52-NOTES

Volume 2 Number 6 48/39 June 1977

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

THE NEW TI PPCs

The speculation is over (for awhile) following TI's announcement the last week in May of the New TI Programmable 57, 58, and 59 calculators. The 57 looks like even tougher HP-25 competition than the SR-56, with 10 labels and an $80 list price; the 58 and 59 are something else, packing some fascinating new features that it is hard to believe can be squeezed into a slightly smaller package than the 52, and at list prices of $125 and $300 respectively. Both use the same instruction set, accept 5000-step plug-in read-only-memory (ROM) program modules, and plug into the PC-100A printer; the 58 has half the 59's memory, and no card reader. The discussion that follows applies to the 59, but is generally applicable to the 58 except for the differences just noted.

120 general purpose memory registers are partitionable between data and program storage use: 1 data register is equivalent to 8 program steps from a 0/960 data/program mix to 100/160 in partitionable increments of 10 registers (11 possible configurations). Once a partition has been established, boundaries cannot be crossed (as with the 52). Partitioning/repartitioning can be done from the keyboard or under program control, except when a protected (in the proprietary sense) card has been read, in which case the user cannot get into LRN mode, single step, or recall as data, or copy on to another card the program memory contents.

The 5000-step ROM modules (which TI calls CROM for Constant ROM) are accessed by program number, and run by subroutine name for execution through the keyboard or under user-program control. The Master Library (which comes with each machine) holds 25 programs from a 52-step D.MS add/subtract/scaling routine to an 898-step matrix determinant/inverse/simultaneous equations leviathan that will handle up to a 9 X 9 matrix, or a system of 8 equations. Once a CROM program has been accessed from the keyboard, subsequent pressing of user-defined keys (A-E') refer to the CROM program subroutines, but a CROM program must be accessed each time a subroutine is to be called under user-program control. CROM programs may be down-loaded into user-program memory for inspection and/or editing, and run as user programs, but not the reverse. Configuring new CROMS is an expensive manufacturing process, and is not a practical user capability.

One of the most powerful of the new features is a set of 39 Control Operations that enable the user to control the printer in a variety of 64-character alphanumeric formats and list programs

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 interested person: $6.00 includes six future issues of 52-NOTES; back issues start June 1976 @ $1.00 each.

(including mnemonics), data, and labels; calculate a few special statistical functions; control and announce partitioning; configure error-state processing; and control increment/decrement of the first ten data registers. Control operations are executable manually and under program (user or CROM) control.

The 59's mag cards are slightly smaller than the 52's, and so cannot be used interchangeably. Each card reads/writes 2 of four memory banks, irrespective of partitioning. Each bank holds 30 data registers or 240 program steps, or equivalent mixes of each. Data registers number from the last of bank 4 up through the last third of bank 1 (2-digit addressing does not reach past Reg 99); program steps start at the beginning of bank 1, with step 959 ending bank 4. As examples: program steps 160-167 are equivalent to data register 99; steps 472-479/Reg 60; and 952-959/Reg 00. Partitioning is specified by data registers. For example, the sequence: 2 *Op 17 produces a 20/800 data/program-step mix. But since program protection is by bank, a bank that contains any data registers cannot be protected, and steps 720-791 in this example could not be protected.

The 59 keyboard is the 52's 5 X 9 matrix with some rearrangement, renaming, and the following changes: x!, D/R, iferr, ifpos, ifzro, and xrty have been eliminated, and CP, Pgm, xEt, Eng, Int, abs, Pause, x=t, Nop, Op, xGEt, sigma +, x bar, Rad, and Grad added. Most key codes follow the usual quasi-x,y addressing, with a few exceptions to handle merged Ind functions. 8 of the possible 100 op-codes are unused, and are the 59's pseudos. Cursory investigation reveals that of the 8 (op-codes: 21, 26, 31, 41, 46, 51, 56, and 82), p82 is the most interesting: no matter what is displayed, it always causes the next step to be skipped; if zero (or in some cases a small number) is displayed, when the skipped step is R/S, Nop, or INVSBR (the 59's rtn), execution of the p82 sets an error condition. P21, p26, p31, and p41 (SST) appear to behave as they do for the 52. So it seems that there will be some challenge to exploring the 59's pseudos, if not so much as for the 52, Incidently, the 59's pseudos are easier to create because of a register-address merging feature: in LRN mode, any 2-digits following a keyed register command (i.e, STO, RCL, etc) are merged into a single step, and the STO, RCL, ... can be deleted later. (You can do the same thing with GTO). But when you're editing a register sequence you have to rewrite the command as well as the address. For example, if you wanted to change STO 73 to STO 74, starting at the step containing the 74 would produce the 2 steps: 07 04 instead of the desired 74, so instead, beginning at the STO step: STO 74 would give you the desired sequence.

