

Speed Up Calculation Time With a Math Processor Board

For most applications, the TRS-80 is acceptably fast in its mathematical calculations. This month's column is for those applications where greater processing speed is necessary or desirable. I'll describe the construction and operation of a mathematical slave processor for 48K RAM Models I and III.

Adding a slave processor can decrease calculation times by well over 90 percent. You can use the slave processor with Basic programs as described below. But you may find it even more helpful in Assembly-language programs, since you can access most mathematical functions without having to write software routines which, of course, are slower than those for hardware.

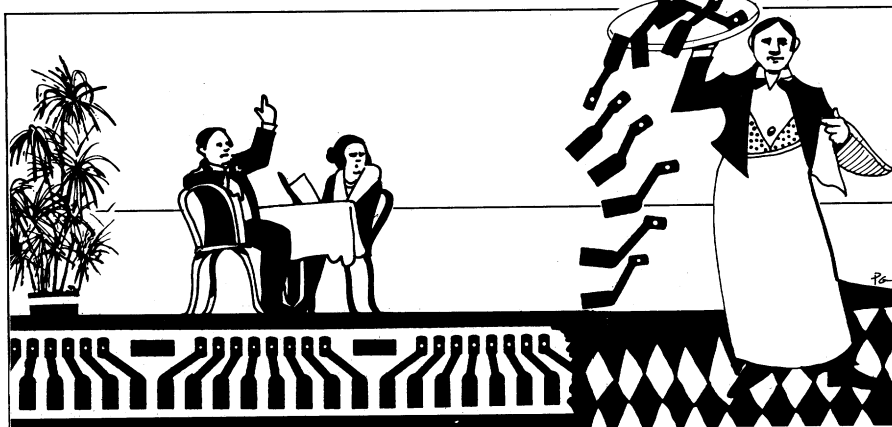
Meet the Intel 8231A and 8232

Intel offers two pin-compatible mathematical slave processors. The 8231A, their arithmetic processing unit, is capable of 16-bit and 32-bit integer calculations as well as 32-bit floating-point calculations. It performs the four basic math functions as well as most trigonometric functions. Throughout this column, my examples will refer to the 8231A.

The 8232, Intel's floating-point processing unit, operates on 32- or 64-bit floating-point numbers, conforms to the proposed IEEE (Institute of Electrical and Electronics Engineers) floating-point format standard, and is limited to the four basic functions.

Integers

Integers represent the most basic data format type and are usually stored as "two's complement" values, where the binary number ranges in value from -2^{N-1} to $+2^{N-1}-1$ (N is the number of bits in the binary number). An 8-bit integer, for example, would have the range -128 to $+127$. In two's complement format, the high-order bit determines whether the value is positive (bit is zero) or nega-



tive (bit is 1). The value zero is represented by all zeros and thus is considered positive.

To determine the decimal value of a positive two's complement number, calculate the value in the normal manner, multiplying each bit value by the corresponding column weight and adding the results. For example, you can find the number 00101011 by calculating right to left: $1*2^0 + 1*2^1 + 0*2^2$ to get 43.

If the number is negative, first complement all of the bits, then add one. This gives the two's complement of the negative number, which is its absolute value. Now find the value of the new number as described above; this tells you of what the original number is a negative.

For example, the number 11101001 is a negative two's complement value. First complement all of the bits to get 00010110, then add one to get 00010111. This value can be found to be 23 using the method described above; therefore, the original value, 11101001, is -23 . Be careful to note that the most negative number (for example, -128 for an 8-bit binary integer) is always its own two's complement.

Binary Floating-Point Numbers

You represent binary floating-point numbers in much the same manner as decimal numbers in scientific nota-

tion. Numbers in scientific notation are in the form $X.YYYY*10^{ZZ}$, where X is a digit from one to nine, inclusive. The Y digits represent further significant digits of the number, and the number of Y digits determines the number's accuracy or precision. The Z digits represent the exponent that determines the dynamic range of the number, that is, its magnitude range.

Notice how the number has only one non-zero digit to the left of the radix (decimal point). This is called "normalization." The number $0.00234*10^{-4}$ is not in proper scientific notation since it isn't normalized. The proper representation is $2.34*10^{-7}$. The only exception to normalization is the value zero.

The number is separated into a mantissa part and an exponent part. The binary mantissa is always normalized. The three formats I'm discussing have this much in common. But from here on, the actual normalization format varies, as do the form of the exponent and the placement of the mantissa sign bit.

A typical binary floating point number format is $0.1XXX*2^{ZZ}$, where $1XXX$ represents the mantissa portion, and ZZ represents the exponent portion. Both the X and Z bits are either zero or one values. Note that binary floating-point numbers are often normalized to the right of the radix (binary point).

TRS-80 Floating-Point

The TRS-80 floating-point format for a single-precision value requires 4 bytes of memory. The high-order byte is the exponent portion of the number, with the remaining 3 bytes making up the mantissa portion. The exponent byte is in excess 128 format. That is, the exponent byte is an absolute binary (*not* two's complement) byte, which is 128 greater than the actual exponent value it represents. Since an 8-bit absolute binary value can range from zero (not used as an exponent value) to 255, the actual exponent value ranges from -127 to +127.

The mantissa consists of the 3 low-order bytes of the 4-byte block, the low-order byte of the block being the low-order byte of the mantissa. Since the mantissa portion of the number is normalized to the right of the radix, a decimal point is assumed to be above the high-order mantissa bit. Since the mantissa is normalized, the high-order bit must be a one for any non-zero value. Because of this, the one is omitted and simply assumed. The high-order bit of the mantissa, instead of holding the 1 bit, holds the mantissa sign bit. This bit determines if the mantissa is positive (bit 0) or negative (bit 1).

The TRS-80 single-precision floating-point format as 24 bits of precision, and an 8-bit exponent range of 2^{-127} to 2^{+127} . The actual decimal range calculates out to approximately $\pm 1.701411 \times 10^{-38}$ to $\pm 1.701411 \times 10^{+38}$. The value zero is represented by a zero exponent byte, without regard to the bits in the mantissa portion of the

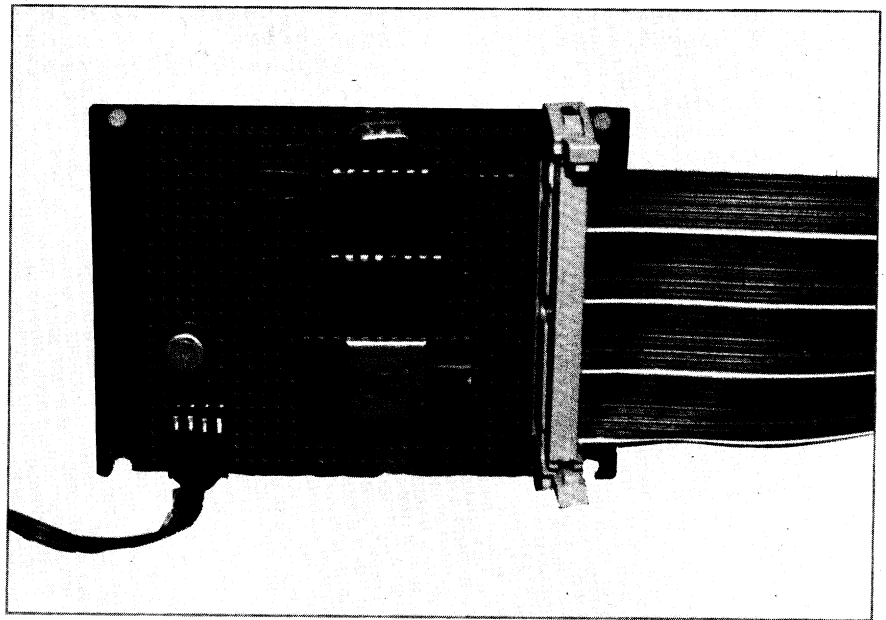


Photo. Math processor board.

number. Figure 1 shows the floating-point format the TRS-80 uses.

Intel 8231A Floating-Point

Like the TRS-80, the 8231A uses 4 bytes to represent its value. Again, the high-order byte is the exponent byte, with the remaining 3 bytes making up the mantissa. As in the TRS-80, the mantissa is normalized to the right of the radix. Unlike the TRS-80, the high-order 1 bit must be present, even though it is a necessary part of any normalized, non-zero number and can be assumed.

Since the high-order bit must be a 1, it cannot be used as a sign bit, as is the

case with the TRS-80 format. The mantissa sign bit, it turns out, is in the exponent byte. Since you move the sign bit, you still have 24 bits of mantissa precision, the same as the TRS-80 format.

The 8231A exponent portion is more different from the TRS-80 format than is the mantissa portion. The high-order bit of the exponent byte is the sign bit for the mantissa; this leaves 7 exponent bits to work with. These remaining exponent bits represent the exponent value in two's complement format. Since there are 7 bits, the exponent value can range from -64 to +63. This is one bit less than the TRS-80 format, giving the 8231A about half the dynamic range of the TRS-80, but the same accuracy (precision).

The value zero is represented by all 32 bits being zero. The decimal numeric range of the 8231A is approximately $\pm 2.7 \times 10^{-20}$ to $\pm 9.2 \times 10^{+18}$ —significantly smaller than that for the TRS-80, although sufficient for most applications. Figure 2 shows the format the 8231A uses.

8232 Floating-Point

The single-precision format of the 8232 comes the closest to the TRS-80 single-precision format, and provides the same precision (24 bits) and roughly the same dynamic range.

The 8232 format may seem some-

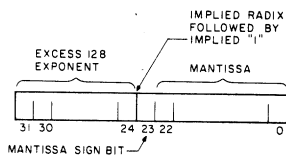


Figure 1. TRS-80's floating-point format.

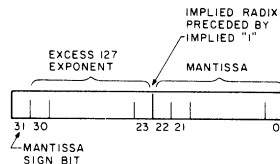


Figure 3. Intel 8232's single-precision format.

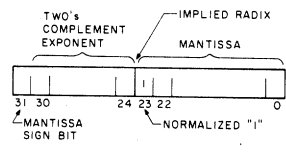


Figure 2. Intel 8231A's floating-point format.

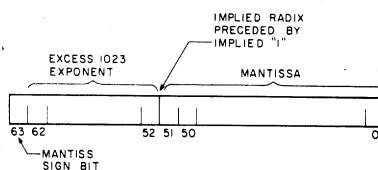


Figure 4. The 8232's double-precision format.

what awkward, since the mantissa and the exponent are not completely separated on a byte boundary. Like the TRS-80 format, the 8232 has an 8-bit exponent. The high-order byte of the 4-byte single-precision block contains most of the exponent portion; however, the low-order bit of the exponent is in the high-order bit position of the next-lower-order byte. The high-order

bit of the high-order byte contains the mantissa sign bit.

The 8232 exponent byte is in excess 127 form, similar to the excess 128 form used in the TRS-80. Like the TRS-80 format, a zero exponent value represents the numeric value of zero, regardless of the value of the mantissa. In the 8232, the exponent value with all bits set to 1 is an invalid condition.

The exponent can range from -126 to $+127$.

