

52-NOTES

Volume 3 Number 3

48/39/38

March 1978

Newsletter of the SR-52 Users Club
published at
9459 Taylorsville Road
Dayton, OH 45424

BOOK REVIEW: USER SURVIVAL GUIDE FOR TI-58/59 MASTER LIBRARY, by Fred Fish (606), \$10.25 postpaid to members, 119 pp, copyright 1978

Fred has put together a comprehensive package designed for the serious user of the ML programs, who wants to know how they work, how to interface them with his own programs, and how to take shortcuts. He brings to the user's attention conditional caveats, and pitfalls to avoid, and details specific program calling sequences to optimize register usage, execution speed, and required user code. This work appears to be technically sound, includes listings, flowcharts, and even a few cartoons, is well-proofed, and I am pleased to give it my endorsement. Include your Club membership number with your order, and ask Fred about quantity discounts.

A REVIEW OF THE TI-57 VIS-A-VIS ITS CONTEMPORARY PPCs

The 57 has a few unique features which fall into both + and - categories when compared with other PPCs. This review is intended as an aid to 57 users as they either start from scratch, or attempt to translate other PPC programs, and to acquaint non-57 users with this somewhat different machine.

It is interesting that the low-end TI PPC's unique features are common to some of the high-end HP's architecture (67 or 97): 1) The numerals 0-9 are used as labels, 2) All composite instructions are merged 3) The step following a conditional can be any instruction, and 4) During execution, the display flickers intelligibly (which the 67/97 can be made to do under certain circumstances). Perhaps the most critical limitation is the comparatively large amount of register sharing. While the number of data registers (8) puts the 57 into the same league as the 56 or HP-25, shared use of some of them is excessive: 1) The last 2 (of a max of 4) pending operations go into Reg 5 and 6, 2) Reg 7 is also the T register, and 3) The Σ^+ uses all the data registers except Reg 6. However, the latter results from a positive feature: The Σ^+ function processes 2 variables at a time. Another important limitation is the 8-place display (with 3 guard digits). While this may be acceptable for most applications, the user should bear in mind that actual accuracy may in some cases be lower than expected (1 INV lnx produces e to only 9 places). Execution speed of the 57 is generally slower than that of the other TI PPCs, especially for the trig and exponential functions. A realistic comparison of the 49 merged steps with the 56's 100 unmerged ones will be deferred until a few representative challenging test cases

The SR-52 Users Club is a non-profit loosely organised group of TI PPC owners/users who wish to get more out of their machines by exchanging ideas. Activity centers on a monthly newsletter, 52-NOTES edited and published by Richard C Vanderburgh in Dayton, Ohio. The SR-52 Users Club is neither sponsored nor officially sanctioned by Texas Instruments, Inc. Membership is open to any person: \$6.00 includes six future issues of 52-NOTES; back issues start June 1976 @ \$1.00 each.

surface. However, as a starter, when I attempted translation of Dix Fulton's V2N10pl program, it overflowed by about 20 steps.

A few programming tips come to mind: 1) Instead of inputting via a S1 R/S S2 R/S ... programmed sequence, write S1 S2 ... and execute with the SST key; 2) Take advantage of the anything-goes step following a conditional, where such functions as R/S, rtn, Pause, \pm , etc may be better than GTO n; and 3) If you run out of user-defined labels, SBR 2nd nn from the keyboard (but not under program control) will start execution at step nn. 57 users are invited to comment and share their tips, inventions, and discoveries.

FRACTURED DIGITS: MORE CONTROL (58/59)

