52-NOTES

Volume 3 Number 6 48/39/38 June 1978

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

FRIENDLY COMPETITION

It's a bit disappointing that none of HP's newly announced Series E calculators is even on a par with the 67. So there are still no HP PPCs qualified to match the TI-58 or 59 in our on-going Friendly Competition (V2N4) with HP users. However, continuing on with an old counter-challenge, Hal Brown (V2N8p3 and V2N12p5) has collaborated with another HP user in improving his matrix program (see PPC JOURNAL (formerly 65-NOTES) V5N4p6,24) such that although manual rearrangements are still required in some cases, the new version runs faster, and can solve a system of 5 simultaneous equations as well as calculate the determinant and inverse of a 5 X 5 matrix. Avid SR-52 users may find it challenging to translate the Brown/Robinson program, and see if they can produce as good or better 52 versions.

But until HP (or some other PPC manufacturer does catch up, 58/59 users will have to content themselves with same-machine challenges. The calendar one (V3N5) is turning into a lively contest, and it is now apparent that beating 2 minutes per printed month (V3N5p2) isn't too hard to do. Lou Cargile (625) and Fred Fitzgerald (252) are about even with programs which print a year in ten minutes. Lou uses indirect subroutine calls to specific entry points in long strings of numeral instructions to generate print-code strings in the display. Such calls are fast, but time is sometimes wasted getting to the next rtn after the display has been filled, and this approach requires specialized processing for non-31 day months. Fred takes a related, but somewhat different approach, using indirect subroutine calls to short sequences of the form: a.bc rtn, where a is the print-code for the tens place, and bc the print-code for the units place for all the days (1-31).

But it turns out that indirect recalls are faster than indirect subroutine calls to even the shortest sequence: rtn, and Maurice Swinnen (779) was able to devise a single-month printer which executes in 40-45 seconds using indirect recalls from a block of consecutive registers, each containing 2 print codes for 2 of the integers 1-31. Incidently, Maurice included a detailed algorithm along with mag cards and listings, greatly facilitating the job of determining how his program works. While close to the fastest calendar printer I've yet seen, Maurice's program does a good deal of special-case handling, which offsets the speed gain of the print-code retrieval machanization, and leaves no room for routines to handle more than one month at a time.

The SR-52 Users Club is a non-profit loosely organized 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.

Bill Skillman (710) worked out a simpler way to pack print buffers for all cases from a block of consecutive registers, each containing print-code for a single day. His approach is the fastest yet: 7½ minutes per year. The program which follows is a revision of Bill's, cutting size and execution time a bit with the substitution of INV Dszs for x-t comparisons, and a bunch of equally spaced GTOs for a double indirect mechanization (the latter being more costly in registers, and slightly slower, even though more elegant).

Record banks 1-3 with turn-on partition.

TI-59/PC-100A Program: Calendar Printer   Bill Skillman (710)/Ed

Program Listing:

```
000:  GTO 127 GTO 025 GTO 040 GTO 051 GTO 066 GTO 097 GTO 112 LA' Op0
025:  R*2 Op22 EE 6 INV EE Op1 + INV Dsz4 1' R*2 Op22 = Op1 INV Dsz4 1'
051:  R*2 Op22 EE 4 INV EE Op2 + INV Dsz4 1' R*2 Op22 EE 2 ± INV EE S5
077:  Int = Op2 R5 INV Int EE 10 INV EE Op3 + INV Dsz4 1' R*2 Op22
101:  EE 2 = INV EE Op3 INV Dsz4 1' R*2 Op22 EE 6 INV EE Op4 + INV Dsz4
126:  1' R*2 Op22 = INV EE Op 4 Op5 Dsz4 A' Adv CLR rtn L1' Op5 CLR
149:  Adv rtn LB R1 + 64 = S07 1 S2 R9 S3 Pgm20 SBR 086 S0 ÷ xXt 7 ± =
180:  Int X 7 + xXt = X 3 = S6 Op21 R9 S3 Pgm20 SBR 086 - R0 = S4 Op0
210:  R*7 Op4 R9 Op6 R64 Op1 R63 Op2 R62 Op3 R61 Op4 Op5 Op00 30 S2
242:  GTO*6 LA S9 S3 R/S S01 1 xXt Pgm1 SBR Write INV x=t Grad 8 Op17
265:  Adv Adv Adv rtn LC 13 - R1 = S8 B Dsz8 279 rtn LD - R9 + 1 =
293:  S77 C Op29 1 S1 Dsz77 295 Adv Adv R/S
```