The mantissa portion of the 8232 single-precision format consists of the 23 bits of lower order than the exponent bits. The radix is assumed to be between the exponent and the mantissa bits. An assumed 1 bit is to the left of the radix point (instead of the right as in the previous two formats discussed). The decimal numeric range for the 8232's single-precision format is approximately $\pm 1.17 \cdot 10^{-38}$ to $\pm 3.40 \cdot 10^{+38}$ (see Fig. 3).

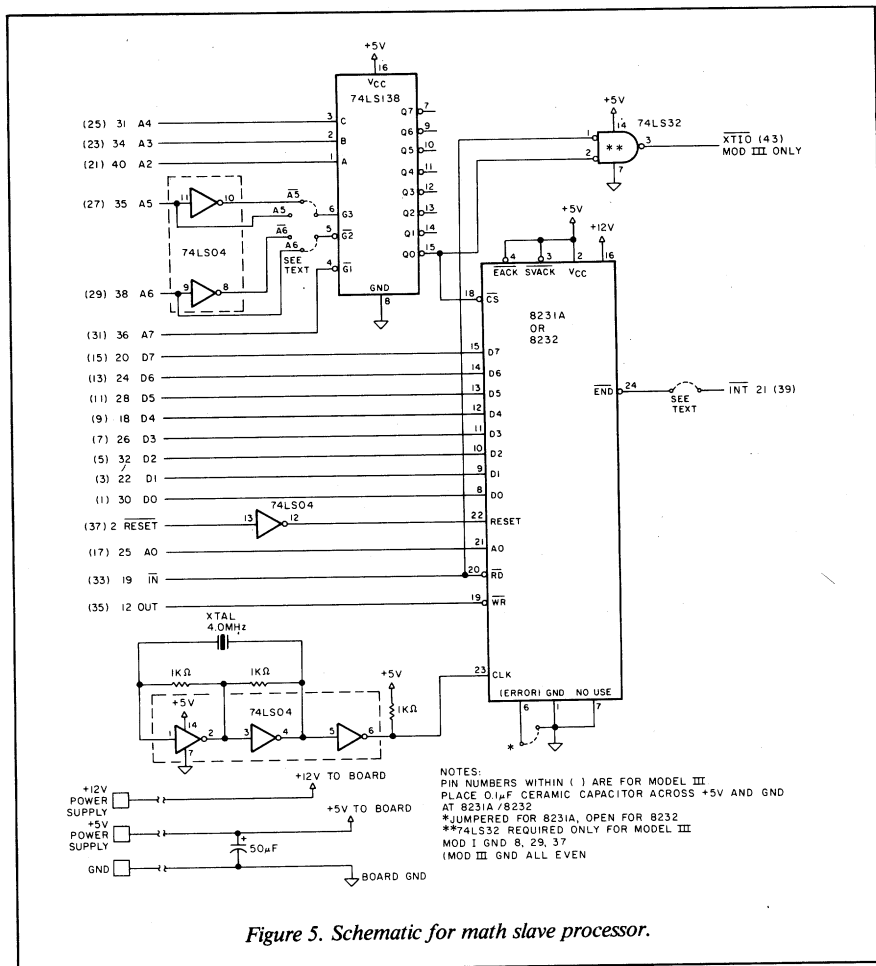
The 8232 double-precision format is a logical extension of its single-precision format, moving from 32 bits to 64 bits (8 bytes). The high-order bit again is the mantissa sign bit. The exponent consists of bits 52 through 62 (portions of the high-order 2 bytes) in excess 1023 form. Again, all zero bits in the exponent represent the numeric value zero, and all 1's represent an invalid condition. There are 52 bits of precision. The decimal numeric range of the 8232's double-precision format is approximately $\pm 2.22 \cdot 10^{-308}$ to $\pm 1.80 \cdot 10^{+308}$ (see Fig. 4).

Note that the 8232's and the TRS-80's double-precision formats differ greatly. The 8232 increases both its precision and its dynamic range from its single-precision format, whereas the TRS-80 increases only its precision (from 24 bits to 56 bits), leaving its dynamic range the same.

Format Conversions

To convert from the TRS-80 single-precision format to the 8231A format, the computer follows these steps:

1. Checks the high-order TRS-80 byte for zero value; if zero, clears all four 8231A bytes, then goes to Step 7.
2. Moves the 3 low-order TRS-80 bytes to the 3 low-order bytes of the 8231A storage area.
3. Sets the high-order bit of the third byte (the high-order mantissa bit).
4. Subtracts 128 from the high-order TRS-80 byte.
5. Sees if the high-order 2 bits are the same; if not, goes to Step 8; otherwise, saves byte in 8231A high-order byte position.
6. Alters the high-order bit of the high-order 8231A byte as necessary to be the same as the high-order TRS-80 mantissa bit (the mantissa sign bit).
7. Ends.
8. Signifies error. Ends.



Quantity	Description	Manufacturer	Part Number	Price
1	74LS138 chip	DIGI-KEY		\$.66
1	74LS04 chip	DIGI-KEY		.41
1	8231A or 8232	Washtenaw		175.00
*1	math processor	Digital Systems		
1	74LS32 chip	DIGI-KEY		.43
3	4.0MHz crystal	DIGI-KEY		2.95
3	1 kΩ resistors (pkg. of 5)	DIGI-KEY		.50
1	0.1 μF capacitor	Radio Shack	272-135	.49
1	100 μF capacitor (pkg. of 2)	Radio Shack	272-1028	.79

*Model III only

See end of article for manufacturers' and suppliers' addresses.

Table 1. Parts list and ordering information.



```

20 DEFUSR0=6HF806
30 A=.2345:B=7:C=0
40 V=VARPTR(A):GOSUB1000
50 V=VARPTR(B):GOSUB1010
60 V=VARPTR(C):GOSUB1020
65 FORI=1TO50000
70 X=USR(12)
75 NEXTI
80 PRINTC:END
1000 V1=INT(V/256):POKE-2047,V1:POKE-2048,V-V1*256:RETURN
1010 V1=INT(V/256):POKE-2045,V1:POKE-2046,V-V1*256:RETURN
1020 V1=INT(V/256):POKE-2043,V1:POKE-2044,V-V1*256:RETURN
    
```

Program Listing 1. VARPTR user function.

Plain Talk About Printers...

<p>DOT MATRIX</p> <p>C. ITOH</p> <p>Prowriter</p>  <p>C. Itoh's Prowriter (120 cps) features 10, 12, & 18 cpi, a proportional/correspondance quality font, double strike, double-width, sub/super scripts, dot graphics (180 x 144 dpi). The Prowriter 2 is the 138 column version.</p> <p>Prowriter \$399.88 Prowriter 2 \$719.88</p>	<p>OKIDATA</p> <p>Microline Series</p>  <p>The Microline 92 (180 cps) is ideal for word processing. It features 10, 12 & 17 cpi, a correspondence font, double-width, emphasis/boldface, sub/super scripts, underlining, pin/friction feed (tractor is optional on the 92) & dot-addressable graphics (120 x 144 dpi). The 93 is the 138 column version. Parallel interfaces are standard; the RS-232C interface is optional.</p> <p>The Microline 84 (132 cpi) is the Step 2 version, featuring 200 cps at 10, 12 & 17 cpi (w/double-width), all with a correspondence mode & dot addressable graphics. Parallel or RS-232C interfaces available.</p> <p>The Microline 82A (120 cps) is a data cruncher. Features 10 & 15 cpi (w/double-width). Dot-addressable graphics are optional. The 83A is the 138 column version.</p> <p>Microline Series SCALL</p>	<p>TOSHIBA</p> <p>P1350</p> <p>A exceptional printer that produces the best near-letter quality we've ever seen. The P1350 prints at 192 cps (100 cps in near-letter quality), & features 132 columns, 10 & 12 cpi (plus double-width) & a near-letter quality font with proportional spacing. It has all the sub/super-scripting features you'd expect, plus dot graphics (180 x 180 dpi). Parallel or RS-232C interface (specify).</p> <p>P1350 \$1759.88</p> <p>Other Dot Matrix Printers We Carry</p> <p>ANADEX</p> <p>DP-9501B \$1119.88 DP-1020B \$1199.88 DP-9625B \$1299.88 WP-5000 \$2349.88</p> <p>DIABLO</p> <p>Series 32 \$1049.88</p> <p>IDS/DATAPRODUCTS</p> <p>Prism 80 \$1079.88 w/4-color \$1439.88 Prism 132 \$1239.88 w/4-color \$1859.88 MicroPrism \$559.88</p> <p>INFOPRINTER</p> <p>Rileman \$349.88</p>	<p>SILVER REED</p> <p>EXP-550/500</p>  <p>The Silver Reed EXP-550 (17 cps) is a 132 column letter-quality printer with 10, 12 or 15 pitch sub/super-script, underlining and true Diablo 1610 emulation, making it compatible with most word processing software. It's friction fed, and it features a page injector, an optional tractor is also available.</p> <p>The EXP-500 (12 cps) is a 100 column letter-quality printer with the same specs as the EXP-550, but slower and without page inject.</p> <p>EXP-550 (Parallel) \$689.88 EXP-550 Tractor \$139.88 EXP-500 (Parallel) \$439.88 EXP-500 Tractor \$129.88</p>
<p>EPSON</p> <p>RX/FX Series</p> <p>The FX-80 (180 cps) has a correspondence font, 10, 12 & 17 cpi, italics, double-strike/width/emphasis & dot graphics, plus a 2K buffer. Friction & pin feed is standard; the adjustable tractor is optional & cost extra. The FX-100 is the 138 column version & includes the adjustable tractor.</p> <p>The RX-80 & RX-80 P/T (100 cps) are upg. add versions of the MX Series.</p> <p>Epson FX/RX SCALL</p>	<p>PANASONIC</p> <p>KX-P1090</p> <p>A smart entry by Panasonic, the KX-P1090 (80 cps) features 10, 12 & 16 cpi, italics, double-width, half-width, enhanced/bold print, dot graphics (120 x 144 dpi), friction/tractor feed & a 4,000,000 character ribbon. The Epson-compatible escape code sequences make it easy to install. Quiet printing & a sharp design make it ideal for home or office. Nationally serviced by Panasonic.</p> <p>KX-P1090 \$339.88</p>	<p>LETTER QUALITY</p> <p>The new, low-speed letter-quality printers are making quality affordable. And the high-speed models are coming down in price too. Still get a dot matrix printer for drafts & as a backup.</p>	<p>NEC</p> <p>Spinwriters</p> <p>The new 3000 Series are slower (20 cps), but they've retained all the quality of the 3500/7700 Series. Uses the same thinblades & ribbons.</p> <p>2010/2030 \$679.88 2050 \$699.88 3530 \$1639.88 3550 \$1989.88 7710/7730 \$2159.88</p>
<p>MANNESMANN TALLY</p> <p>MT-160 L/180 L Spirit</p>  <p>The MT-160 L (180 cps) features 10, 12, 17 & 20 cpi, a correspondence font, italics, enhanced/boldface print, double-width, sub/super scripts & underline, friction/tractor feed. Parallel & RS-232C interface standard. The MT-180 L is the 138 column version.</p> <p>The Spirit (80 cps), Tally's new, low cost draft printer, has 10, 12 & 17 cpi font, italics, friction/tractor feed, and a unique square-wire printhead, 80 columns & parallel only.</p> <p>MT-160 L \$649.88 MT-180 L \$649.88 MT-Spirit \$329.88</p>	<p>STAR MICRONICS</p> <p>Gemini 10X/15X Delta 10/15</p>  <p>The Gemini 10X (120 cps) features 10, 12 & 17 cpi, italics, a correspondence font, double-width, emphasis/boldface print, sub/super scripts, underlining, graphics (120 x 144 dpi), a 1K buffer & friction/tractor feed. The Gemini 15X is the 132 column version.</p> <p>The Delta 10 (160 cps) features both parallel & RS-232C interfaces & an 8K buffer, plus all the 10X features mentioned above. The Delta 15 is a 138 column version.</p> <p>Gemini 10X \$299.88 Gemini 15 \$419.88 Delta 10 \$529.88 Delta 15 SCALL</p>	<p>C. ITOH</p> <p>A10 Starwriter F10 Starwriter Printmaster</p>  <p>The C. Itoh Starwriter (40 cps) features 10 or 12 cpi, sub/super scripts, underlining, 8/8 ipi, 1/48 line feed, 1/120 horizontal resolution. It uses Quime code & Diablo wheels & ribbons. The A-10 Starwriter has the same specs, but it prints slower (20 cps). The Printmaster has the same specs, but it prints faster (65 cps). Both the Tractor Feed and the Sheet Feeder fit all three models.</p> <p>A-10 Starwriter \$609.88 F-10 Starwriter \$1219.88 Printmaster \$1589.88 Tractor Feed \$289.88 Sheet Feeder \$619.88</p>	<p>SMITH-CORONA</p> <p>Messenger</p>  <p>The Memory Correct III Messenger (the full name) is ideal for the home or small office. It combines the features of an electrotypewriter and a letter-quality printer. It features 12 cps, 3 pit./hes (10, 12 & 15), variable line spacing, 10.5 writing line, backspacing & auto-correction. It comes complete with parallel/serial interface.</p> <p>Memory Correct III Messenger \$599.88</p>
<p>PERIPHERALS</p> <p>Smartform 300 \$229.88 Smartform 1200 \$639.88</p> <p>SmartCat 1200 \$499.88 AutoCat 300 \$229.88 AutoCat 1200 \$629.88</p> <p>TRS-80 External Drives (SDDI, 40 TPI) \$199.88</p>	<p>Orders/Prices (800) 343-0726</p>	<p>Other Letter Quality Printers We Carry</p> <p>COMREX</p> <p>CR-1 \$849.88 CR-1 Tractor \$1049.88 CR-2 \$809.88 CR-2 Tractor \$949.88</p> <p>DIABLO</p> <p>820 (RS-232C) \$999.88 630 (PCI) \$1979.88</p> <p>QUIME</p> <p>Sprint 11+ \$1939.88</p>	<p>COMREX</p> <p>CR-1 \$849.88 CR-1 Tractor \$1049.88 CR-2 \$809.88 CR-2 Tractor \$949.88</p> <p>DIABLO</p> <p>820 (RS-232C) \$999.88 630 (PCI) \$1979.88</p> <p>QUIME</p> <p>Sprint 11+ \$1939.88</p>