Other non-52 or 56 features include: 6-level subroutine calls; 10 flags; indirect flag, Pgm, Op, fix addressing; Dsz of Reg 00-09; SBR n from the keyboard initiates execution (true for the 57 also); 13-digit display arithmetic; and a manually initiated R/S during program execution will not halt execution in the middle of connected sequences.

On the negative side, varying in degree of severity and subjective importance: Only 8 mantissa digits are displayed when in Sci or Eng format (like the HP-25), although 13 digits can be carried in the display register unless EE is executed, in which case rounding is to 8 places; there is no built-in factorial function, and although Pgm 16 (Combinations, permutations, factorials) will calculate a factorial, 3 subroutine calls are required, and Reg 01-04 are used; polar/rectangular functions and character-print buffers use 4 stack registers; there is no manual D/R switch or other equivalent to the 52's interrupt processing capability; there is apparently no user access to the stack registers, and thus no fractured digits; the same 52/56 trig and log function anomalies prevail (V1N2p3 and V2N2p1); the same code-transfer rules apply (V1N2p2); there is no INV or Ind viability through subroutine calls; and a timing comparison of a short Dsz loop showed that the 59 took twice as long as the 52 (varying display format appeared to have ne effect). This longer Dsz execution time is probably due to the addressing of a particular register (which neither the 52 nor 56 have to do).

The following features will probably be viewed by some users as plus and by others as minus: Program execution halts when an undefined label is encountered; mag cards cannot be protected from accidental over-write (although since read is automatic there should be little need for the black tab type of protection); there appears to be no special 0 divide error state (V1N1p2); and there are no "crashes" except for the p21 sin sequence (see the pseudo table elsewhere in this issue).

One of the powerful features of the 59/PC-100A combination is that they can be programmed to operate like an interactive computer terminal. The following program illustrates this capability. Press A, and you're off on a graded course in arithmetic!

TI-59 Program: Interactive Arithmetic Teacher   Ed

Program Listing:

```
000:  *Lbl B STO 0 *Lbl E RCL*Ind 0 *Op 1 *Op 20 RCL*Ind 0 *Op 2 *Op 20
018:  RCL*Ind 0 *Op 3 *Op 20 RCL*Ind 0 *Op 4 *Op 20 *Op 5 INVSBR
033:  *Lbl C *CP *B' X RCL 79 = *Int *x=t C STO 75 *Lbl 3' *B' X RCL 79
052:  = *Int *x=t 3' xEt RCL 75 INVSBR *Lbl D *Pause xEt *Pause INVSBR
066:  *Lbl A 1 B E E E E E *Adv E *Adv *Lbl 9' 10 STO 79 0 STO 77 STO 78
089:  *Lbl 1' 29 B 0 R/S *CP *x=t 2' C + D STO 76 = xEt 47 *Op 4
110:  *Lbl 6' CLR R/S STO 74 *x=t 4' *A' 313237 *Op 4 xEt *Op06 453241
136:  *Op 4 RCL 74 *Op 06 53 B *Adv GTO *1' *Lbl *2' E 0 R/S *CP *x=t
155:  *5' C + xEt = STO 75 - D STO 76 = xEt 20 *Op 4 GTO *6' *Lbl *5'
176:  E 0 R/S *CP *x=t *7' C X D STO 76 = xEt 50 *Op 4 GTO *6' *Lbl *7'
197:     E 0 R/S *CP *x=t *8' C X xEt = STO 75 div D STO 76 = xEt 72
217:  *Op 4 GTO *6' *Lbl *8' *Adv 57 B 3221 *Op 4 *RCL 77 *Op 6 RCL 78
239:  *Prt E E 0 R/S *CP INV *x=t *9' E *Adv *Adv *Adv CLR R/S *Lbl *4'
256:  1 SUM 77 *A' 3632 *Op 4 xEt *Op 06 45 B E 0 R/S *CP *x=t *1' 10
280:  *Prd 79 GTO *1' *Lbl *A' 1 SUM 78 RCL 75 *Op 06 64 *Op 4 RCL 76
299:  *Op 6 INVSBR *Lbl *B' *pi X RCL 73 = INV *Int STO 73 INVSBR
```

