52-NOTES

Volume 4 Number 1 48/39/38 January 1979

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

OFF-THE-SHELF CUSTOM CROMs

A number of companies have commissioned TI to produce CROMs to their specifications, in some cases with the intent of marketing them as part of special-purpose 59/PC packages for small business and professional groups. Although the vendors of such packages obviously prefer to sell them in as comprehensive a configuration as they can, including software support and hardware maintenance, they may be willing in some cases to sell the modules separately to users who have their own machines and who need no help either using the custom CROMs directly or in writing RAM programs to call CROM routines.

From what I've learned so far from people who have bought custom CROMs, code is apt to be protected (can't be downloaded), and a single module could cost a second buyer $600 or so. This, ofcourse, includes a share of the software engineering which goes into its design and checkout, as well as TI's charges for emulator time (typically $2000) and fabrication, and vendor profit. This compares with going to TI directly with your own programs, and paying $12,000 for 250 units, $17,000 for 500, or $25,000 for 1000.

Anyway, here are a few facts, figures and contacts for members who may wish to take advantage of already-made custom CROMs. I make no representation whatsoever as to quality or utility, but merely pass this information along, hoping it will be useful to some members.

Reynolds & Reynolds, Dayton, OH 45401 (513) 226-0808 x539 is marketing a 59/PC custom CROM package which they call the COMPEG 2001 microprocessor for payroll computations. TI identifiers have been covered over by R&R markings, and a colored keyboard overlay labels special-use keys. The system is designed to automate so-called pegboard payroll accounting, with one mag-card "... containing the entire payroll history of one employee." The total package costs $1300, plus $100 per year Maintenance fee.

Horizons Technology, 7830 Clairmont Mesa Blvd San Diego, CA 92111 (714) 292-8331 "... currently does custom programming for government agencies. We design and program custom CROMs, produce our own magnetic cards, and distribute comprehensive software documents. Most of our customers are given a large system that is cost-effective; small individualized programming jobs would have to be assessed on a case by case basis."

Z-Comp Division of Davis-Gilbert Co, Inc #616 The Landmark Bldg Alamo Plaza, San Antonio, TX 78205 (512) 227-7777 has a Texas-oriented custom CROM covering consumer installment financing loan computations,

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 ($10.00 US abroad) includes 6 issues of 52-NOTES; back issues start June 1976 @ $1.00 each ($1.67) abroad).

aimed at banks as primary users. Prices are $970 with 59/PC, $835 with 59 alone, $795 with 58/PC, $660 with 58 alone, and $575 for the CROM only. This company plans to have other modules made which would apply specifically to other states.

Other custom CROMs include: Oil and gas drilling engineering computations by IMCO Services Box 22605 Houston, TX 77027; and Ophthalmological computations by Sonometrics, Inc 16 W 61st St NY, NY 10021. Members with additional information concerning either the availability or quality of custom CROMs are invited to share same.

THE GAME OF LIFE

A decade or so ago, University of Cambridge (England) mathematician John Horton Conway developed a solitaire game which he called LIFE, having as its purpose the challenge to the player to accurately predict successive changes to initial starting patterns resulting from the application of a few simple rules, and to formulate behavioral hypotheses to support the predictions. As originally conceived, LIFE also challenges the player to avoid easy-to-make playboard mistakes! The latter can be, and has been eliminated by computer mechanization, giving way to a new challenge: Writing efficient programs to produce any desired number of successive patterns from input initial patterns. It is this challenge which will be addressed in 52-NOTES. Members wishing to pursue the playing challenges are referred to Martin Gardner's Mathematical Games discussions, beginning with the October 1970 issue of Scientific American.

The programming requirement is to take an input 2-dimensional array of elements (organisms) and produce successive new arrays, each derived from its predecessor according to 3 rules: 1) An existing element survives if it has 2 or 3 neighbors, 2) A new element is created (born) when an empty position has exactly 3 neighbors, and 3) An element vanishes (dies) when condition 1) is not met. A neighbor is defined as an existing element occupying any of 8 possible positions surrounding a given position, and all changes are considered to occur simultaneously. Conceptually, the 2-dimensional field is infinitely large, to allow patterns to move/grow without restriction. Practically, ofcourse, a physical board, or computer memory is of finite size, imposing an upper limit on pattern movement or growth.