The only error that can occur is if the exponent value of the TRS-80 number is outside the 8231A's exponent range, which can occur since the TRS-80 uses an 8-bit exponent (the 8231A uses a 7-bit exponent). I implement this algorithm in Z80 Assembly-language code in two of my program listings.

The reverse algorithm to convert from the 8231A floating-point format to the TRS-80 single-precision floating-point format requires the computer to:

1. Check for error code from 8231A. If error, go to Step 8.
2. Check for zero flag bit set in the 8231A. If set, zero the high-order TRS-80 byte, pop the value from the 8231A stack, then go to Step 7.
3. Move all four 8231A bytes into TRS-80 storage locations.
4. Change the high-order mantissa bit (the high-order bit of the third byte) to the same value as the high-order bit of the high-order byte (this is the mantissa sign bit).
5. Make the high-order bit of the exponent (high-order) byte the same value as the next-lower-order bit.
6. Add 128 to the exponent byte.
7. End.
8. Signify error. End.

Constructing the Board

Now that you understand the relevant floating-point formats, you can construct the math processor board. The board design is actually very simple (see Photo); it requires only three chips (four for the Model III), a crystal, three resistors, and two capacitors (see Table 1). You need a +5V (at 100 mA) and +12V (at 95 mA) power supply. Place a 50-microfarad, 10-volt electrolytic capacitor across the +5V power input to supply short-term surge requirements, and an additional 0.1-microfarad ceramic capacitor across the 8231A/8232 +5V line for decoupling (not shown in the Photo).

Figure 5 shows pin numbers for both the Model I and the Model III (in parentheses). The Model III requires an extra circuit, since it must change the direction of an internal bidirectional buffer during every external I/O read. You need the 74LS32 shown in the Photo for the Model III. Of the several jumpers shown on the schematic (Fig. 5), two at the input to the



HIGH TECHNOLOGY AT AFFORDABLE PRICES
THE BOTTOM LINE



MILFORD, NH 03055-0423 □ TELEPHONE (603) 881-9855

Continued on p. 207

PROJECT 80

Program Listing 2. Math processing routine.

```

00100 ;*****
00110 ; MATH PROCESSING ROUTINE FOR INTEL 8231A MATH
00120 ; PROCESSOR.
00130 ;
00140 ; Created by Roger C. Alford
00150 ; October, 1983
00160 ;
00170 ; This program is entered via the Basic USR
00180 ; function. The code specified during the user call
00190 ; determines which math function is to be executed.
00200 ; Currently, only floating point operations are
00210 ; supported. The functions available, and their
00220 ; corresponding code, are shown here:
00230 ;
00240 ; 0 = ADDITION
00250 ; 1 = MULTIPLICATION
00260 ; 2 = INVERSE COSINE
00270 ; 3 = INVERSE SINE
00280 ; 4 = INVERSE TANGENT
00290 ; 5 = SIGN CHANGE
00300 ; 6 = COSINE
00310 ; 7 = EXPONENTIAL
00320 ; 8 = DIVISION
00330 ; 9 = SUBTRACTION
00340 ; 10 = COMMON LOGARITHM
00350 ; 11 = NATURAL LOGARITHM
00360 ; 12 = POWER FUNCTION
00370 ; 13 = SINE
00380 ; 14 = SQUARE ROOT
00390 ; 15 = TANGENT
00400 ;
00410 ;*****
00420 ;
00430 ;
F800 00440 ORG 0F800H
00450 ;
0001 00460 COMAND EQU 01H ;8231A COMMAND PORT ADDRESS
0001 00470 STATUS EQU 01H ;8231A STATUS PORT ADDRESS
0000 00480 DATA EQU 00H ;8231A DATA PORT STACK ADDRESS
0A7F 00490 FNCTN EQU 0A7FH ;TRS-80 'GET USR PARAM.' ROUTINE
0010 00500 FADD EQU 10H ;8231A ADD COMMAND BYTE
0012 00510 FMUL EQU 12H ;8231A MULTIPLY COMMAND BYTE
0006 00520 ACOS EQU 06H ;8231A INVERSE COSINE CMD BYTE
0005 00530 ASIN EQU 05H ;8231A INVERSE SINE CMD BYTE
0007 00540 ATAN EQU 07H ;8231A INVERSE TANGENT CMD BYTE
0015 00550 CHSF EQU 15H ;8231A SIGN CHANGE CMD BYTE
0003 00560 COS EQU 03H ;8231A COSINE CMD BYTE
000A 00570 EXP EQU 0AH ;8231A EXPONENTIAL CMD BYTE
0013 00580 FDIV EQU 13H ;8231A DIVIDE CMD BYTE
0011 00590 FSUB EQU 11H ;8231A SUBTRACT CMD BYTE
0008 00600 LOG EQU 08H ;8231A COMMON LOG CMD BYTE
0009 00610 LN EQU 09H ;8231A NATURAL LOG CMD BYTE
000B 00620 PWR EQU 0BH ;8231A POWER CMD BYTE
0002 00630 SIN EQU 02H ;8231A SINE CMD BYTE
0001 00640 SQRT EQU 01H ;8231A SQUARE ROOT CMD BYTE
0004 00650 TAN EQU 04H ;8231A TANGENT CMD BYTE
0018 00660 POPF EQU 18H ;8231A STACK POP COMMAND
033A 00670 PRCHAR EQU 033AH ;TRS-80 'PRINT CHAR.' ROUTINE
00680 ;
0002 00690 PARAM1 DEFS 2 ;FIRST MATH PARAMETER ADDRESS
0002 00700 PARAM2 DEFS 2 ;SECOND MATH PARAMETER ADDRESS
0002 00710 DEST DEFS 2 ;DESTINATION VARIABLE ADDRESS
00720 ;
F806 F3 00730 MATH DI ;INTERRUPTS ARE DISABLED
F807 CD7F0A 00740 CALL FNCTN ;GET MATH FUNCTION VALUE
F80A 7C 00750 LD A,H ;GET HIGH BYTE OF VALUE
F80B B7 00760 OR A ;BYTE SHOULD BE ZERO
F80C C2FFF8 00770 JP NZ,ERROR1 ;IF NOT, ERROR
F80F 7D 00780 LD A,L ;GET LOW BYTE OF VALUE
F810 FE10 00790 CP 16 ;CHECK FOR VALID VALUE
F812 D2FFF8 00800 JP NC,ERROR1 ;IF NOT, ERROR
F815 CB25 00810 SLA L ;MULTIPLY VALUE BY TWO
F817 116CF9 00820 LD DE, FNCTBL ;GET FUNCTION TABLE ADDR.
F81A 19 00830 ADD HL,DE ;POINT TO FNCTN ADDRESS
    
```

Listing 2 continued

IT'S HERE!

"CMON" is a machine language monitor program. Using single letter commands, the user program can be executed under full control of "CMON". This gives the programmer the option to see the program as each instruction is executed, making "CMON" invaluable for finding those inconspicuous, and sometimes fatal little gremlins haunting your software. "CMON" is easy to use, and will quickly earn a place in your software library.

FEATURES: STANDARD MONITOR COMMANDS + USER PROGRAM EXECUTION WITH REGISTER DISPLAY, RAM, ROM BREAKPOINTS. READ — WRITE OBJECT FILES TO DISK. DISPLAY TO CRT OR PRINTER. AND MORE.

AVAILABLE ON NON-SYSTEM DISK OR CASSETTE. SPECIFY MEDIA TYPE AND M1 OR M3 FORMAT. (Cassette version coming soon)

COST IS \$23.00 ppd. MONEY ORDER OR CASHIERS CHECK. NO PERSONAL CHECKS PLEASE. SEND \$1.00 FOR SAMPLE PRINTOUTS AND COMMANDS SUMMARY. DEDUCTABLE WITH ORDER.

Please send payment or inquiries to:

THREE STATE DATA
14453 TURIN LANE.
CENTREVILLE, VA 22020

✓72

TABLES & SUCH

A line of solid computer furniture of sufficient size to accommodate your computer with space for your working papers. Designed to be lifted apart for easy cleaning, moving or storage.