Pre-stored Data:

```
01:  2324572465 3000223224 3122003732 37173637 4532413500 1335243723
07:  3017372415 0 1314242724 3745401331 3643173500 3441173620 3724323136
14:  2002132 3500451736 5700010000 2132350031 140273232 2600213235
20:  16243620 3327134520 3313413617 1600323317 3513311636 2617450013
26:  3136431735 5737231731 35413140 4313313700 3732001316 1671000000
32:  0 4313313700 3732003641 1437351315 3771000000 4313313700 3732003041
39:  2737243327 4571000000 4313313700 3732001624 4224161771 0 4532410043
46:  1735170035 2422233773 43133137 2313351617 3500333532 1427173036
52:  7100000000 1424351620 1435132431 1716001641 3115177300 4532413500
58:  3615323517 24360000 0 3335321427 1730364043 1331370037 3200373545
65:  2132350013 14173737 1735003615 3235177100 2313421700 1300312415
71:  1700161345 4014451740 .1415926536 0 0 0 0 0 0
```

While you probably won't be able to try out this program for awhile (the first production 58/59 machines aren't expected to be on retailers' shelves until late June or early July), at least one retailer is selling the owner's manual now (Lectro-Media Ltd Box 1770 Philadelphia, PA 19105 (800) 523-2906, for $12.95) ... a 250 page 8½ X 11 combination programming guide and functional analysis which covers a lot of ground, and can get you started writing programs and deciphering mine (and/or improving it) while you're waiting to get a 58 or 59.

Although 52-NOTES will be giving the new machines increasing coverage, some space will continue to be devoted to significant topics concerning the old, depending upon member interest; important functional and operational comparisons of one machine with another will be of general interest for some time to come; then, ofcourse, the 58 and 59 will have to wait for HP to catch up before they can fairly enter the Friendly Competion arena!

- - - - - - - - - - - -

PSEUDO BEHAVIOR SUMMARY (52)

Herewith a quick reference for pseudos, which consolidates inputs from many members:

21: A, E, F; causes crash* when followed by sin, cos, tan, P/R, or D.MS during program execution
26: A, B, D, F;
31: D; causes a halt in LRN mode during program execution
61: A, B, D;
62: B, C, D;
63: D; executes as iferr
64: D; executes as INV
66: A, B, D;
71: A, E; SST of the first of a string of n p71s causes a skip of n+1 steps
72: B, C, D; cannot be inversed
73: D; executes as rset
74: D; executes as GTO
76: A, B, D;
82: B, C, D; cannot be inversed
83: D; causes complex interaction among display, arithmetic stack, internal registers, and functional states
84: D; similar to p83
92: B, C, D;

A=conditionally neutral; B=can provide missing operand; C=executes as EE and softens display selectively; D=may be used as a label; E=executes as a no-operation; F=when SST'd, creates a pending shift;

*Machine displays the two minus signs, and will not respond to any keyboard command except power off.

HARDWARE MODIFICATIONS

Bob Edelen (100), Michael Rak (502), and Tom Scogin (517) are emerging as the Club's top hardware modification enthusiasts. Both Bob and Mike are starting chip add-on services, and expect to do other machine mods for interested users. I will publish the names of other Club members wishing to announce services related to PPCs and their use, but in no case should such announcements be construed as carrying Club affiliation or endorsement. Incidently, members who wrote to Mike (V2N5p3) may wish to contact him again for more details; both he and Bob report special mounting requirements for SR-52As.

MORE ON CORNER THE LADY (WYTHOFF'S NIM) (52)

Michael Brown (128), Dix Fulton (83), and Larry Mayhew (145) have written SR-52 programs for this game (V2N4p3), each taking a different approach. Michael's handles the largest playing field (1D10 X 1D10) but follows a partially iterative algorithm that costs execution time. Dix's is limited to a 99 x 99 field and also requires some trial-and-error search. Larry devised a non-Fibonacci Notation closed-form algorithm, and his program runs the fastest, handling a playing field up to 99999 X 99999. All three programs are well written, and it would be difficult to decide which represents the best programming, but I would judge Larry's to be the most playable. Larry modestly suggests that his algorithm is probably not as mathematically elegant as the Fibonacci Notation one, but there is no denying the proof of his pudding! While 52-NOTES is not the appropriate forum to pursue mathematical games in great detail, I will devote some future space to this topic should member interest/effort be sufficient. In the meantime, interested members might wish to contact Mike, Dix or Larry for further details of their programs.

