tion with respect to the falling liquid than in a refluxing situation. Being able to move the flame independent of the falling drop and stirrer will made the coded routine more useful. It will certainly help when we attempt to combine flame and stirrer together. The code will be a loop of the form: -

```
        Go to flame location: Draw flame
image:  Go to stirrer location:
Draw stirrer image:  Loop back
```

The stand alone flickering flame routine consistent with the previous loop will have to be in the form: -

```
              Go to flame location:  Draw
a flame image:  Loop back
```

The delay is deliberately introduced so that the routine can be tested independent of the stirrer and drop routines to be added later. Those routines will later replace the delay, but it is needed now so that we can be sure that the independent routine will be easily integrated with the others.

Now we cannot draw the same image of a flame each time through the loop or it will not flicker. We could blank out the flame image on alternate passes to make it flicker, but the result would be rather dull. The CHEMUTIL-2 method is to overprint nine different flame images rapidly. The loop will not be of the form: -

```
              Go to flame location:  Draw
flame (1):  Delay:  Go to flame location:
Draw flame (2):  Delay:  etc.
```

After nine flame images have been drawn, the loop starts over again.

Finally, we do not want the loops to be endless. There must be some way to turn off the flame. It could be done by stopping the execution of the code after a predetermined number of times through the loop. More generally useful is to us a keystroke to terminate it. The advantage of that is that the code now allows easy introduction of alternatives depending upon which key is struck. For example, striking the S-key could stop the animation and turn off the flame, while striking the C-key could leave the flame on but start the stirrer, too. The point is not that we intend to use the C-key in that way, but that we allow for that possibility before starting to write code. In a complex animation it is not easy to add additional features after coding without undesirable effects appearing.

A BASIC program to show a flickering flame can now be written. The following one is for Apple II+, IIe or IIc microcomputers and will show the flame on the text screen.

```
        10 Home
        20 For I = 97 to 105
        30 VTAB 10 : HTAB 10
        40 PRINT CHR$(I)
        50  X  -  Peek(-16384):POKE(-16368),0: IF X=211 THEN I = 105:GOT0 80
        60 NEXT I
        70 GOTO 20
        80 VTAB 10 : HTAB 10 : PRINT " "
        90 VTAB 20 : END
```

The flame image is made up of successive images from ASCII(33) through ASCII(41). Line 50 stops the illusion when the S-key is struck and Line 80 extinguishes the flame. If CHEMUTIL-2 is available, BLOAD it and replace line 10 by CALL 25042:PRINT "&". The routine will show a true flame on the graphics screen.

BASIC is not the best language for this routine. A satisfactory illusion is obtained here only because the routine has no other tasks. The only delays are the usual ones as the computer reads and interprets each line. If even a small delay loop is added to simulate an additional task, the illusion is lost. For example, add the line: -

```
        55 FOR K = 1 to 30 : NEXT K
```

Upon execution, the routine shows each character clearly and is unacceptable. Yet this delay is too short to allow the introduction of the stirrer and falling drop loops. We must resort to 6502 machine language code.

The source code shown was generated using the Toolkit Assembler (2) and used CHEMUTIL-2 to print the flame images contained in character set 2. To see this program in action you will need to enter the monitor (CALL-151) and enter the program directly. e.g. 7C10 : A9 OB 8D 00 7C A9 09 etc. Exit the monitor and BSAVE FLAME, A$7C00, L$8E then, with CHEMUTIL-2 in place, run the following BASIC program.

```
        10 CALL 25042 : CALL 31760 : END
```

In the next part of this series, I will discuss and show a listing of a routine which shows a rotating stirrer bar suitable for combining with this flame sequence. Subsequent parts of this series will all require CHEMUTIL-2 for the successful execution of the code.

(1) Bendall, V. "CHEMUTIL-2, A chemistry Programming Utility"; Project SERAPHIM, NSFScience Education; Department of Chemistry, Eastern Michigan University, Ypsilanti, MI 48197, 1985.

(2) "Applesoft Tool Kit"; Apple Computer, Inc., 20525 Mariani Avenue, Cupertino, CA 95014.

BOOK REVIEW COLUMN

This issue's column includes books that cover a variety of computer topics, ranging from chip architecture to campus wide networking, and most readers should find at least one book that is of interest. As usual readers are invited (and urged) to express their opinions about this column. Whether you would like to review a book, suggest a topic area that doesn't receive enough coverage, or just make a general comment, write to Dr. Harry Pence, Professor of Chemistry, SUNY-Oneonta, Oneonta, NY 13820.