The hardwood veneered tops of oak or birch are trimmed with select 1 1/2" hardwoods. The basic computer table top is 26" x 48" with the printer tabletop of 18" x 24".

Unfinished, the computer table is \$90.00 and the printer table is \$60.00. Finished, the computer table is \$120.00 and the printer table is \$90.00. Plus shipping.

For further information or to order write or call:

Wooden Hen
4665 Ruhle Rd.
Coleman, Michigan 48618
(517) 465-6441

✓79

ARE YOU TIRED OF HEADACHES AND WATERY EYES FROM STARING AT YOUR TRS-80'S HARSH WHITE VIDEO DISPLAY?

Well, a **FATIGUE FIGHTER** optical filter changes that display to a nice, easy on the eyes, green. It is made of hard 1/8 inch thick acrylic for durability, attaches in seconds without tools, and matches TRS-80® styling. So, invest in some relief, get a **FATIGUE FIGHTER** for your Model I, II, or III. It will probably be one of the most used accessories you will ever buy.

P.S. Available direct from us or at computer stores.

TRS-80 IS A TANDY CORP. TRADEMARK.

TO ORDER:

Send Name & Address Typed or Clearly Printed with Check or Money Order (U.S. FUNDS) for \$14.95 Each, Including Shipping. Canadian Orders Add \$1.00 Each. All Other Foreign Orders Add \$3.00 Each for Shipping. NO CREDIT CARD ORDERS. COD's (U.S. ORDERS ONLY) are \$3.00 Additional per Order & are Accepted by Mail or by Calling 904-378-2494 or 9-5 M-F. Florida Residents Add 5% Sales Tax.

SID

SOUTHERN INNOVATIVE DESIGN
3033-15 N. E. 19TH DRIVE
GAINESVILLE, FL 32601-3326

✓35

— DEALER INQUIRIES INVITED —

THE BYTE GENERAL

3 Sierks Lane
Roslyn Harbor, NY 11576
24 Hour Order Line

Order: 516-625-0920

Technical: 516-338-4083

TRS-80:

Model 4, 64k,

2 Drives, RS-232 \$1549.00

Model 4, 64k, 2 Disk Drive, RS232

Micro Design. MDX-6,

Upgraded. \$1429.95

1 Year Warranty (Parts and Labor)

Model 100 Computers:

Model 100 Portable 8k. \$659.00

Model 100 Portable 24k. \$820.00

Covered by The R/S
90 Day Warranty.

Elephant Diskettes:

EMS #1 SS/SD Soft. \$15.95

EMS #2 SS/DD Soft. \$18.95

EMS #5 DS/DD Soft. \$25.95

Modems:

Signalman Mark I. \$74.00

Signalman 300/1200. \$259.00

Smartmodem 300. \$199.00

Smartmodem 1200. \$479.00

Smartmodem 1200B

IBM Version. \$429.00

Novation J-Cat. \$99.95

Smart Cat 300/1200. \$410.00

Printers:

Prowriter 8510ap. \$340.00

Gemini 10X. \$279.00

Okidata 92ML. \$425.00

Prowriter Ribbons. \$6.00

F10-40cps. \$1125.00

F10-55cps. \$1339.00

M.B.S. Business System:

Invoices Mail List

Accts Payable Accts Receivable

General Ledger Check Register

\$225

Specify TRS-80 Model I, III or 4

Look Around, Find The Best Prices
And The General Will Beat Them.

Dealer Inquiries Invited

TERMS OF SALE: There is a 3% Service Charge for Mastercard or Visa. Orders Under \$30 please add \$2 shipping and handling. Personal checks require two weeks for clearance. Prices subject to change without notice.

TRS80 is a Tandy trademark

382

PROJECT 80

Listing 2 continued

```

F81B 5E      00840      LD      E,(HL)      ;GET LOW ADDRESS BYTE
F81C 23      00850      INC     HL           ;POINT TO HIGH BYTE
F81D 56      00860      LD      D,(HL)     ;GET HIGH ADDRESS BYTE
F81E EB      00870      EX     DE,HL       ;PUT ADDRESS INTO HL
F81F E9      00880      JP      (HL)       ;EXECUTE SPECIFIC FNCTN
                00890      ;
                00900      ;
                00910      ;*** SUBROUTINE CONVRT:
                00920      ; CONVRT gets the Radio Shack floating point value
                00930      ; pointed at by the IX register and converts it to
                00940      ; the Intel 8231A format, and pushes it onto the 8231A
                00950      ; stack.
F820 DD7E03  00960      CONVRT LD      A,(IX+3) ;GET EXPONENT BYTE
F823 B7      00970      OR      A           ;CHECK FOR ZERO VALUE
F824 2009    00980      JR      NZ,CONV2   ;IF NOT, GO ON
F826 D300    00990      OUT    (DATA),A   ;ELSE, PUSH 00 BYTE
F828 D300    01000     OUT    (DATA),A   ;PUSH 00 BYTE
F82A D300    01010     OUT    (DATA),A   ;PUSH 00 BYTE
F82C D300    01020     OUT    (DATA),A   ;PUSH 00 BYTE
F82E C9      01030      RET                    ;DONE - RETURN
F82F DD7E00  01040     CONV2 LD      A,(IX)     ;GET LOW ORDER BYTE
F832 D300    01050     OUT    (DATA),A   ;PUT INTO 8231A
F834 DD7E01  01060     LD      A,(IX+1)  ;GET NEXT ORDER BYTE
F837 D300    01070     OUT    (DATA),A   ;PUT INTO 8231A
F839 DD7E02  01080     LD      A,(IX+2)  ;GET HIGH MANTISSA BYTE
F83C F680    01090     OR      80H       ;SET HIGH BIT
F83E D300    01100     OUT    (DATA),A   ;PUT INTO 8231A
F840 DD7E03  01110     LD      A,(IX+3)  ;GET EXPONENT BYTE
F843 D680    01120     SUB    128        ;PUT INTO REAL FORMAT
F845 4F      01130     LD      C,A       ;SAVE EXPONENT IN C REG
F846 07      01140     RLCA            ;MOVE HIGH BIT INTO CARRY
F847 0600    01150     LD      B,00H    ;CLEAR THE B REGISTER
F849 CB18    01160     RR                    ;PUT CARRY BIT INTO B REG
F84B E680    01170     AND    80H       ;MASK LOW 7 ACC. BITS
F84D A8      01180     XOR    B         ;CHECK FOR SAME BIT 7'S
F84E C204F9  01190     JP      NZ,ERROR2 ;IF NOT, OUT OF RANGE ERR
F851 79      01200     LD      A,C       ;ELSE, GET EXPONENT BACK
F852 E67F    01210     AND    7FH       ;CLEAR HIGH BIT
F854 DDCB027E 01220     BIT    7,(IX+2)  ;CHECK MANTISSA SIGN BIT
F858 2802    01230     JR      Z,NOCHG   ;IF POSITIVE, NO CHANGE
F85A CBF5    01240     SET    7,A       ;ELSE SET BIT FOR NEGATIV
F85C D300    01250     NOCHG OUT    (DATA),A ;PUT EXP. BYTE INTO 8231A
F85E C9      01260     RET                    ;DONE - RETURN
                01270      ;
                01280      ;*** ROUTINE ADD:
                01290      ; This routine causes the 8231A to add the two
                01300      ; provided numbers.
F85F 3E10    01310     ADD    A,FADD     ;GET FLOATING ADD CMD BYT
F861 182E    01320     JR      TWOVAL    ;TWO-VALUED OPERATION
F863 3E12    01330     MULT  LD      A,FMUL ;GET MULTIPLY COMMAND BYT
F865 182A    01340     JR      TWOVAL    ;TWO-VALUED OPERATION
F867 3E06    01350     INVCOS LD      A,ACOS ;GET INV COSINE CMD BYT
F869 1843    01360     JR      ONEVAL    ;ONE-VALUED OPERATION
F86B 3E05    01370     INVSIN LD      A,ASIN ;GET INV SINE CMD BYT
F86D 183F    01380     JR      ONEVAL    ;ONE-VALUED OPERATION
F86F 3E07    01390     INVTAN LD      A,ATAN  ;GET INV TANGENT CMD BYT
F871 183B    01400     JR      ONEVAL    ;ONE-VALUED OPERATION
F873 3E15    01410     CHGSGN LD      A,CHSP   ;GET CHANGE SIGN CMD BYT
F875 1837    01420     JR      ONEVAL    ;ONE-VALUED OPERATION
F877 3E03    01430     COSINE LD      A,COS    ;GET COSINE COMMAND BYTE
F879 1833    01440     JR      ONEVAL    ;ONE-VALUED OPERATION
F87B 3E0A    01450     EXPON  LD      A,EXP   ;GET EXPONENTIAL CMD BYT
F87D 182F    01460     JR      ONEVAL    ;ONE-VALUED OPERATION
F87F 3E13    01470     DIV    LD      A,FDIV  ;GET DIVIDE CMD BYT
F881 180E    01480     JR      TWOVAL    ;TWO-VALUED OPERATION
F883 3E11    01490     SUB    LD      A,FSUB ;GET SUBTRACT CMD BYT
F885 180A    01500     JR      TWOVAL    ;TWO-VALUED OPERATION
F887 3E08    01510     COMLOG LD      A,LOG    ;GET COMM LOG CMD BYT
F889 1823    01520     JR      ONEVAL    ;ONE-VALUED OPERATION
F88B 3E09    01530     NATLOG LD      A,LN     ;GET NAT LOG CMD BYT
F88D 181F    01540     JR      ONEVAL    ;ONE-VALUED OPERATION
F88F 3E0B    01550     POWER LD      A,PWR   ;GET POWER COMMAND BYTE
F891 08      01560     TWOVAL EX     AF,AF'     ;SAVE COMMAND BYTE
F892 DD2A00F8 01570     LD      IX,(PARAM1) ;POINT TO FIRST PARAMETER
F896 CD20F8  01580     CALL   CONVRT    ;CONVERT TO 8231A FORMAT
F899 DD2A02F8 01590     LD      IX,(PARAM2) ;POINT TO SECOND PARAM.
F89D CD20F8  01600     CALL   CONVRT    ;CONVERT TO 8231A FORMAT
F8A0 1814    01610     JR      EXECUT    ;EXECUTE DESIRED FUNCTIN
F8A2 18ED    01620     JR      TWOVAL    ;TWO-VALUED OPERATION
F8A4 3E02    01630     SINE  LD      A,SIN   ;GET SINE COMMAND BYTE
F8A6 1806    01640     JR      ONEVAL    ;ONE-VALUED OPERATION
F8A8 3E01    01650     ROOT  LD      A,SQRT   ;GET SQUARE ROOT CMD BYT
F8AA 1802    01660     JR      ONEVAL    ;ONE-VALUE OPERATION
F8AC 3E04    01670     TANGNT LD      A,TAN   ;GET TANGENT COMMAND BYTE
F8AE 08      01680     ONEVAL EX     AF,AF'     ;SAVE COMMAND BYTE
F8AF DD2A00F8 01690     LD      IX,(PARAM1) ;POINT TO FIRST PARAMETER
F8B3 CD20F8  01700     CALL   CONVRT    ;CONVERT TO 8231A FORMAT
F8B6 08      01710     EXECUT EX     AF,AF'     ;GET COMMAND BYTE
F8B7 D301    01720     OUT    (COMAND),A ;SEND COMMAND TO 8231A
F8B9 DB01    01730     LOOP1 IN      A,(STATUS) ;CHECK FOR COMPLETION
F8BB CB7F    01740     BIT    7,A       ;STILL BUSY?
F8BD 20FA    01750     JR      NZ,LOOP1  ;IF YES, CHECK AGAIN
F8BF E61E    01760     AND    LEH       ;CHECK FOR ANY ERROR
F8C1 C20AF9  01770     JP      NZ,ERROR3 ;IF YES, CALC. ERROR
F8C4 DD2A04F8 01780     LD      IX,(DEST) ;ELSE, POINT TO DEST.
F8C8 DB01    01790     IN      A,(STATUS) ;GET STATUS AGAIN
F8CA CB6F    01800     BIT    5,A       ;CHECK FOR ZERO RESULT
F8CC 280F    01810     JR      Z,RESULT  ;IF NOT, GET RESULT
F8CE DD360300 01820     LD      A,(IX+3),0 ;ELSE STORE 00 EXP BYTE
F8D2 3E18    01830     LD      A,POPF   ;GET STACK POP COMMAND
F8D4 D301    01840     OUT    (COMAND),A ;POP 8231A STACK