A practical 59/PC limit would appear to be a 20 X 20 field, within which a reasonable amount of movement/growth can be accommodated by restricting input patterns to the center 10 X 10 positions. Over the past year or so, I've received 59/PC LIFE programs from Mike Brown (128), Rusty Wright (581), Lou Cargile (625), Bill Skillman (710), and Maurice Swinnen (779), the latter being a translation of a German version appearing in DISPLAY (V4N1/2) written by a Peter Klinghardt, whose program inspired Bill to write his. Of these, I judge Bill's to be the best, and his 13 Nov 78 version lists below. Bill does some tricky things, and it's a challenge just to figure out how his program works! He chose the - to represent an empty position, the 0 (zero) for an established organism, a Q for a new birth, and a blank to signify a recent death. While Bill's choices are good, they may not be the best for all users, and members wishing to experiment with other characters can do so by altering the numerals at steps 147-148, 159-160, and 162-163 which as written produce the blank, 0, and Q respectively. For example, writing 50 at steps 159-160, and 04 at steps 162-163 changes the 0 to II and Q to I.

TI-59/PC Program: LIFE   Bill Skillman (710)

User Instructions: Key g.i where g is the generation number (0 or 1 if just starting) and i is any desired identifier, press E. Input the starting pattern as keyed strings of zeros and ones, up to ten per line, press R/S following each input, see next line number displayed. Processing begins automatically following input of the tenth line, or may be initiated sooner by pressing A. Successive generations are printed indefinitely until stopped with R/S. To prepare to continue a run from one turn-on to another, push R/S immediately following the printing of the next population sum (s   Σ), note the current step number, and record banks 3 and 4. At next turn-on, read the recorded card into banks 3 and 4 (in addition to a program card into banks 1 and 2), and press GTO 222 R/S, unless the last stopped step was between 216-220, in which case press GTO 220 R/S.

Program Listing:

```
000:  LC' = SUM55 LE' INV Int X 1 EE 10 = INV EE rtn LB' R56 LD INV EE
026:  Op*1 Op31 rtn LD' Op24 R*4 S59 0 S53 R54 ÷ 1 EE 5 = INV Int xXt
052:  0 INV xGEt 069 B' 1 EE 5 ± Prd54 Prd59 GTO 072 SBR086 3 xXt R54
076:  xGEt 341 B' 1 EE 4GTO 347 R56 S08 5 S9 CP R54 Int X .1 Prd53 =
103:  S54 x=t 172 R59 Int X .1 = S59 INV Int INV x=t 124 INV Stflg2 Nop
127:  R54 INV Int xXt .3 x=t 154 INV Ifflg2 172 .4 x=t 159 60 SUM8 GTO
152:  172 INV Ifflg2 162 47 + 14 = SUM08 1 SUM53 Op20 .01 Prd8 Dsz9 093
181:  R8 SBR009 D rtn 221731 Op4 R2 Op6 Nop rtn CLR 77 Op04 0 S8 S9
212:  Exc0 Op6 Adv CP x=t R/S Op22 SBR188 20 S03 7 S05 9 S4 S06 11 S7
241:  CP SBR325 S54 x=t 262 ÷ 10 ÷ SUM54 1 EE 8 C' SUM54 SBR325 x=t 284
268:  S53 SUM55 X 10 ÷ SUM55 100 C' SUM54 4 S1 D' Exc58 Op35 S*5 Op25
296:  0 Exc55 S54 D' Op5 Exc57 S*5 Dsz3 241 0 Exc58 S48 0 Exc57 S49 GTO
323:  202 Op25 Op26 Op27 R*5 + R*6 + R*7 = rtn SBR086 1 EE 9 X R53 =
351:  rtn LE CMs S2 SBR188 2020202020 S56 20 S01 10 S07 11 - R7 = R/S
387:  . Prt ÷ 1 EE 5 X 3 = S*1 E' Exc*1 Int Op21 S*1 Op21 Dsz7 380 LA
414:  SBR446 10 S07 20 S06 4 S1 SBR463 SBR463 Op5 Dsz7 425 SBR446 GTO
444:  202 5 S05 4 S1 B' D D D Op5 Dsz5 456 rtn R*6 S54 2 S4 D' S*6
473:  Op26 rtn
```

Note: Bill's program was written with Adv at step 200. Also note that the . at step 387 is to clear a hardened display for all-zero inputs.