Roger Gentry (398) has discovered some new ways to produce fractured digits, one of which leads to a capability to display a few selected characters with or without specified numerals. For example, with a 59 at turn-on or a 58 at turn-on and a 459.0 partition, key: GTO 441 LRN List 0 0 0 0 0 LRN Pgm 11 SBR 444 R/S LRN, and see 2 ° symbols looking like a pair of eyes. The machine at this point appears to be in LRN mode, expecting a 2-digit register address. Pressing any non-numeral except LRN writes code in the usual way; pressing a numeral "closes" the eyes, producing 2 - symbols in addition to writing a number code. Press LRN LRN BST and see the keycode for the numeral you just keyed, or key another numeral then BST, and see the merged code for both numerals. Other (but not all) permutations of fractured digits in 3 consecutive display positions can be produced similarly, following rules based on this discovery of Roger's: The current step number in the user-memory IAR which prevails when the sequence: Pgm 12 SBR 999 R/S LRN is keyed manually, determines what characters are formed at display positions 8, 9, and 10 (using the 0-11 full display convention). Roger observed that when a step is converted into 3 such codes by dividing it by 8, interpretation of the quotient, and remainder in integer form as V1N2p5 character code produces the expected characters. In the above example, the call to Pgm 11 is made at user memory step 447. Division by 8 produces 55 remainder 7 which are the codes for ° ° blank. But there are other requirements to be met to assure the desired display. If we label the octet containing the "launch" step (in this example step 447): AB, CD, ... OP (as in V1N1p5) the following rules appear to be necessary to follow to produce just the 2 eyes: A must be zero, B can be anything, C must be non-zero, and OP can be anything. Note that MN and OP are set to 72 and 42 respectively, following execution. Non-zero EF, GH, and IJ can produce up to 6 numerals preceding the eyes; non-zero A and C show up to the right of the open and closed eyes, respectively; fix n, n: 1-8 positions a decimal point in accordance with n; launch at other than the last step in the octet produces different results ... there seems to be no end to the variations. Members are invited to explore further, and to report useful (or what may appear to be potentially useful) creations.

LABEL SEARCH METHODS: A COMPARISON OF TI/HP MACHINES

John Hirsch (736) notes that while considerable attention has been given to the pros and cons of AOS vs RPN, label addressing comparisons have received little attention. John points out that the TI machines have absolute and label addressing, contrasting with the newer HP machines' relative and label addressing.

Perhaps a definition of terms at this point would help to avoid confusion. In past issues of 52-NOTES I have used the term relative to describe addressing within relocatable code, and absolute for unrelocatable code. John uses the term relative with the HP machines to mean an addressing method whereby an address is calculated by adding/subtracting an integer value to/from the address of the currently executed instruction. In other words, the program counter (or what I prefer to call the IAR (V2N4p5)) is either incremented or decremented by a specified amount. It is in this same way that the word relative is used to describe general purpose computer addressing, and since it is quite possible that a new TI PPC (or personal microcomputer) might have such addressing, henceforth I will use the term relative addressing with this meaning, and as John suggests: label addressing for what I used to call relative addressing.

HP-67/97 relative addressing is backwards only (a negative value stored in the I register is added to the contents of the IAR), so for forward transfers, the user must count the number of steps back through 000, 224, 223, ... to the desired step, and then provide for storage of the negative of this count in the I register. The primary advantage of this scheme is that it shares the same speed enhancement and memory-region independence of the TI machines' absolute addressing without the unrelatability restriction. Another important difference is the label search mechanism: The HP machines start a search downward from the instruction invoking the search. This makes it difficult to optimally locate subroutines called throughout a main program, but makes it possible to use the same label more than once, and expedites transfers within code located low in memory.

John made some label-search-execution time comparisons among the SR-52, SR-56, TI-57, TI-59, and HP-29c machines, and reached the following conclusions:

1) There is really not much difference in label search time when differences in total program steps are considered, although the TI-57 came out best.

2) The HP disadvantages offset the advantages.

and suggests for 3rd generation machines a key which gives the user the option of specifying label search to be forward, backward, or from top to bottom.

Other members are invited to comment, and contribute further to this topic.

ROUTINES

Fibonacci Number Generator Revisited (V3N1p3): Upon applying his routine to large numbers, Joel found that correct answers could be extended a bit by calculating the $(1 + \sqrt{5}) \div 2$ constant (known to mathematicians as Tau or Phi, or popularly as the Golden Ratio) as $(1 + (1 \div \sqrt{.2})) \div 2$. I will refer to $(1 + \sqrt{5}) \div 2$ as t1, and $(1 + (1 \div \sqrt{.2})) \div 2$ as t2, and Tau in general as t. Incidentally, some interesting things about t are due to the relationship: $t^2 = t + 1$, which may also be written: $t = (t - 1)^{-1}$. It turns out that t2 is a better approximation than t1 by one place, because of machine rounding and truncation. As calculated, t1=1.618033988749, t2=1.618033988750, but since t=1.61803398874989... t1 is good to only 12 places, while t2 is good to 13.