```

Listing 2 continued



**SATURN
QUALITY PRODUCTS
WITH
DOWN
TO
EARTH PRICES**

MOD 4 16K	\$ 890
MOD 4 64K	\$ 990
MOD 4 128K	\$ 1090
MOD 4 64K 1 Drive, RS232	\$ 1590
MOD 4 64K 2 Drive, RS232	\$ 1740
MOD 4 128K 1 Drive, RS232	\$ 1690
MOD 4 128K 2 Drive, RS232	\$ 1840
20 Meg Hard Drive, complete	\$ 3890

**UPGRADE
YOUR MOD 4
WITH
SATURN'S
HIGH QUALITY
UPGRADE KITS**

- KIT #1
Disk controller board
RS232 board, cables, brackets.
Connectors, Sound board and instructions,
includes all hardware\$ 400
 - KIT #1 + One Drive\$ 500
 - KIT #1 + Two Drives\$ 600
 - HAYES Smartmodems 300/1200\$ CALL
 - Gemini 10X Printer\$ 285
 - Delta 10 Printer(160 cps)\$ 520
 - MBS (The best business software
package ever!)\$ 250
- All types of cables made up to order. Bare Disk drives are also available. Call for our FAST Upgrade services. Everything we sell is guaranteed or your money back.

Dealer Pricing available, call or write. Look in **HOT COCO** for all our Color computer products. If you don't see what you're looking for call us.

AVAILABLE FROM:  
Saturn Electronics Inc.
62 Commerce Drive
Farmingdale, NY 11735
(516) 249-3388 ✓277

PROJECT 80

Continued from p. 204

74LS138 select the address where the 8231A or 8232 appears in I/O space to the TRS-80. The dotted lines show the combination I used here and in the upcoming program listings. The four combinations give addresses as in Table 2.

The slashes following the address bit represent the inverted (barred) signals on the schematic. I assigned four addresses to each combination above, although the math processor only needs two. The math processor is therefore "double addressed" in the selected addressing space; that is, you can access it by either the lower two addresses or the upper two addresses (or a combination thereof).

The jumper between pin 6 of the 8231A/8232 and ground should be in place when you use the 8231A and removed when you use the 8232. This is

G2	G3	Address Range
A6/	A5/	40 hex-43 hex
A6/	A5	60 hex-63 hex
A6	A5/	00 hex-03 hex
A6	A5	20 hex-23 hex

Table 2. Address ranges for the math processor.

the only hardware interface difference between the two devices. Pin 6 is not used on the 8231A and must be tied to ground. The 8232 uses that pin as an error output; it can generate a special interrupt if other hardware is available to support it.

The final jumper, between the END/ output of the 8231A/8232 and the TRS-80 interrupt line, goes active (low) at the completion of its command processing, and can be used to alert the TRS-80 when it's ready to be serviced. This is great for applications where the Z80 can do other useful work while the math processor cranks away, but can cause problems if not properly handled.

If you don't intend to use the interrupt feature, don't hook it up. I did hook up my interrupt line, but the upcoming software does not use the interrupt feature. When the interrupt (END/) line goes active, any access to the math processor forces it inactive.

The crystal on the schematic (Fig. 5)

**"Any other
Pascal is
too much
hassle!"**

Picture this: you want to make a change in a 1000 line Pascal program. You read the source code from disk into a full screen editor and make your changes. You type control Q to quit the editor and R (for run). In 15 seconds, without further disk access, your program has compiled and is executing.

Pascal 80 is so interactive and easy to use that many people have given up Basic completely, even for hasty ten line programs! We do trade some power for interactivity, but at this price you can use **Pascal 80** as a "front" end for another system to have both ease of use and power. Many of our users develop and test their software with **Pascal 80**, then recompile our source code on another system.

Thousands of satisfied users worldwide! (We have a liberal return policy and have had only 2 returns per thousand copies sold.) Many top universities use **Pascal 80** to teach Pascal. (Names on request)

Features — Standard Pascal, writes ASCII source, p-code and .com files. Full screen editor, compilation from memory or disk, include function. Pointer variables are addressable (like C). Limitations — Variant records, with, and page not implemented. Mark and release instead of Dispose.

**PASCAL 80CPM —
Special introductory price — \$79.**
Requires CP/M 80. 8 inch SSSD, Apple CP/M, Xerox, IBM, Osborne SD formats available. Call for information on other formats. Free brochure.

**NEW CLASSICS
SOFTWARE** ✓255

 239 Fox Hill Road
Denville, NJ 07834
201-625-8838 
See our other ad for TRSDOS version

PROJECT 80

shows a 4 MHz value. There are various speed ranges available for the 8231A/8232 math processors, ranging from 2.5 MHz to 6 MHz. Only the crystal frequency needs to be changed to use a different speed math processor.

To work properly, the math processor must get a reset signal before you use it. You can guarantee this by turning on the power to the math processor board before turning on the computer power, or by pressing the TRS-80 reset button while both are on.

Operation

A hardware math processor is most beneficial when used strictly with Assembly-language programs, since it requires no format conversions, takes up no Basic (or other language) overhead, and since all of the common mathematical functions are available (when using the 8231A), without requiring any software math routines. You save programming time, and attain much faster speeds than executing software math routines.

But since it's likely that you use programs written in Basic, you need a way for them to access the math processor. Do this through the user (USR) function.

First create a way to pass variables to and from the Basic program. Basic provides the VARPTR function to find the addresses of specified variables. You can POKE the desired variable addresses into predetermined memory locations (reversed by the Assembly-language routine) so that the Assembly-language routine knows where to find its operands. In Program Listing 1, the variables A, B, and C are used: A or A and B are the operand(s); C is the result.

My Assembly-language routines begin at location 0F800 hex (in a 48K RAM system) reserving more memory than necessary. You can shrink the routines as well as place them at the top of memory for other memory configurations; make sure memory size is set properly to reserve the used space. I set memory size to 63487 for my routines as addressed. The six memory locations reserved for the operands are the first six in the reserved area: 0F800 hex, 0F802 hex and 0F804 hex, for the first Assembly-language routine, and the first 16 locations for the second routine.

To set up Basic to access the user function, follow line 20 of Listing 1 for a disk-based system, or POKE the proper address into locations 16526 and 16527 for a non-disk system. For the two Assembly-language routines below, the first is at 0F806 hex, requiring you to POKE the values 6 and 248, respectively. The second Assembly-language routine, starting at 0F810 hex, requires that you POKE the values 16 and 248, respectively.

The first Assembly-language routine (Program Listing 2) converts the TRS-80 operand(s) into 8231A for-

Listing 2 continued

```

F8D6 DB01 01850 LOOP2 IN A, (STATUS) ;CHECK FOR COMMAND DONE
F8D8 CB7F 01860 BIT 7,A ;DONE?
F8DA 20FA 01870 JR NZ,LOOP2 ;IF NOT, KEEP CHECKING
F8DC C9 01880 RET ;DONE - RETURN
F8DD DB00 01890 RESULT IN A, (DATA) ;GET EXPONENT BYTE
F8DF 4F 01900 LD C,A ;SAVE VALUE TEMP.
F8E0 CB7F 01910 BIT 7,A ;CHECK FOR NEG. MANTISA
F8E2 DB00 01920 IN A, (DATA) ;GET HIGH MANTISSA BYTE
F8E4 2002 01930 JR NZ,SKIP ;IF BIT SET, OK
F8E6 CBBF 01940 RES 7,A ;ELSE, RESET HIGH MANT BT
F8E8 DD7702 01950 SKIP LD (IX+2),A ;STORE HIGH MANT BYTE
F8EB 79 01960 LD A,C ;GET EXPONENT BYTE
F8EC 07 01970 RLCA ;ROTATE LEFT ONE BIT
F8ED CB2F 01980 SRA A ;DUPLICATE HIGH BIT
F8EF C680 01990 ADD A,128 ;MAKE EXCESS 128 FORM
F8F1 DD7703 02000 LD (IX+3),A ;SAVE EXPONENT BYTE
F8F4 DB00 02010 IN A, (DATA) ;GET NEXT RESULT BYTE
F8F6 DD7701 02020 LD (IX+1),A ;SAVE IN RESULT AREA
F8F9 DB00 02030 IN A, (DATA) ;GET RESULT LSB
F8FB DD7700 02040 LD (IX),A ;SAVE IN RESULT AREA
F8FE C9 02050 RET ;DONE - RETURN
02060 ;*
02070 ;*** ERROR ROUTINES:
F8FF 2116F9 02080 ERROR1 LD HL,ERMSG1 ;POINT TO ERR MSG #1
F902 1809 02090 JR PRRTN ;PRINT AND RETURN
F904 E1 02100 ERROR2 POP HL ;POP ADDR FROM STACK
F905 2131F9 02110 LD HL,ERMSG2 ;POINT TO ERR MSG #2
F908 1803 02120 JR PRRTN ;PRINT AND RETURN
F90A 2154F9 02130 ERROR3 LD HL,ERMSG3 ;POINT TO ERR MSG #3
F90D 7E 02140 PRRTN LD A,(HL) ;GET CHARACTER
F90E B7 02150 OR A ;CHECK FOR ZERO BYTE
F90F C8 02160 RET Z ;IF ZERO, DONE
F910 CD3A03 02170 CALL PRCHAR ;ELSE, PRINT THE CHAR.
F913 23 02180 INC HL ;POINT TO NEXT CHAR.
F914 18F7 02190 JR PRRTN ;LOOP AGAIN
02200 ;
02210 ;
F916 0D 02220 ERMSG1 DEFB 0DH
F917 2A 02230 DEFB '*** INVALID COMMAND CODE'
F918 0D 02240 DEFB 0DH
F919 00 02250 DEFB 00H
F91A 0D 02260 ERMSG2 DEFB 0DH
F91B 2A 02270 DEFB '*** PARAMETER VALUE OUT OF RANGE'
F91C 0D 02280 DEFB 0DH
F91D 00 02290 DEFB 00H
F91E 0D 02300 ERMSG3 DEFB 0DH
F91F 2A 02310 DEFB '*** CALCULATION ERROR'
F920 0D 02320 DEFB 0DH
F921 00 02330 DEFB 00H
02340 ;
F922 5FF8 02350 FNCTBL DEFW ADD ;ADDITION ROUTINE
F923 63F8 02360 DEFW MULT ;MULTIPLICATION ROUTINE
F924 67F8 02370 DEFW INVCOS ;INVERSE COSINE ROUTINE
F925 6BF8 02380 DEFW INVSIN ;INVERSE SINE ROUTINE
F926 6FF8 02390 DEFW INVTAN ;INVERSE TANGENT ROUTINE
F927 73F8 02400 DEFW CHGSGN ;CHANGE SIGN ROUTINE
F928 77F8 02410 DEFW COSINE ;COSINE ROUTINE
F929 7BF8 02420 DEFW EXPON ;EXPONENTIAL ROUTINE
F92A 7FF8 02430 DEFW DIV ;DIVISION ROUTINE
F92B 83F8 02440 DEFW SUB ;SUBTRACTION ROUTINE
F92C 87F8 02450 DEFW COMLOG ;COMMON LOG ROUTINE
F92D 8BF8 02460 DEFW NATLOG ;NATURAL LOG ROUTINE
F92E 8FF8 02470 DEFW POWER ;POWER ROUTINE
F92F A4F8 02480 DEFW SINE ;SINE ROUTINE
F930 A8F8 02490 DEFW ROOT ;SQUARE ROOT ROUTINE
F931 ACF8 02500 DEFW TANGNT ;TANGENT ROUTINE
02510 ;
F806 02520 END MATH
    
```

```

10 DEF FN ACS(X)=-1*ATN(X/SQR(-1*X*X+1))+1.5708
25 PI=3.14159
30 A=2.445:B=1.44556:C=2.5:D=6:E=.456:F=2.456:G=15
65 FORI=1TO1000
70 Z=A[4+C*TAN(B*4)+2*PI*(SIN(D[7]+FNACS(E)+EXP(F))+SQR(LOG(G))]
75 NEXTI
80 PRINTZ:END
    
```