CAMPUS NETWORKING

STRATEGIES

Edited by Caroline Arms
Digital Press, Bedford, MA, 1988
336 pages, hardbound, $30.00
Reviewed by Harry E. Pence*

This second volume in EDUCOM's "Strategic Series on Information Technology" is similar in approach to its predecessor, Campus Computing Strategies (reviewed in the December, 1984 issue of the "Newsletter"). Ten institutions that are considered to be leaders in the implementation of computing technology are reviewed, each in an individual chapter, to assess their current status and future plans for computer networking. The universities included are Wesleyan University, Dartmouth College, Carnegie Mellon University, Rensselaer Polytechnic University, Massachusetts Institute of Technology, Stanford University, Cornell University, The University of Michigan, The University of Minnesota, and The Pennsylvania State University.

Even though these schools represent a variety of campus types, many readers of this Newsletter probably will not find a school in this list that could serve as a model for their own campuses. Pioneering on the scale described here requires both deep pockets and extensive prior expertise. Wesleyan University, the smallest institution included, has a combined graduate and undergraduate enrollment of 3,100 and consists of 118 buildings on a 120 acre campus. Many small undergraduate schools may find that their needs are adequately met by much simpler networking systems than any in these examples.

Each chapter is written by administrators who are closely concerned with the implementation of the campus network. The authors provide considerable technical detail without becoming too immersed for the reader with little previous networking background. Perhaps more important, their narratives seem to be unusually candid about the problems that have been encountered. Therefore, these presentations should be useful to a broad audience, networking experts, campus administrators, and users.

Perhaps the most important generalization that can be drawn from these various experiences is the need for careful planning and especially for oversight of the actual construction work. Many of these reports dramatize the problems of being on the "bleeding edge" of technology. Indeed, one author was moved to formulate the First Networking Aphorism, "The sooner you start, the longer it takes." Even though these campuses did possess considerable expertise in computing, many emphasized the importance of a good consultant. This advice was underscored by the reports that at least two of these institutions encountered significant problems with the contractors who installed their networks.

The last three chapters of the book offer concise, but very informative, discussions of protocols and standards, campus wiring, and national networks, three important topics that are frequently confusing to those who are just becoming interested in this networking. The glossary provided at the end will also be very useful for readers who are not yet familiar with this variety of computer jargon.

Exaggeration and hype are not uncommon in describing computer products, but the development of computer networking appears to have been accompanied by more than the average amount of hyperbole. It seems that almost every year for the past decade has been identified by some group or other as "The Year of the Network." As a result, it's been easy to overlook the steady progress that has really been made. This book realistically describes the current status of the field, both the problems and the potential, and so provides an excellent overview of an important area.

The case studies described in this volume indicate that campus networks are becoming a major education resource at many institutions; it seems likely that these changes will have effects far beyond the few institutions that are chosen for the case studies. Although this book will probably be most useful for those who are directly involved in the formulation of campus or department computing plans, it also provides some excellent perspective on the problems and potential advantages of computer networks. It should be considered for purchase on that basis.

*Chemistry Department
SUNY-Oneonta
Oneonta, NY 13820

PROGRAMMING THE 6502

by Rodney Zaks
SYBEX, Berkeley, CA
1983, 408 pages, paperbound, $17.95
Reviewed by William K. Nonidez*

As the new fast generation of microcomputers begin to invade our lives with their vastly enhanced computational and graphics capabilities, it is a temptation to shove our older systems over to a quiet part of the lab and quietly let them die. This may well be a blessing for those of us in education because, if we are clever, we can beg these machines from our more affluent colleges and use them in our own labs and classrooms to teach the fundamentals of computers and microprocessors.

Machines based upon the 6502 microprocessor are ideal for this purpose due to the relative simplicity of 6502 architecture and to the openness of most manufacturers in documenting the architecture of their machines. The most prevalent example of such an instrument is the Apple II series, which is shipped with a detailed reference manual which explains in quite readable detail how the instrument works and even includes detailed ROM listings and entry points. Although these manuals contain sections on the 6502 instruction set and brief descriptions of their structure, most of us can only make sense of this material after we have a sufficient background in the basics. I can strongly recommend Programming the 6502 for obtaining this background.

Programming the 6502 is divided into eleven chapters and nine appendices. Chapter 1 is the obligatory chapter on basic concepts which must be understood before continuing with the rest of the book. These topics include basic elements of programming, information representation and flowcharting. Zaks' treat-