Pre-stored Data:

```
30:  2 3 4 5 6 7 8 9 12 201 202 203 204 205 206 207 208 209 212 301
50:  302 303 304 305 306 307 308 309 312 401 402 2135003613 1700372300
63:  37410043 3641003032 251331 211714 301335 133335 301345 25413117
71:  25412745 134122 36173337 321537 313242 161715
```

User Instructions:

1. Key start year, press A.
2. Key start month, press R/S; if wrong module, display flashes.
   For 1 Month: Press B; for balance of year, press C; for specified interval, key end year, press D.

Incidently, as some of us found from frustrating experience, ML-20 sometimes alters input month or year. The routine which calculates the day of the week (D or SBR 177) unfortunately uses Reg 1 for temporary storage, and when the input months are either January or February, steps 112-115 decrement the year.

Carl Seel (328) notes that "...the 4 year leap year rule doesn't hold for centesimal years (1900, 2000, 2100 , ...) which aren't evenly divisible by 400." ML-20 accounts for this, but as Charles Kluepfel (757) notes, Lou Cargile's V3N5p3 program doesn't, since it neither uses ML-20 to calculate the number of days in February, nor accounts for the centesimal rule in any other way. Charles also notes: "As to the far future, there is question whether 4000 AD will be leap year or not, as further corrections may be necessary to the calendar." Members wishing to explore time keeping in greater detail are invited to read "Measuring Time" by T R Parkin in POPULAR COMPUTING V6N6p7. In this first of a 2-part article, Parkin presents recent Naval Observatory figures defining the mean solar year as 365.2421946412 days on 1 Jan 1975 and "...slowing down by .53 seconds for each century after Jan 1900".

After printing V3N6p2, I received a calendar printer program from Panos Galidas (207) which uses Maurice's doubled print-code method, but with a simpler way to handle all the special cases, and which prints a year in 5½ minutes. But as I examined Panos' approach (via a detailed algorithm, which like Maurice's, expedited comprehension), it appeared that further speed enhancements could be made, along with squeezing the code into 3 memory banks. It turned out that some of Bill Skillman's approach was applicable, and that using H5-8 in place of Op1-4 print buffer loading instructions, would speed up the required left-right number shifting. The program which follows prints a year in just under 5 minutes (1978: 4 min 53 seconds), and actually has more prestored data than is evident at first glance. These data are listed as steps 465-719 for input convenience. I suggest first writing into program memory all listed steps: 000-326 and 465-719 with a 1 Op 17 partition; then storing the Reg 62-78 constants with an 8 Op 17 partition, in order to avoid data shifts due to insertions or deletions; record banks 1-3 with a 6 Op 17 partition. During program execution there is an effective 327.78 partition. The V3N6p2 user instructions apply, except that there is no CROM module test.