Program Listing 3. A complex equation.

PROJECT 80

Program Listing 4. Modification of math processing routine.

```

00100 ;*****
00110 ; MATH PROCESSING ROUTINE FOR INTEL 8231A MATH *
00120 ; PROCESSOR. *
00130 ; *
00140 ; Created by Roger C. Alford *
00150 ; October, 1983 *
00160 ; *
00170 ; This program is entered via the Basic USR *
00180 ; function call. It executes the function: *
00190 ; Z=A**4+C*TAN(B*4)+2*PI*(SIN(D**7)+ACOS(E)+EXP(F)) *
00200 ; +SQR(LN(G)) *
00210 ; *
00220 ; Where ** is 'to the power of', LN is the natural *
00230 ; log function, and ACOS is the inverse cosine *
00240 ; function. *
00250 ; *
00260 ;*****
00270 ;
00280 ;
F800 00290          ORG      0F800H
00300 ;
0001 00310  COMAND  EQU      01H    ;8231A COMMAND PORT ADDRESS
0001 00320  STATUS EQU      01H    ;8231A STATUS PORT ADDRESS
0000 00330  DATA  EQU      00H    ;8231A DATA PORT STACK ADDRESS
00A7F 00340  FNCTN  EQU     0A7FH   ;TRS-80 'GET USR PARAM.' ROUTINE
0010 00350  PADD   EQU      10H    ;8231A ADD COMMAND BYTE
0012 00360  FMUL  EQU      12H    ;8231A MULTIPLY COMMAND BYTE
0006 00370  ACOS   EQU      06H    ;8231A INVERSE COSINE CMD BYTE
0005 00380  ASIN   EQU      05H    ;8231A INVERSE SINE CMD BYTE
0007 00390  ATAN   EQU      07H    ;8231A INVERSE TANGENT CMD BYTE
0015 00400  CHSF   EQU      15H    ;8231A SIGN CHANGE CMD BYTE
0003 00410  COS    EQU      03H    ;8231A COSINE CMD BYTE
000A 00420  EXP    EQU      0AH    ;8231A EXPONENTIAL CMD BYTE
0013 00430  FDIV  EQU      13H    ;8231A DIVIDE CMD BYTE
0011 00440  FSUB   EQU      11H    ;8231A SUBTRACT CMD BYTE
0008 00450  LOG    EQU      08H    ;8231A COMMON LOG CMD BYTE
0009 00460  LN     EQU      09H    ;8231A NATURAL LOG CMD BYTE
000B 00470  PWR   EQU      0BH    ;8231A POWER CMD BYTE
0002 00480  SIN    EQU      02H    ;8231A SINE CMD BYTE
0001 00490  SQRT  EQU      01H    ;8231A SQUARE ROOT CMD BYTE
0004 00500  TAN    EQU      04H    ;8231A TANGENT CMD BYTE
0018 00510  POPF   EQU      18H    ;8231A STACK POP COMMAND
001A 00520  PUPI   EQU      1AH    ;8231A PUSH PI ON STACK COMMAND
003A 00530  PRCHAR EQU     033AH   ;TRS-80 'PRINT CHAR.' ROUTINE
00540 ;
0002 00550  PARAM1 DEFS    2      ;FIRST MATH PARAMETER ADDRESS
0002 00560  PARAM2 DEFS    2      ;SECOND MATH PARAMETER ADDRESS
0002 00570  PARAM3 DEFS    2      ;THIRD PARAMETER ADDRESS
0002 00580  PARAM4 DEFS    2      ;FOURTH PARAMETER ADDRESS
0002 00590  PARAM5 DEFS    2      ;FIFTH PARAMETER ADDRESS
0002 00600  PARAM6 DEFS    2      ;SIXTH PARAMETER ADDRESS
0002 00610  PARAM7 DEFS    2      ;SEVENTH PARAMETER ADDRESS
0002 00620  DEST  DEFS    2      ;DESTINATION VARIABLE ADDRESS
00630 ;
F810 F3          MATH  DI          ;INTERRUPTS ARE DISABLED
F811 DD2A06F8    00650  LD          IX,(PARAM4) ;POINT TO D VARIABLE
F815 CDCFF8     00660  CALL     CONVRT ;PUT ONTO 8231A STACK
F818 DD21B4F9    00670  LD          IX,CONST7 ;POINT TO THE CONSTNT 7
F81C CDCFF8     00680  CALL     CONVRT ;PUT ONTO 8231A STACK
F81F CD3EF9     00690  CALL     POWER  ;RAISE TO THE POWER
F822 CD42F9     00700  CALL     SINE   ;GET SINE OF RESULT
F825 DD2A08F8    00710  LD          IX,(PARAM5) ;POINT TO E VARIABLE
F829 CDCFF8     00720  CALL     CONVRT ;PUT ONTO 8231A STACK
F82C CD16F9     00730  CALL     INVCOS ;GET INVERSE COSINE
F82F CD0EF9     00740  CALL     ADD    ;ADD INTERMED. RESULTS
F832 DD2A0AF8    00750  LD          IX,(PARAM6) ;POINT TO F VARIABLE
F836 CDCFF8     00760  CALL     CONVRT ;PUT ONTO 8231A STACK
F839 CD2AF9     00770  CALL     EXPON  ;GET NATURAL ANTILOG
F83C CD0EF9     00780  CALL     ADD    ;ADD TO INTERMED. RESULT
F83F CD4EF9     00790  CALL     PI     ;PUSH PI ONTO 8231A STACK
F842 CD12F9     00800  CALL     MULT   ;MULTIPLY W/INTERMED. RES
F845 DD21ACF9    00810  LD          IX,CONST2 ;POINT TO CONSTNT 2
F849 CDCFF8     00820  CALL     CONVRT ;PUT ONTO 8231A STACK
F84C CD12F9     00830  CALL     MULT   ;MULTIPLY W/INTERMED. RES
F84F DD2A0CF8    00840  LD          IX,(PARAM7) ;POINT TO G VARIABLE
F853 CDCFF8     00850  CALL     CONVRT ;PUT ONTO 8231A STACK
F856 CD3AF9     00860  CALL     NATLOG ;GET NATURAL LOG
F859 CD46F9     00870  CALL     ROOT   ;GET SQUARE ROOT
F85C CD0EF9     00880  CALL     ADD    ;ADD TO INTERMED. RESULT
F85F DD2A02F8    00890  LD          IX,(PARAM2) ;POINT TO B VARIABLE
F863 CDCFF8     00900  CALL     CONVRT ;PUT ONTO 8231A STACK
F866 DD21B0F9    00910  LD          IX,CONST4 ;POINT TO CONSTNT 4
F86A CDCFF8     00920  CALL     CONVRT ;PUT ONTO 8231A STACK
F86D CD12F9     00930  CALL     MULT   ;MULTIPLY B VARIABLE
F870 CD4AF9     00940  CALL     TANGNT ;GET TANGENT
F873 DD2A04F8    00950  LD          IX,(PARAM3) ;POINT TO C VARIABLE
F877 CDCFF8     00960  CALL     CONVRT ;PUT ONTO 8231A STACK
F87A CD12F9     00970  CALL     MULT   ;MULTIPLY WITH INTER. RES
F87D CD0EF9     00980  CALL     ADD    ;ADD INTERMED. RESULTS
F880 DD2A00F8    00990  LD          IX,(PARAM1) ;POINT TO A PARAMETER
F884 CDCFF8     01000  CALL     CONVRT ;PUT ONTO 8231A STACK
F887 DD21B0F9    01010  LD          IX,CONST4 ;POINT TO CONSTNT 4
F88B CDCFF8     01020  CALL     CONVRT ;PUT ONTO 8231A STACK
F88E CD3EF9     01030  CALL     POWER  ;DO POWER FUNCTION
F891 CD0EF9     01040  CALL     ADD    ;ADD TO INTERMEDIATE RES.

```

Listing 4 continued

SuperSCRIPIT PRINTER DRIVERS

 With an ALPS Printer Driver (software), you can attach your printer to SuperSCRIPIT Word Processing for automatic interfacing. Over 70 printers now supported. Call or write for latest Product Review Sheet describing features supported.
\$49 each w/o Proportional, \$59 with Proportional

MEMDISK FOR MODEL 4 IN MODEL III MODE

 Easy to use. (LDOS only). Use MEMDISK as your System Drive or load other files into memory for Instant Access. Use 80K of 128K memory. Too many features to describe here. Call or write for more details. **\$39**

4 MODEL 4 UTILITIES (FOR MODEL III MODE) \$19

 Run Model 4 at High CPU Speed (4 MHz)
 Test Memory (To 128K Automatic & Clear Memory Easily Sound (Beep) in JCL file or BASIC

MEMORY DUMP UTILITY (80x24 Screen) \$24
LOADMAP UTILITY \$19

Mail / Phone Orders Accepted
ALPS
 23 Angus Road
 Warren, New Jersey 07060
201-647-7230
 374

FREE
business software directory

- Radio Shack's Model 1, 2, 3 & 16
- CPM: Xerox, Alto...
- IBM PC & compatibles

Data base manager, integrated accounting package, inventory, word processing, and advanced mailing list.