Efficient I/O Handling For Engineering Problems (58/59): Clyde Durbin (618) found that the nature of some of his engineering problems suggested some useful tricks/shortcuts, which should also be helpful in related applications. One situation which arises often concerns the user's desire to specify printing and tagging of a few outputs out of many possible ones, following initial processing. The sequence: LA S0 R*0 H8 GT0*0 when called by nA, where n is a register address, will cause code to be executed and tagged results printed, both specified by the contents of Reg n, provided Reg n is initialized with: sss.00tttttttt where sss is the step where specified processing is to begin, and tttttttt the print code for the desired tag. The step sss sequence ends with ... Op6 R/S. Clyde has also found that in cases where 2 or more input parameters are in mutually exclusive numerical ranges, they can all be entered with a single user defined key. The program first compares an input with prestored boundary values, and branches accordingly.

Copying The Display Into the T Register (56,57,58,59): There are times when it is useful to effectively store the display in the T register without changing the display. Jared Weinberger (221) suggests: ... (CP + xXt)... . Does anyone have a shorter way? (...CP + xXt = ... doesn't count!, and for the 57, ST0 7 does the trick).

Digit Reversing (V2N11p6): Reinhold Patzer (689) has trimmed Jared's by 3 steps and one register with: LA (CE ÷ 10 Prd 1 - INV Int SUM1) INV x=t A (Excl X 10) rtn, and notes that both routines will handle up to 13-digit integers.

TIPS AND MISCELLANY

A New Hidden Facet of the Old Machines (52,56): Ken Whipple (849) discovered that his SR-52 dimly displays a numeral representing the number of digits to the right of a displayed decimal point in the otherwise unused decapower LSD position for a real in fix format. Ten digits to the right (the maximum) even shows up as an A (as if the machine knew it should count in the hexadecimal number base)! I confirm Ken's findings, and add that the SR-56 counts the same as follows: ' 0 - blank 5 6 " blank ' A. So here's one way to get fractured digits on the 56. The new machines do not seem to respond in this way, but perhaps there are related frontiers waiting to be explored via darkened room and magnifying glass.

Extra Memory for the TI-58: Bob Edelen (100) reports that he has successfully added 2 TMC 0598 chips (order from TI per V2N5p3 for the TMC 0599, omitting Riggs' name) to his 58, effectively converting it into a 59 without mag card read/write. Bob has a supply of these chips, and is prepared to install them for 58 owners. Write Bob at his business address: 5440 Roslyn #285 Denver, CO 80216.

Data Entry Sensing (V2N7p5): Reinhold Patzer (689) brought to my attention the best approach I've seen yet for program-sensing of data entered from the keyboard. The method is due to Heinrich Schnepf (376) which he describes (in German) in V2N2p35 of his DISPLAY newsletter, and is simply to follow a programmed data-entry R/S with a decimal point and a zero test. It appears to work for all reals so long as the display is zero or hard at the R/S, and is left soft following a datum entry; a keyed zero is treated as no entry. This trick works because the decimal point zeros a hard display, but does not alter a soft one.

A New R-P Convention (57,58,59): J R Merrill (693) notes that the new machines return positive angles in the 0-270° range, and negative to -90°, unlike the older TI and all the HP machines, which return $\pm 180^\circ$, for rectangular to polar conversions. Which is better? J R prefers the latter, but there may be applications favoring the former. Page V-31 of the 58/59 owner's manual notes the 4th quadrant negative angle convention, but doesn't say why it was chosen. The 57 manual doesn't appear to address the matter of returned-angle signs at all.

Fast Constants: When deciding whether to create or store a constant, if execution speed is critical, note that creating a digit string with more than 3 elements using program code takes longer to execute than a RCL.

Runaway Mag Card Drive Motor (59): John Hirsch (736) reports occasions when following card drive, the motor can only be stopped by turning the machine off. The owner's manual (p VII-8) states that keying R/S should stop a running motor, but implies that it may be "normal" to have to turn the machine off in some cases. Anyone else experiencing John's problem, let me know.

More on Pgm mm R/S (V3N1p3): John has found that there are indeed applications for sequences of the form: Pgm mm R/S in user programs. After a CROM routine in Pgm mm has been called by a user program, the effect of Pgm mm R/S is to resume CROM execution at the step following the last rtn encountered, just as a plain R/S does during keyboard interaction with a CROM program. A CROM pointer maintains this restart address through intermediate user-code execution, but not following a manual or programmed call to another specified Pgm label or step number. For example, write: LA 3 Pgm 2 A 1 Pgm 2 B R/S LB Pgm 2 R/S R/S GT0 B. Press A, see 1; key a datum, press R/S, repeat for say 3 data. Now go elsewhere in user memory, write a small routine (that doesn't clobber the ML-02 registers, but may use all 6 subroutine levels), run it, then key the next datum for input to ML-2, and press B; key a few more data with R/Ss and find that all data were stored sequentially as expected in Reg 8, 9,

Creating Large Mantissas (57,58,59): For all 3 machines, creating a number in scientific notation whose mantissa is larger than 8 places can be done, but requires a few steps. For the 57, one way is to store the first 8 digits in a convenient register, SUM up to 3 more (appropriately scaled), then Prd 1 EE dd, where dd is the desired decapower. For the 58/59, the SUM step can be skipped for mantissas less than 11, or included for mantissas of 11 to 13 places.