TI-59/PC-100A Program: Calendar Printer Ed,207,221,252,625,710,779

Program Listing:

```
000:  30 S0 Op27 GTO 115 31 S0 Nop Nop GTO 061 30 S0 Op27 GTO 061 31
029:  S0 Nop Nop GTO 078 31 S0 Nop Nop GTO 096 30 S0 Op27 GTO 096 31
056:  S0 GTO 115 R*0 INV Dsz7 139 H5 INV Dsz7 172 2 SUM0 R*0 + 99.99
086:  = INV Dsz7 145 H6 Op20 R*0 INV Int INV Dsz7 157 H7 INV Dsz7 172
112:  2 SUM0 R*0 INV Dsz7 169 H8 INV Dsz7 172 2 SUM0 Op5 Op0 GTO 061
139:  EE H5 GTO 172 + R62 - R62 = H6 GTO 172 + R62 - R62 = H7 GTO 172
169:  EE H8 Op5 Op0 Adv CLR rtn LB R1 + 66 = S08 1 S2 R9 S3 Pgm20
198:  SBR086 S0 ÷ xXt 7 ± = Int X 7 + xXt = X 9 = S6 Op21 R9 S3 Pgm20
227:  SBR086 -R0 = S7 Op00 8 Op17 R*8 Op4 R9 Op06 7 Op17 R66 Op1 R65
258:  Op2 R64 Op3 R63 Op4 Op5 Op0 GTO*6 LA S9 S3 R/S S1 rtn LC 13 -
289:  R1 = S4 SBR181 Dsz4 294 rtn LD - R9 + 1 = S5 SBR286 Op29 1 S1
320:  Dsz5 312 Adv Adv R/S
465:  0 0 0 CLR'Ind E' E' 0 CLR' Ind 0 E' Ind E' E' 0 E' Ind 0 CLR'
485:  p31 E' E' 0 CLR' p31 0 E' p31 E' E' 0 E' p31 0 0 p31 E' E' 0 0
506:  p31 0 Rad tan E' E' 0 Rad tan 0 Deg tan E' E' 0 Deg tan 0 Abs
525:  tan E' E' 0 Abs tan 0 Ind tan E' E' 0 Ind tan 0 tan tan E' E'
544:  0 tan tan 0 CLR' tan E' E' 0 CLR' tan 0 E' tan E' E' 0 E' tan 0
564:  CLR' p21 E' E' 0 CLR' p21 0 E' p21 E' E' 0 E' p21 0 0 p21 E' E'
584:  0 0 p21 0 Rad CLR' E' E' 0 Rad CLR' 0 Deg CLR' E' E' 0 Deg CLR'
603:  0 Abs CLR'E' E' 0 Abs CLR' 0 Ind CLR' E' E' 0 Ind CLR' 0 tan CLR'
622:  E' E' 0 tan CLR' 0 CLR' CLR' E' E' 0 CLR' CLR' 0 E' CLR' E' E'
640:  0 E' CLR' 0 CLR' 1 E' E' 0 CLR' 1 0 E' 1 E' E' 0 E' 1 0 0 1 E'
663:  E' 0 0 1 0 Rad 0 E' E' 0 Rad 0 0 Deg 0 E' E' 0 Deg 0 0 Abs 0 E'
687:  E' 0 Abs 0 0 Ind 0 E' E' 0 Ind 0 0 tan 0 E' E' 0 tan 0 0 CLR' 0
710:  E' E' 0 CLR 0 0 0 0 E' E'
```

Prestored Data:

```
62:  1000 2135003613 1700372300 37410043 3641003032 251331 211714
69:  301335 133335 301345 25413117 25412745 134122 36173337 321537
77:  313242 161715
```

For those of us actively involved in the calendar printer competition, it has been (and may continue to be: I just thought of a way to save 24 steps and a little time in the V3N6p3 equally-spaced GTOs... and who knows, maybe there is a 4 min/year program on someone's drawing board...) a time-consuming, but stimulating and rewarding contest. To the few who already knew the important programming tools, techniques, and approaches illustrated by 52-NOTES calendar coverage, I apologize for wasting the space. But for most, taking the time and devoting the patience to finding out how and why each of the listed calendar programs works, and why some are faster than others, will result in significantly broadening programming skills, and I hope, justify the 52-NOTES coverage. I expect to extend the range of future Friendly Competition topics, but there aren't many non-trivial problems whose solutions are as universally understood as optimizing a PPC calendar printer.

