

Listing 2, continued:

```

0076: DC 00096 DFD DCR-1 (FX)
0077: SE 00097 DFD MUL-1 (E)
0078: SE 00098 DFD MUL-1 (UNUSED)
0079: SE 00099 DFD MUL-1 (F)
007A: 10 CA 00100 BPL SETZ ALWAYS TAKEN
007C: 05 00 00101 LDA RCL-X
007E: 05 00 00102 STA RCL
007F: 05 01 00103 LDA RCL-X
0080: 05 01 00104 STA RCL
0081: 05 00 00105 STA RCL
0082: 05 01 00106 LDA RCL-X
0083: 05 01 00107 STA RCL
0084: 05 00 00108 STA RCL
0085: 05 00 00109 ST LDA RCL
0086: 05 00 00110 STA RCL-X
0087: 05 01 00111 LDA RCL
0088: 05 01 00112 STA RCL-X
0089: 05 00 00113 RTZ
008A: 05 00 00114 STAT
008B: 05 00 00115 STAT2
008C: 05 00 00116 LDY #30
008D: 05 00 00117 STAT3
008E: 05 00 00118 INC RCL-X
008F: 05 00 00119 DML INR2
0090: 05 01 00120 INC RCL-X
0091: 05 00 00121 INR2
0092: 05 00 00122 LDAT
0093: 05 00 00123 STA RCL
0094: 05 00 00124 LDY #30
0095: 05 01 00125 STA RCL
0096: 05 00 00126 BDC STAT3
0097: 05 00 00127 LDY #30
0098: 05 00 00128 DEC POP3
0099: 05 00 00129 POPD
009A: 05 01 00130 LDA RCL-X
009B: 05 00 00131 TAY
009C: 05 00 00132 POP2
009D: 05 00 00133 LDA RCL-X
009E: 05 00 00134 STA RCL
009F: 05 01 00135 STY RCL
00A0: 05 00 00136 LDY #30
00A1: 05 00 00137 STY RCL
00A2: 05 00 00138 LDY #30
00A3: 05 00 00139 LDAT
00A4: 05 01 00140 LDA RCL-X
00A5: 05 00 00141 STA RCL
00A6: 05 00 00142 JNR
00A7: 05 00 00143 STDAT
00A8: 05 00 00144 LDA RCL
00A9: 05 00 00145 STA RCL-X
00AA: 05 00 00146 JNR
00AB: 05 00 00147 STPAT
00AC: 05 00 00148 LDA RCL
00AD: 05 00 00149 STA RCL-X
00AE: 05 00 00150 POP3
00AF: 05 00 00151 DCR
00B0: 05 00 00152 DML DCR2
00B1: 05 01 00153 DEC RCL-X
00B2: 05 00 00154 DCR2
00B3: 05 00 00155 RTS
00B4: 05 00 00156 SUB
00B5: 05 00 00157 CBR
00B6: 05 00 00158 LDA RCL
00B7: 05 00 00159 SBC RCL-X
00B8: 05 00 00160 STA RCL-X
00B9: 05 01 00161 LDA RCL
00BA: 05 01 00162 SBC RCL-X
00BB: 05 01 00163 STA RCL-X
00BC: 05 00 00164 TYP
00BD: 05 00 00165 ADC #30
00BE: 05 00 00166 STA RCL
00BF: 05 00 00167 RTZ
00C0: 05 00 00168 ADG
00C1: 05 00 00169 ADC RCL-X
00C2: 05 00 00170 STA RCL
00C3: 05 01 00171 LDA RCL
00C4: 05 01 00172 ADC RCL-X
00C5: 05 00 00173 LDY #30
00C6: 05 00 00174 SEC
00C7: 05 00 00175 DS
00C8: 05 00 00176 LDA RCL
00C9: 05 01 00177 JST
00CA: 05 01 00178 LDA RCL
00CB: 05 00 00179 JST
00CC: 05 00 00180 DMC
00CD: 05 01 00181 DML
00CE: 05 01 00182 BPL
00CF: 05 00 00183 DCM
00D0: 05 01 00184 ADC
00D1: 05 01 00185 STA
00D2: 05 01 00186 TYP
00D3: 05 01 00187 ADC
00D4: 05 01 00188 STA
00D5: 05 00 00189 RTS
00D6: 05 00 00190 BCS
00D7: 05 00 00191 RTS
00D8: 05 00 00192 DP
00D9: 05 00 00193 TAX
00DA: 05 01 00194 LDA RCL-X
00DB: 05 00 00195 DPL
00DC: 05 00 00196 DPL
00DD: 05 00 00197 DM
00DE: 05 01 00198 LDA RCL-X
00DF: 05 01 00199 LDA RCL-X
00E0: 05 01 00200 DML
00E1: 05 00 00201 RTZ
00E2: 05 00 00202 LDA RCL-X
00E3: 05 00 00203 ORA RCL-X
00E4: 05 00 00204 DEC
00E5: 05 00 00205 RTZ
00E6: 05 00 00206 DCM
00E7: 05 00 00207 ASD
00E8: 05 00 00208 DMC
00E9: 05 00 00209 TAX
00EA: 05 00 00210 LDA RCL-X
00EB: 05 00 00211 ORA RCL-X
00EC: 05 01 00212 DMC
00ED: 05 00 00213 RTZ
00EE: 05 00 00214 DML
00EF: 05 00 00215 TAX
00F0: 05 00 00216 LDA RCL-X
00F1: 05 01 00217 AND RCL-X