Overflow and Underflow (57,58,59): The V2N1p3,4 discussion applies to the new machines, for the most part, but with a few exceptions. For all 3, display rounding of large numbers to 8 mantissa places corresponds to 10 places for the 52 and 56. When the 57 transfers any conditionally overflowed number from a data register to the display, it is converted to the upper-limit overflow number: 9.999999999D99. So operations performed on a conditionally overflowed number should be via register arithmetic.

Membership Address Changes: 128: Back to original address; 452: 3812 Kendale Dr Gautier, MS 39553; 553: 11 Plumeria St APRA Hts FPO SF 96630; 664: 100 Barton St #10 London, Ont N6A 1N2 CANADA.

Absolute Code Execution Speed (52,56,58,59): Roger Gentry (398) notes that absolute transfers near the top of memory are faster than those near the bottom (high addresses). The effect is the same as for label transfers, but is not so pronounced. Short Dsz loops placed within a 1-register octet of steps run a bit faster than when straddling 2 or more registers.

HIR Operator Modifications (58/59): Roger discovered that under certain circumstances HIR operations in the middle of pending arithmetic can change the originally designated arithmetic operators. For example, $n - H11\ H1 =$ produces $2n$, and $n \div H11\ H1 =$ produces n^2 . This behavior suggests that numbers pushed into the HIRs via pending arithmetic may not be so "clean" as once thought (V2N9p1).

More on CROM HIRs (V3N2p6): The Leisure Library program which T S Cox meant to refer to is Craps (13), which contains a "real" HIR at step 240. I say real, not "intended", because my definition for the latter is poor in this example: The only way this HIR operation could do anything is via a SBR 240 call, which would add the display value to the contents of HIR 2; SBR CLR R/S would add zero to HIR 2, and I don't see any other way for the program to even get to step 240. Also, there doesn't appear to be any use made of the contents of HIR 2. I make this assessment based on a listing kindly provided by T S (I'm still waiting for my free copy of the Leisure Library).

Mag Card Tips: Jerry Johnston (493) suggests folding a strip of paper around a mag card before inserting it into one of the TI wallets, to facilitate extraction, and recommends Pelikan #17 black ink for permanent card marking.

More on the Hardware Interrupt (V3N2p5,6): Roy Chardon (515) adds an improvement to make the interrupt handler module-independent: Make the CROM call to Pgm 1 SBR 021, since every CROM appears to have a rtn at that step. Roy also warns that the manual RST does, ofcourse, reset all flags and clear the subroutine-return register. The R/S at step 000 of the V3N2p6 sequence serves no purpose, and should be deleted or replaced by GTO N or GTO nnn, where label N or step nnn head an interrupt processing sequence, if such is not to start at step 000.

CROM Pause: The statement concerning step 353 of Pgm 19 in V2N8p6 is incorrect. That step is a rtn, not a pause. Code 66 in CROM, either intended or unintended executes as pause.

DC Adaptor: Mack Maloney (246) notes that TI's 12 volt DC to 120 volts AC converter (model DC 9105) was designed for use with the 52,58, and 59 (but would presumably work with other PPCs), and is available at the Washington DC TI Exchange Center.

Quick Manual Flag Tester (58/59): Nolan Tobias (853) notes that the V1N4p6 method for the 52 can be shortened to Ifflgn N, where N is an undefined label.

The R/S' Function (58): J R Merrill (693) notes that the 58's equivalent of the 59's Write function appears to execute as Int, and Jared Weinberger finds that when used as a label, R/S' behaves like R/S (V2N11p4). Roger Gentry (398) has found that a prevailing fix n condition, n: 1-8, followed by R/S' multiplies the integer part of the display by 10^n .

CLR/CLR' Differences (58/59): Joel Pitcairn (514) wonders if anyone has found an application where CLR and CLR' execute differently.