BOOK REVIEW: PROGRAMMABLE CALCULATORS - BUSINESS APPLICATIONS; by Aronofsky, Frame and Greynolds; 203 pp, $8.95, McGraw-Hill 1978. Tracy Bollinger (864) and James Merrill (693) brought this new book to my attention, and both report in essence that it would be most useful to those novice users who need more spoon-feeding than the users manuals provide, but who don't care to go beyond the manuals' programming scope. The title might more appropriately be worded: "Learning How to Program Your TI-58/59/PC-100A For Elementary Business Applications" since non-TI machines are not covered at all, and non-58/59 very little. The authors teach at Southern Methodist University, and write in an easy-to-comprehend textbook style, covering manual as well as basic programming topics. While many functions and concepts are more fully explained than in the 58/59 users manual, there are misleading oversimplifications, and a few technical errors. For example, Sec 8-2 titled: Indirect Instructions covers only indirect stores and recalls (although the text claims to cover indirect summing as well); page 143 confuses card protection with program protection, and page 172 incorrectly states that a programmed CP resets all flags. Important (to many of us) unannounced features such as the HIR operations and 2-digit Dsz registers are not mentioned at all.

TIPS AND MISCELLANY

<u>Membership Address Changes</u>: 185: 856th ASA Co APO NY, 09039; 246: 2554 Key Largo Ln Ft Lauderdale, FL 33312; 491: 10571 Kerrigan Ct Santee, CA 92071; 493: 2800 S Lamb Blvd #181 Las Vegas, NV 89121; 581: 10172 Saluda Ave San Diego, CA 92126; 663: change 8333 to 7830; 684: 1505 Locust St Pasadena, CA 91106; 755: 1046 Gen Allen Ln West Chester, PA 19380; 845: 10409 Towlston Rd Fairfax, VA 22030; 997: M E Holthaus 1181 E Walnut Bldg 5 #6 Carbondale, IL 62901.

<u>CROM RST Behavior</u>: Fred Fitzgerald (252) found a code 81 at ML-19 step 377, and reports that it executes as RST R/S (which I confirm). Incidently, Fred has made copies of CROM program listings arranged on 8½ X 11 sheets for modules 1 and 7 and will provide these to members, at $3.00 for one, $4.00 for both. See V2N5p4 for Fred's current address.

<u>Structural and Applied Mechnics PPC Applications</u>: Bob Thacker (30) has volunteered as a focal point for interested MEs, and wonders if anyone has tried clock speed-up mods for the 59.

Last-Step Instruction Execution Under Programmed List (58/59/PC): Jared Weinberger (221) has found that the execution of a last-step rtn during a programmed List (V2N10p3) applies also to other last-step instructions. A sequence of the form: LA B seq1 rtn LB List seq2 f seq3 rtn, where f is an instruction located in the last step of the current partition, on a call to A lists seq2 and instruction f, and executes instruction f. If f is rtn, seq1 is executed following the listing of seq2 f; if f is R/S, execution halts at the last step; if f is RST, execution continues at step 000; but if f is any other instruction, an error condition is set and the program halts following its execution. Composite instructions beginning with f are completable manually from the keyboard. If f is C, and if routine C repartitions memory such that seq3 is executable, a call to A lists seq2 C, executes C then seq3, seq1. Members are invited to find new practical applications for any of this behavior.

Subroutine Execution From the Printer (52,56,58,59,PC): Jared found that with one of the new machines connected to the printer, a manually keyed SBR PRINT or SBR ADV will initiate processing at defined Prt or Adv labels, where the PRINT or ADV are the printer keys. With the 52, SBR PRINT or SBR ADV execute as GTO Prt or GTO Adv, as might be expected; with the 56, error states are produced. While this behavior may have little practical value, it demonstrates another backward link from the printer to the PPC, akin to PRINT or ADV writing code into PPC memory when in LRN mode (V1N3p6).

An All-Digits Generator: George Hartwig (638) suggests that 80 ÷ 81 = is a short way to produce .987654321 . However, the 13-digit machines actually produce .9876543209876 and the 11-place 57 produces .98765432098. So depending upon use, it might be well to follow the 80÷81= with EE INV EE.

Strange Pause Effects (58/59): George found that if while holding down the GTO key during program execution to pause-display the results of each step, you key other keys, strange things happen in the display. Results appear to be digit-value and number-size dependent.