```

memory locations like 6502 instructions. The main loop at SW16B repeatedly calls the "execute instruction" routine at SW16C which examines one op code for type and branches to the appropriate subroutine to execute it.

Subroutine SW16C increments the program counter (R15) and fetches the next op code which is either a register operation of the form OP REG (2 hexadecimal digits) with OP between hexadecimal 1 and F, or a nonregister operation of the form 0 OP with OP between hexadecimal 0 and D. Assuming a register operation, the register specification is doubled to account for the 2 byte SWEET16 registers and placed in the X register for indexing. Then the instruction type is determined. Register operations place the doubled register specification in the high order byte of R14 indicating the "prior result register" to subsequent branch instructions. Nonregister operations treat the register specification (right-hand half-byte) as their op code, increment the SWEET16 PC to point at the displacement byte of branch instructions, load the A-Reg with the "prior result register" index for branch condition testing, and clear the Y-Reg.

When Is an RTS Really a JSR?

Each instruction type has a corresponding subroutine. The subroutine entry points are stored in a table which is directly indexed by the op code. By assigning all the entries to a common page, only a single byte of address need be stored per routine. The 6502 indirect jump might have been used as follows to transfer control to the appropriate subroutine:

```

LDA #ADRH High order address byte
STA IND+1
LDA OPTBL,X Low order byte
STA IND
JMP (IND)

```

To save code the subroutine entry address (minus 1) is pushed onto the stack, high order byte first. A 6502 RTS (ReTurn from Subroutine) is used to pop the address off the stack and into the 6502 program counter (after incrementing by 1). The net result is that the desired subroutine is reached by executing a subroutine return instruction! [This ironic situation is an example of what is commonly referred to as "cleverness."]

Op Code Subroutines

The register operation routines make use of the 6502 "zero page indexed by X" and "indexed by X indirect" addressing modes to access the specified registers and indirect data. The "result" of most register ops is left

The Art of Computer Programming

Praised by many critics as the best books in their field, The Art of Computer Programming, Volumes I, II and III, are part of a projected seven volume omnibus survey of computer science now being completed by Donald E Knuth.

—Volume I, *Fundamental Algorithms*, begins with a thorough discussion of the mathematics used in computer programming, followed by a treatment of information structures, stacks, arrays, linked lists, dynamic storage allocation, and trees. 634 pp; \$20.95.

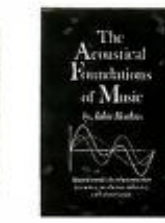
—Volume II, *Seminumerical Algorithms*, is concerned with random numbers, statistical tests, random sequences, as well as arithmetic (floating point and multiple precision), polynomials, and rational arithmetic. 624 pp; \$20.95.

