INTRODUCTION TO FLEX-FORTH

FLEX-FORTH 1.0 is a hi-level language that follows the Forth Intrest Group standard. It includes a Forth compiler, intepreter, editor and assembler. This version of FLEX-FORTH requires 16K RAM starting at address $2000. User applications in this arrangement for KIM are saved on cassette and brought back whenever needed by single commands. FLEX-FORTH gives the progammer the power of hi-level languages such as Pascal and Basic, yet almost the speed of machine language, combined with a flexibility and extensibility unattained in nonFORTH languages.

GETTING STARTED

The FLEX-FORTH cassette contains 3 copies of the language, the first recorded at hypertape speed, the last two at normal speed. They all have the ID=01. There should be no problem loading it, but if there is, it is probably due to head misalignment. Try adjusting the volume or tone controls, and if that fails, try the head alignment screws on your recorder. The cold start is at $2000, and the warm start is at $2004. Type 'G' at address $2000 and you should get the response:

```
FLEX-FORTH  1.0
```

Your first command should be

```
EMPTY-BUFFERS   <return>
```

This will clear the tape buffer area (<return> means type the return key on your terminal). To get the "feel" of the Forth stack, try pushing numbers on the stack and printing them with the "dot" word (.) as follows:

```
5   6   .   .  <return>
```

FLEX-FORTH should then print

```
6   5   OK
```

FLEX-FORTH can operate in any number base desired, up to 35, with base 10 being default on cold and warm starts. Changing the number base affects only input and output, as all numbers are stored internally in binary. Words for two bases have already been defined for you, DECIMAL and HEX. Here is an example of number-base conversion, i.e., entering a number in decimal and printing it's hexadecimal equivalent.