETA
Enter output file name
?D2:BETA.COD

End Compile
INSERT BASIC DISK&RETURN
Ready

/F Option

Advan BASIC generates code which uses the built-in ATARI floating point routines. The Optimizing Compiler can generate code which uses either the ATARI floating point routines or special high speed routines. There are special routines for floating point add, subtract, multiply, sin, cos, tan, atan, log, sqr, exp, and ^. These routines are used only in the FAST sections of code and will be about two to four times faster than the built-in ATARI routines. Using the high speed routines will increase the compiled program length since their code must be added to the end of the program. To keep the program as small as possible, only those routines actually used will be added to the end of the program. In other words if you don't use TAN, the code for it will not be added. Placing a /F at the end of the program name tells the compiler to add and use the special floating point routines. For example:

EXEC OPCOM.COD

Insert program disk & then
Enter program name
?GAMMA.BAS/F
Enter output file name
?GAMMA.COD

Other Compiler Options

Both the Advan BASIC compiler and the Optimizing Compiler provide jumps around function and subroutine definitions. Consider the following program:

```
100 A@:PRINT FNT
110 SUB A@:PRINT "A"
120 SUBEND
130 DEF FNT=5
140 END
```

When the above program is compiled, a jump to line 130 is inserted at the start of line 110 and a jump to 140 at the start of line 130. This allows the functions and subroutines to be placed anywhere in the program. Many programmers put all of their functions and subroutines at the end of the program and protect them with a GOTO around the entire group, or they put the END command before them. In either case the jumps around the functions and subroutines are not needed and just waste memory. You can