—Volume III deals with *Searching and Sorting*, and as the name implies, the emphasis is on algorithms for sorting, including combinatorial properties of permutations, internal sorting, optimum sorting, and external sorting. Also included is a section on sequential searching, hashing, digital searching, and more. 722 pp; \$20.95.

A hypothetical assembly language called MIX has been developed by the author to illustrate programming examples throughout the series. MIX is easily convertible to other assembly languages.

Prof Knuth writes with style and wit (among many memorable quotes is one from *McCall's Cookbook!*). This classic work belongs on the reference shelf of everyone seriously interested in computer science.

... and Some of the Best for Beginners



—Computer Science — A First Course. The title of this book belies its thoroughness and rigor. *Computer Science: A First Course* is one of the most complete and well laid out computer science books we've seen in quite a while. This 760+ page book is for the serious experimenter who wants a definitive treatment of every aspect of software design and development. Chapters deal with algorithms, flowcharts, looping structures, stepwise decomposition, trees, storage concepts, interpreting and compiling, data processing, numerical processing, character strings, and a discussion of SAMOS, a special hypothetical computer system used throughout the book to illustrate computer science principles. *Computer Science: A First Course* makes an excellent source book. \$16.95, Hardcover.

—Understanding Microcomputers and Small Computer Systems by Nat Wadsworth. What does "hexadecimal" mean? What is Boolean Logic? What's the difference between a PROM and an EPROM? These questions and more are answered in Nat Wadsworth's *Understanding Microcomputers and Small Computer Systems*. You'll learn about the basic operation of a microcomputer, the types of instructions that microcomputers are able to carry out, machine language and higher level language programming, input and output devices, etc. The book concludes with a discussion of system considerations. The combined index and glossary is a useful bonus. If you're a beginner in the personal computing field, get a copy today! This 300 page book is only \$9.95.

—The Acoustical Foundations of Music, subtitled *Musical Sound: a lucid account of its properties, production, behavior, and reproduction*, by John Backus. One of the problems involved in trying to produce music by computer is the scarcity of useful reference material on the basic theory of sound generation and musical acoustics. John Backus has filled this gap with *The Acoustical Foundations of Music*, a readable and informative guide that covers the physiological properties of sound, the ear and its perception of sounds, the effect of acoustic environment, the acoustical behavior of musical instruments, and the various applications of electronics and computers to the production, reproduction and composition of music.

The book assumes some slight knowledge of music on the part of the reader, but covers enough elementary physics and acoustics to orient the reader who has no knowledge of physics.

This clear and lively exposition of music's physical basis will be invaluable to all those experimenters interested in creating music by computer. 312 pages, and only \$9.95 in hardcover.

—Your Home Computer, by James White, is a clearly written nontechnical description of personal computers that requires no prior knowledge of computers or electronics. The emphasis is on understanding; over 100 illustrations are included. Topics include: computing and you; communication inside a computer; computer thought processes; fixed memory; inputs and outputs; peripherals; systems components; how to choose a microcomputer; and so on. *Your Personal Computer* is the ideal book for readers who thought they could never understand how computers work. And the best part is that it's easy and fun to read. Yours for \$6.

Send to: BITS, Inc 70 Main Street Peterborough NH 03458		Check Payment method: <input type="checkbox"/> My check is enclosed <input type="checkbox"/> Bill my MC No. _____ Exp. date _____ <input type="checkbox"/> Bill my BAC No. _____ Exp. date _____	
Name _____		Total for all books checked \$ _____	
Address _____		Postage, 50 cents per book for _____ books \$ _____	
City _____ State _____ Zip Code _____		Grand Total \$ _____	
Signature _____			
You may photocopy this page if you wish to leave your BYTE intact.			
Prices shown are subject to change without notice. All orders must be prepaid. In unusual cases, processing may exceed 30 days.			

Circle 12 on inquiry card.