A Search For Useful CROM Dsz Loops (58/59): For routines which loop many times, significant execution time can be saved if the required code can be found in a CROM sequence (see V2N12p2). For example a sequence of the form: LA R*n x=t C Dszn A ...LC... will perform sequential searches through a specified block of registers. But to do this acceptably for an unintended application, Lbl C needs to be followed by a rtn, with intervening steps that do no harm, and don't waste too much time. Members are invited to search CROM listings for such a sequence, or other potentially useful loops which could save time.

Shortened Paper Drive (PC-100/100A): Fred Fish (606) notes that the SR-60A manual advises users to precede the first print command with Adv to preclude reduced paper movement caused by tear-off tension through the drive mechanism, and that such advice applies also to the PC-100 and 100A. I find that the problem is minimized when the tear-angle is minimized: pull the paper down across the teeth, not up.

More on Runaway Motor (V3N3p5): S H Hartman (920) reports that the card-drive motor can (always?) be stopped by keying INV Write.

Linear Programming: John Hirsh (736) brings up this topic, and solicits membership interest vis-a-vis PPC applications.

EE On a 0-Digit Mantissa (58/59): James Merrill (693) notes that the manually keyed sequence: 777777777 EE 34 = displays as 7.7777778D09, but as 7777777773 when followed by INV EE. They both represent the same number in the display register: 7.777777773000D09 which indicates that the 3 following the EE got appended to the mantissa. In general, a manually keyed (or programmed and run) sequence of the form: abcdefghi EE j produces the integer: abcdefghij. Apparently the EE is ignored as a decapower prefix when the mantissa is too large for a turn-on display, but small enough to accept another digit, if such is keyed following the EE.

A Manual SBR Quirk (58/59): Izzy Nelken (576) found that a manually keyed SBR 0 x^2 in RUN mode squares the display, then executes code starting at step 000. This generalizes to the following: SBR ss m executes monadic function m on the display, then code starting at step 0ss; and SBR ss d executes dyadic function d on the display and on a second operand produced by the code starting at step 0ss.

Computer Listings of SR-52 Programs: Roger Cowell (1010) has written a computer program in HP-2000E BASIC which he runs from a highschool interactive terminal to print 58/59/PC-type listings from 2-digit SR-52 instruction code inputs. Write Roger at 115 Fairview Cir Webster, NY 14580 for details.

CROM PARTITIONING: Arthur Ehrlich (969) reports that steps 164-166 of LE-13 are 6 Op 17, which when executed in a 58, lock out all of user memory. So don't try to call that part of LE-13 from a user program in a 58!

ML-24/25 Precision: John Mickelsen (990) reports that the statute to nautical miles conversion factor is inaccurate (good to only 8 places). Users of these programs should check the values listed on pages 84 and 86 of the ML Manual against the accepted standards, and note to how many places they are good.

Card-Drive Adjustments (59): Some hardware-knowledgeable users have attempted to reduce card read/write errors by adjusting a motor speed pot. But Robert Cruse (889) quotes a TI source as saying that "...alignment of the four heads and the azimuth of each..." is more critical than speed adjustment. Members are reminded that user-performed hardware adjustments/mods may invalidate warrantees. Robert also reports the existence of "...a new book on the repair and maintenance of the TI-59 for $11.00", but did not disclose the source or availability.

CLR For INV EE: Lou Cargile (625) and Fred Fish (606) suggest waiting until a CLR can be used safely to return the display to an INV EE format, saving a step.

ROUTINES

CROM-Call Selective Trace (58/59/PC): For CROM programs such as ML-8 which call user-written routines, Jared Weinberger (221) suggests: LA' Ifflg0 B' Stflg9 B' INV Stflg9 Stflg0 rtn LB'...rtn as a preprocessor to the called routine (in this case A') which causes trace-printing of the routine itself (B') the first time it is called, but not thereafter.

A General Purpose Integer Routine (V2N12p1): George Rapasy (981) devised the sequence: ...CP xGEt 1' (CE - 1) Ll' Int... which is 3 steps shorter than Joel's, protects pending arithmetic, and can be made to work on all the TI PPCs except the 52.