A FEW TIPS ON PRESENTING YOUR INPUTS TO 52-NOTES

It will be easier for me to use your inputs to 52-NOTES if 1) separate topics are on separate pieces of paper, 2) little or no text rewording is required, and 3) programs are typed following a modified Dix Fulton format (V1N3p4), i.e. mnemonic strings with row-wise catenation (filled lines), without omitting the * (for 2nd) symbol. Incidently, although several have suggested the / symbol to denote division, in certain contexts it might be misinterpreted as a delimiter, and I prefer to stick with div. Try to use mnemonics as close to keyboard inscriptions as can be typed, follow upper and lower case conventions, and run merged code together. Note that for the new machines (58 and 59), LBL is Lbl, and rtn is INVSBR. I suggest that we adopt the symbols: abs for absolute value, S+ for sigma plus, and use 0', 1', ... 9' for labels, rather than the actual shift functions (Dsz, ifflg, ... Op) they represent. I am now inclined to reverse an earlier position (V1N1p5) and suggest that programs submitted for 52-NOTES publication be in relocatable form to make them easier to modify for individual preference (the SR-56 excepted).

SR-52 PROGRAM REVIEW: ANALYTIC COMPUTER MODEL by Ron Zussman (88), Computer Design, May 1977 pp105-109; reprints @$1.50 ea or entire manuscript @$3.00 ea available from Ron. As big and fast as modern computer operating systems are, their limitations are finite, and their efficiency depends in large part on how well hardware resourses have been configured to match specific job loadings. Analytic models have been devised to determine optimum mixes and arrangements of computing machinery for predicted user traffic. Ron's program mechanizes one of these, based on queuing theory. From 7 inputs: 1) Time the central processing unit(s) is (are) active (CPU(s)) for each job, 2) proportion of (1) used for I/o, 3) average service time for periferal storage access, 4) total number of jobs, 5) number of system users, 6) Average wait times (for interaction), and 7) the number of CPUs, this program calculates CPU utilization, thruput, and response time for both exponential and constant models. Members applying this program to real operating systems are invited to convey comments/results to Ron and/or me. Users should note that the user instructions in the CD article incorrectly list results under a "Press" column (p107). Corresponding keys may be found at the bottom of p 109.

TIPS

A Y^x Workaround: In cases where you want y^x with x already displayed, Roy Grubb (483) suggests: X y lnx = INV lnx which takes advantage of the mathematical identity: log y^x ≡ xlog y.

A Decrement Only dsz (56): Roy Chardon (515) notes that ... *dsz CE ... decrements Reg 0 without causing an error condition (analagous to the V2N2p1 sequence for the SR-52).

0 divide Error State (56): Roy also notes that several functions are affected during a 0 div error state for the 56: notably, the statistical and rectangular/polar; also that PROD executes as INV PROD (contrary to Dave Johnston's finding (V1N2p6)).

TI NOTES

The customer service toll-free number (V1N1p6 should be changed to 800-858-1802; use 806-747-3841 (not toll-free) for technical assistance.

For those of you going to Dallas for the NCC (13-16 June), TI expects to have several applications people there to answer questions concerning the new machines, and if they're not too swamped, you might be able to get your hands on a 57, 58, or 59. The Professional Calculator Division is in the process of moving to Lubbock, so just who will be where and when at the NCC can best be found out by contacting TI at the Dallas Convention Center when you get there.

TI is starting a separate program exchange for the 59 (PPX-59), with its own newsletter and catalog. More details as they become available. There are no plans to drop PPX-52.

COPING WITH BUILT IN FUNCTION ANOMALIES

Since most built-in functions produce approximations to desired results, and since the accuracy of the approximations is often dependent upon the magnitude of inputs, it is not surprising that some data regions cause problems. Manufacturers are motivated to find a compromise between complicated algorithms that handle all cases with equal accuracy and simple ones that break down significantly for specific inputs. It appears that HP considers it cost effective to spend more for "cleaner" functions than TI does, although the best algorithms don't necessarily cost the most to implement. Seemingly unrelated machine architecture features can affect the accuracy of results. For example, HP machines do not flag underflow conditions, which makes it easier to process data whose magnitude is close to zero. Joel Pitcairn (514) notes that knowledge of algorithm construction would enable users to identify critical data regions, but unfortunately the manufacturers consider such information to be proprietary. A TI spokesman points out that even publishing information identifying critical data regions would give the competition useful ammunition. So I suspect the best approach is to find out as much as you can about the way a machine processes problems of interest before you buy, and then for significant programs, devise test procedures that cover all expected data ranges. And then ofcourse, when in doubt, round! For you mathematicians, Joel and Barbara Osofsky (420) both note that lnx of numbers in the neighborhood of 1.0000007 results in only 6-place accuracy. Joel also finds similar accuracy degredation for sin .00007 and tan 1.0000000002. Incidently, the 58 and 59's built-in math functions appear to be the same as for the 52 and 56.