149

Micro Architect Inc.
 6 Great Pine Ave.
 Burlington, MA 01803
 671-273-5658

PROJECT 80

Listing 4 continued

```

F894 DD2A0EF8 01050 LD IX,(DEST) ;POINT TO DEST. (Z) VAR.
F898 DB01 01060 IN A,(STATUS) ;GET LAST STATUS BYTE
F89A CB6F 01070 BIT 5,A ;CHECK FOR ZERO RESULT
F89C 280F 01080 JR Z,RESULT ;IF NOT ZERO, SKIP
F89E DD360300 01090 LD (IX+3),0 ;ELSE STORE 00 EXP BYTE
F8A2 3E18 01100 LD A,POPF ;GET 8231A POP COMMAND
F8A4 D301 01110 OUT (COMAND),A ;SEND TO 8231A
F8A6 DB01 01120 LOOP2 IN A,(STATUS) ;CHECK STATUS BYTE
F8A8 CB7F 01130 BIT 7,A ;DONE YET?
F8AA 20FA 01140 JR NZ,LOOP2 ;IF NOT, LOOP AGAIN
F8AC C9 01150 RET ;ELSE, DONE - RETURN
F8AD DB00 01160 RESULT IN A,(DATA) ;GET EXPONENT BYTE
F8AF 4F 01170 LD C,A ;SAVE VALUE TEMP.
F8B0 CB7F 01180 BIT 7,A ;CHECK MANT. SIGN
F8B2 DB00 01190 IN A,(DATA) ;GET HIGH MANT. BYTE
F8B4 2002 01200 JR NZ,SKIP ;IF BIT SET, OK
F8B6 CBBF 01210 RES 7,A ;ELSE, CHANGE TO NEG.
F8B8 DD7702 01220 SKIP LD (IX+2),A ;SAVE HIGH MANT. BYTE
F8BB 79 01230 LD A,C ;GET EXPONENT BYTE
F8BC 07 01240 RLCA ;ROTATE LEFT ONE BIT
F8BD CB2F 01250 SRA A ;DUPLICATE HIGH BIT
F8BF C680 01260 ADD A,128 ;MAKE EXCESS 128 FORM
F8C1 DD7703 01270 LD (IX+3),A ;SAVE EXPONENT BYTE
F8C4 DB00 01280 IN A,(DATA) ;GET NEXT RESULT BYTE
F8C6 DD7701 01290 LD (IX+1),A ;SAVE IN RESULT AREA
F8C9 DB00 01300 IN A,(DATA) ;GET NEXT RESULT BYTE
F8CB DD7700 01310 LD (IX),A ;SAVE IN RESULT AREA
F8CE C9 01320 RET ;DONE - RETURN
01330 ;
01340 ;
01350 ;*** SUBROUTINE CONVRT:
01360 ;* CONVRT gets the Radio Shack floating point value
01370 ;* pointed at by the IX register and converts it to
01380 ;* the Intel 8231A format, and pushes it onto the 8231A
01390 ;* stack.
F8CF DD7E03 01400 CONVRT LD A,(IX+3) ;GET EXPONENT BYTE
F8D2 B7 01410 OR A ;CHECK FOR ZERO VALUE
F8D3 2009 01420 JR NZ,CONV2 ;IF NOT, GO ON
F8D5 D300 01430 OUT (DATA),A ;ELSE, PUSH 00 BYTE
F8D7 D300 01440 OUT (DATA),A ;PUSH 00 BYTE
F8D9 D300 01450 OUT (DATA),A ;PUSH 00 BYTE
F8DB D300 01460 OUT (DATA),A ;PUSH 00 BYTE
F8DD C9 01470 RET ;DONE - RETURN
F8DE DD7E00 01480 CONV2 LD A,(IX) ;GET LOW ORDER BYTE
F8E1 D300 01490 OUT (DATA),A ;PUT INTO 8231A
F8E3 DD7E01 01500 LD A,(IX+1) ;GET NEXT ORDER BYTE
F8E6 D300 01510 OUT (DATA),A ;PUT INTO 8231A
F8E8 DD7E02 01520 LD A,(IX+2) ;GET HIGH MANTISSA BYTE
F8EB F680 01530 OR 80H ;SET HIGH BIT
F8ED D300 01540 OUT (DATA),A ;PUT INTO 8231A
F8EF DD7E03 01550 LD A,(IX+3) ;GET EXPONENT BYTE
F8F2 D680 01560 SUB 128 ;PUT INTO REAL FORMAT
F8F4 4F 01570 LD C,A ;SAVE EXPONENT IN C REG
F8F5 07 01580 RLCA ;MOVE HIGH BIT INTO CARRY
F8F6 0600 01590 LD B,00H ;CLEAR THE B REGISTER
F8F8 CB18 01600 RR B ;PUT CARRY BIT INTO B REG
F8FA E680 01610 AND 80H ;MASK LOW 7 ACC. BITS
F8FC A8 01620 XOR B ;CHECK FOR SAME BIT 7'S
F8FD C25EF9 01630 JP NZ,ERROR2 ;IF NOT, OUT OF RANGE ERR
F900 79 01640 LD A,C ;ELSE, GET EXPONENT BACK
F901 E67F 01650 AND 7FH ;CLEAR HIGH BIT
F903 DDCB027E 01660 BIT 7,(IX+2) ;CHECK MANTISSA SIGN BIT
F907 2802 01670 JR Z,NOCHG ;IF POSITIVE, NO CHANGE
F909 CBBF 01680 SET 7,A ;ELSE SET BIT FOR NEGATIV
F90B D300 01690 NOCHG OUT (DATA),A ;PUT EXP. BYTE INTO 8231A
F90D C9 01700 RET ;DONE - RETURN
01710 ;
F90E 3E10 01720 ADD LD A,FADD ;GET FLOATING ADD CMD BYT
F910 183E 01730 EXECUT JR ;TWO-VALUED OPERATION
F912 3E12 01740 MULT LD A,FMUL ;GET MULTIPLY COMMAND BYT
F914 183A 01750 EXECUT JR ;TWO-VALUED OPERATION
F916 3E06 01760 INVCOS LD A,ACOS ;GET INV COSINE CMD BYT
F918 1836 01770 EXECUT JR ;ONE-VALUED OPERATION
F91A 3E05 01780 INVSIN LD A,ASIN ;GET INV SINE CMD BYT
F91C 1832 01790 EXECUT JR ;ONE-VALUED OPERATION
F91E 3E07 01800 INVTAN LD A,ATAN ;GET INV TANGENT CMD BYTE
F920 182E 01810 EXECUT JR ;ONE-VALUED OPERATION
F922 3E15 01820 CHGSGN LD A,CHSF ;GET CHANGE SIGN CMD BYTE
F924 182A 01830 EXECUT JR ;ONE-VALUED OPERATION
F926 3E03 01840 COSINE LD A,COS ;GET COSINE COMMAND BYTE
F928 1826 01850 EXECUT JR ;ONE-VALUED OPERATION
F92A 3E0A 01860 EXPON LD A,EXP ;GET EXPONENTIAL CMD BYTE
F92C 1822 01870 EXECUT JR ;ONE-VALUED OPERATION
F92E 3E13 01880 DIV LD A,FDIV ;GET DIVIDE CMD BYTE
F930 181E 01890 EXECUT JR ;TWO-VALUED OPERATION
F932 3E11 01900 SUB LD A,FSUB ;GET SUBTRACT CMD BYTE
F934 181A 01910 EXECUT JR ;TWO-VALUED OPERATION
F936 3E08 01920 COMLOG LD A,LOG ;GET COMM LOG CMD BYTE
F938 1816 01930 EXECUT JR ;ONE-VALUED OPERATION
F93A 3E09 01940 NATLOG LD A,LN ;GET NAT LOG CMD BYTE
F93C 1812 01950 EXECUT JR ;ONE-VALUED OPERATION
F93E 3E0B 01960 POWER LD A,PWR ;GET POWER COMMAND BYTE
F940 180E 01970 EXECUT JR ;TWO-VALUED OPERATION
F942 3E02 01980 SINE LD A,SIN ;GET SINE COMMAND BYTE
F944 180A 01990 EXECUT JR ;ONE-VALUED OPERATION
F946 3E01 02000 ROOT LD A,SQRT ;GET SQUARE ROOT CMD BYTE

```

Listing 4 continued

mat, pushes them onto the 8231A internal stack, gives the 8231A command to start the desired operation, then processes and stores the result. It also prints appropriate error messages. If any calculation errors occur, the program displays a general Calculation Error message. I did not add the code to display the particular problem (for example, divide by zero), but you can add this since the 8231A provides specific error flags.

The Basic program indicates its choice of function by the parameter passed with the USR function. The top of Listing 2 shows the various commands and operations available. The USR function in Listing 1 is passing the value 12, which indicates the power function (raise A to the B power).

You can run some Basic benchmarks with this method, simply by running the program in Listing 1 as is (with a loop to line 5000), and with line 70 replaced with the Basic math replacement (for example, C = A*B). I did this for several functions and found only small improvement for Basic functions (5 to 9 seconds for add, subtract, multiply, and divide), but a marked improvement for more complex functions (sine, cosine, tangent, powers, etc.). For example, the power function dropped time from 5:04 (all times are in minutes:seconds) to 1:02, the sin function went from 2:49 to 0:55, and the tan function went from 5:23 to 0:56!

The program in Listing 2, as mentioned, provides the necessary interface to the math processor. The CONVRT routine beginning at line 910 converts a TRS-80-formatted variable, pointed to by the IX register, into the proper 8231A format and stores the value in the 8231A processor. The Result routine, beginning at line 1890, gets the result from the 8231A and converts it to TRS-80 format and stores it in the proper location as specified by the DEST variable at 0F804 hex.

The problem with the above method is that there is too much Basic overhead. While times are great for a lot of calculations after the program sets up the variable pointers, there is too much time required to change the variable pointers each time you want a new operation with new variables. The Basic overhead outweighs the advan-

PROJECT 80

Listing 4 continued

```

F948 1806      02010      JR      EXECUT      ;ONE-VALUE OPERATION
F94A 3E04      02020      TANGNT  LD      A,TAN      ;GET TANGENT COMMAND BYTE
F94C 1802      02030      JR      EXECUT      ;ONE-VALUED OPERATION
F94E 3E1A      02040      PI      LD      A,PUPI      ;GET PUSH PI COMMAND
F950 D301      02050      EXECUT  OUT     (COMAND),A      ;SEND COMMAND TO 8231A
F952 DB01      02060      LOOP1  IN      A,(STATUS) ;CHECK FOR COMPLETION
F954 CB7F      02070      BIT      7,A      ;STILL BUSY?
F956 20FA      02080      JR      NZ,LOOP1    ;IF YES, CHECK AGAIN
F958 E61E      02090      AND     1EH      ;CHECK FOR ANY ERROR
F95A C264F9    02100      JP      NZ,ERROR3  ;IF YES, CALC. ERROR
F95D C9        02110      RET     ;DONE - RETURN
          02120      ;
          02130      *** ERROR ROUTINES:
F95E E1        02140      ERROR2 POP     HL      ;POP ADDR FROM STACK
F95F 2171F9    02150      LD      HL,ERMSG2 ;POINT TO ERR MSG #2
F962 1804      02160      JR      PRRTN      ;PRINT AND RETURN
F964 E1        02170      ERROR3 POP     HL      ;POP ADDR FROM STACK
F965 2194F9    02180      LD      HL,ERMSG3 ;POINT TO ERR MSG #3
F968 7E        02190      PRRTN  LD      A,(HL) ;GET CHARACTER
F969 B7        02200      OR      A      ;CHECK FOR ZERO BYTE
F96A C8        02210      RET     Z      ;IF ZERO, DONE
F96B CD3A03    02220      CALL   PRCHAR     ;ELSE, PRINT THE CHAR.
F96E 23        02230      INC     HL      ;POINT TO NEXT CHAR.
F96F 18F7      02240      JR      PRRTN      ;LOOP AGAIN
          02250      ;
          02260      ;
F971 0D        02270      ERMSG2 DEFB   0DH      ;
F972 2A        02280      DEFM   '*** PARAMETER VALUE OUT OF RANGE'
F992 0D        02290      DEFB   0DH      ;
F993 00        02300      DEFB   00H      ;
F994 0D        02310      ERMSG3 DEFB   0DH      ;
F995 2A        02320      DEFM   '*** CALCULATION ERROR'
F9AA 0D        02330      DEFB   0DH      ;
F9AB 00        02340      DEFB   00H      ;
          02350      ;
F9AC 00        02360      CONST2 DEFB   00H      ;CONSTANT VALUE 2 BYTES
F9AD 00        02370      DEFB   00H      ;
F9AE 00        02380      DEFB   00H      ;
F9AF 82        02390      DEFB   82H      ;
F9B0 00        02400      CONST4 DEFB   00H      ;CONSTANT VALUE 4 BYTES
F9B1 00        02410      DEFB   00H      ;
F9B2 00        02420      DEFB   00H      ;
    
```

Listing 4 continued

tage of using the hardware math processor in many applications.

A better solution is to execute an entire equation at a time by the math processor before returning from the USR call. This eliminates the overhead problem and speeds up execution significantly.

Program Listing 3 shows a short Basic program with a reasonably complex equation in line 70. I have set the program to execute the equation 1,000 times. My computer executed the program in 7:45. I then created a significantly modified version of the Assembly-language program in Listing 2 (see Program Listing 4). This program executes the same equation by obtaining the various Basic variables (in the same way as in Listing 2) and calling the various math processor functions. The USR routine calls this program from Basic (see Program Listing 5) and executes the desired equation. The program ignores the parameter passed in the USR function, although it could be used to select one of several possible equations. Notice that we are now

MASTER HORSE HANDICAPPER™



EVALUATES FROM RACING FORM!

Age	Gender	Post (Today)
Class	Jockey (Today)	Post (Last)
Condition	Jockey (Last)	Speed
Consistency	Length	Trainer
Earnings	Time of Year	Workouts

And gives you GRAPHIC REPRESENTATION of finish

ADAPTS TO ANY TRACK IN THE WORLD

Quickly and easily be changing data statements relating to local track records/jockeys and trainers. COMPLETE INSTRUCTIONS INCLUDED

A. Thoroughbred/Maiden

Full featured menu driven thoroughbred handicapping program with graphic finish. Contains special "maiden analysis" for first time starters . . .

MEM 16K \$59⁹⁵

B. Quarterhorse/Maiden

Complete quarterhorse analysis with "maiden" feature and graphic finish . . .

MEM 16K \$59⁹⁵

C. Thoroughbred/Pace

The same as Program A except contains a "pace from finish" analysis that is a must for "close" races, early season, and longshots . . .

MEM 32K \$79⁹⁵

D. Master Handicapper™

"ALL" of the programs listed above in ONE PROGRAM with interactive instructions . . .

MEM 48K \$99⁹⁵

*All programs above come complete with instruction manual

PROGRAMS AVAILABLE FOR	WE CARRY
TRS-80 I, III, 4	MANY
Apple™ II, II+, IIE	ITEMS
CPM 2.2, 3.0	NOT
SOON: MS-DOS, MOD 100	LISTED

SPRING RACING SPECIAL

COMPLETE PROFESSIONAL SYSTEM:

- Deluxe Master Handicapper™ or Master Dog Analysis™
- Master Better™
- Track Management™
- Manual and Instructions

Prof. Jones' Price **\$199⁹⁵** DISK ONLY

DLX. Deluxe Master Handicapper™

Contains Program D and also includes: 1) Print Mode 2) Escape Mode 3) 14 Horse Fields 4) Internal Instruction and Manual

MEM 48K \$129⁹⁵

MB. Master Better™

A complement to ALL Master Handicapper Programs, includes: 1) Win/Place/Show 2) Quinella 3) Exacta 4) Trifecta 5) Pick Six 6) Daily Double 7) Money Management 8) Odds Analysis 9) and Much More

MEM 16K \$59⁹⁵

DA. Master Dog Analysis™

The only professional dog handicapper on the market, includes: 1) Speed 2) Post Today 3) Kennel 4) Post Lane 5) Distance 6) Condition 7) Running Style 8) Weight 9) plus much more. If you are near a greyhound track, you can't afford not to use this program.

MEM 32K \$99⁹⁵

Professor Jones' Football Predictor, Prof. Pix

This complete football analysis will predict: 1) Overlays 2) Point Spreads 3) "Superplays" 4) "Over/Under" Bets. For NFL/USL/College. Specify Mod I/III

T - \$19⁹⁵ D - \$24⁹⁵

Track Management™

A revolutionary data base program designed to keep records on "ALL" horses or dogs running at a track. Can also be used to expand Jockey/Trainer stats in all Master Handicapper™ programs. **A MUST FOR THE SERIOUS HANDICAPPER.**

(48K, Disk Only) \$69⁹⁵

PROFESSIONAL SYSTEMS

LNW II (96K)

With/
Green
Monitor
— 2 Tandon drives
— Cables, etc.
AND ALL SOFTWARE

retail \$4109

Prof. Jones' Price **\$2195** ready to plug in
LOWEST PRICE For Whole System

PROFESSIONAL BUSINESS SOFTWARE

	SALE PRICE
Micro Cash — retail bus.	49.95 129.95
Accounts receivable — inv.	49.95 129.95
M/Zal — editor/ assembler	149.95 129.95
Maxi CRAS — check reg.	99.95 79.95
Maxi Mail — mail system	99.95 79.95
Maxi Stat — powerful stats	199.95 149.95
DOS plus 3.5	149.95 129.95
New DOS 80	149.95 129.95
DOS plus 3.4 (Mod I)	49.95 99.95

Minimum 15% Discount on Most Other Software

Send check / money order / VISA / Mastercharge (Include expiration date) to:

48 HOUR SHIPPING
Call 208-342-6939
M-F 8-7 MST

Prof. Jones
1114 N. 24th St.
Boise, ID 83702



TERMS: FREE SHIPPING ALL SOFTWARE. Add \$6.00 hardware / C.O.D. Add \$6.00 / Add 3 weeks personal checks / Add 4.5% ID residents / Add \$6.00 outside U.S.A. / Prices subject to change.

PROJECT 80

passing eight variable addresses, instead of three.

When I executed the Basic program in Listing 5, calling the Assembly-language program in Listing 4 1,000 times, the processing took only 25 seconds—only 1/19 of the original software-based execution time. ■

Parts for this month's project are available from DIGI-KEY Corp., P.O. Box 677, Thief River Falls, MN 56701; Radio Shack, National Parts Division, 900 East Northside Drive, Fort Worth, TX 76102; Washtenaw Digital Systems, P.O. Box 2014, Ann Arbor, MI 48106. (Add \$2 per order shipping and handling for the math processor; Michigan residents, add 4 percent sales tax.)

For further information send your questions and a self-addressed stamped envelope to Roger C. Alford, c/o Washtenaw Digital Systems, P.O. Box 2014, Ann Arbor, MI 48106, or call him between 7 and 9 p.m. weeknights at 313-973-9763.

Listing 4 continued

```
F9B3 83      02430      DEF8      83H
F9B4 00      02440 CONST7 DEF8      00H      ;CONSTANT VALUE 7 BYTES
F9B5 00      02450      DEF8      00H
F9B6 3C      02460      DEF8      3CH
F9B7 83      02470      DEF8      83H
          02480 ;
F810        02490      END      MATH
```

```
20 DEFUSR0=&HF810
40 A=0:B=0:C=0:D=0:E=0:F=0:G=1:Z=0
50 V=VARPTR(A):GOSUB170
60 V=VARPTR(B):GOSUB180
70 V=VARPTR(C):GOSUB190
80 V=VARPTR(D):GOSUB200
90 V=VARPTR(E):GOSUB210
100 V=VARPTR(F):GOSUB220
110 V=VARPTR(G):GOSUB230
120 V=VARPTR(Z):GOSUB240
130 FORI=1TO1000
140 X=USR(0)
150 NEXTI
160 PRINTZ:END
170 V1=INT(V/256):POKE-2047,V1:POKE-2048,V-V1*256:RETURN
180 V1=INT(V/256):POKE-2045,V1:POKE-2046,V-V1*256:RETURN
190 V1=INT(V/256):POKE-2043,V1:POKE-2044,V-V1*256:RETURN
200 V1=INT(V/256):POKE-2041,V1:POKE-2042,V-V1*256:RETURN
210 V1=INT(V/256):POKE-2039,V1:POKE-2040,V-V1*256:RETURN
220 V1=INT(V/256):POKE-2037,V1:POKE-2038,V-V1*256:RETURN
230 V1=INT(V/256):POKE-2035,V1:POKE-2036,V-V1*256:RETURN
240 V1=INT(V/256):POKE-2033,V1:POKE-2034,V-V1*256:RETURN
```

Program Listing 5. USR routine to call program from Basic.

Circle 244 on Reader Service card.



Hello thayuh. This is Eben Flow, proprietor of the Fish or Cut Bait Company, buyer and seller of lobster bait for 49 years. My hobbies are collecting linoleum samples, squashing flies and playing pac-person on my home computer.

But here on Martinicus Rock, off the coast of Maine, the power can be a tad erratic. So, to cure the brownout and blackout problems, and to keep them spikes and surges off my picture tube, I got me a **MAYDAY** Uninterruptible Power Supply from SUN RESEARCH. Them fellas fixed me up real good and real light on my pocketbook, too. Got me a **MAYDAY** for my mini-calcaputer with a voltage regulator and everything for only 325 clams. They even included the battery in a nice waterproof box. Handy out here, you know. Now, if **MAYDAY** would only keep them sea dogs out of my barrel. . .

MAYDAY - Protection even you can afford!



SUN RESEARCH, INC.
Box 210
New Durham, NH 03855
603/859-7110
TWX 5102974444

✓ 285