Members are invited to try to out-do Bill with better (faster, more versatile, etc) LIFE programs.

MORE ON PROGRAM COPYRIGHT (V3N5p4,5)

Ken Widelitz explores the topic further (KILOBAUD Jan 79 pp 9,10), reviewing the National Commission on New Technological Uses of Copyrighted Works (CONTU) final report, and its recommendations to Congress for ammendments to the Copyright Act of 1976. While Widelitz states that he agrees with the conclusions of the report, much of the text he airs leaves me with the impression that for all of its attempted specificity, the CONTU report fails to provide an achievable means to define, much less protect the rights of originators of computer readable works.

My own feeling is that the only practical way to prevent program plagiarism is to deny the type of access to the program that makes copying possible. But that's easier said than done. Users have already succeeded in getting at and copying the 59 code on a "protected" card, and I suppose there might be some effort being devoted to revealing protected CROM code. However, the CROMs present an interesting special case: Revelation of their code may actually enhance their marketability, since potential users would be able to verify what they're getting, and could very well find that buying the CROM (TI or private custom off-the-shelf) beatsfilling up RAM with plagiarized slower-executing code.

Nevertheless, TI considers its CROMs to be copyrighted, even though it doesn't publish listings. The label on a CROM does indeed contain the circled c and the year, but perhaps that could be construed to apply only to the label itself! The question of whether programs cast in ROM are patentable or copyrightable appears not to have been conclusively answered yet, since as Widelitz states: "Some companies have successfully patented computer programs by getting the patent on 'firmware'. Since patent and copyright are mutually exclusive, it will be interesting to see how this conflict is resolved."

Fred Fish's User Survival Guide (V3N3p1) may turn out to be an interesting test case, since it is copyrighted, and lists the ML CROM programs. Who knows, it might turn out that TI would be required to obtain Fred's permission to publish its own listings!

BOOK REVIEW: SOURCEBOOK FOR PROGRAMMABLE CALCULATORS, TI LEARNING CENTER, 1978, $12.95, 416 pages

This TI publication was designed to provide the user of 58/59 machines with a wide variety of straight-forward applications programs, each accompanied by a brief tutorial. Topics span number theory, algebra, trig, calculas, statistics, music, business, economics, biology, medicine, engineering, physics, and astronomy, with pedagogical treatment generally as light as is possible for each subject. No programming experience is assumed, and program understandability appears to have dominated over I/O and execution efficiency considerations when the routines were designed.

While there is no attempt to teach programming techniques, each listing and corresponding user instructions include Purpose/Comments, and Display/Comments, respectively, facilitating user comprehension. Master Library routines are called when appropriate, but there are no warnings to the user to assure ML module installation.

I haven't yet checked out all the algorithms and routines, but they appear to have the backing of a panel of college-level educators, and I recommend this book to TI PPC users who need help getting started at mechanizing algorithms on their machines, and to users of computing machinery in general who are looking for brief introductions to the covered topics. Users are, ofcourse, advised to be on the lookout for errors. For example, Jerry Johnston (493) spotted some questionable code, and suggests that 2 STO instructions at locations 042 and 061 of the Equation of Time Routine (page 11-39) should be deleted. I suspect that the STOs in question should be replaced by GTOs.

CROM RNG LIMITATIONS

Don Huffman (902) has found that TI's random number generator used in several of the CROMs doesn't always guarantee a period of m (199017, as stated on page 54 of the ML Manual). In fact, a bit of trial and error turned up 2 unrelated 13-digit seeds which Don found produced sequences with periods under 500, which means that a little search produced over 900 seeds which would initiate sequences of cycle length less than 500.

It turns out that with the linear congruential method, a minimum cycle of length m is only guaranteed when the seed is an integer and subsequent seeds are carried as exact reals (see Theorem A in Knuth's

The Art of Computer Programming Vol 2 p 15, and Exercise 1 p 31). But it may be that TI's routine carries most successive reals to sufficient precision to meet the exact reals requirement, provided the input seed is an integer. Has anyone found an integer seed which produces a cycle shorter than m?

THE PC-100C PRINTER

TI is currently marketing domestically a printer labeled PC-100C, which appears to be similar to the 100B (V3N9p6). The 100C differs from the 100A primarily by being adaptable to only the 58/59 machines, having a non-detachable power cord, and allowing calculator operation with PC power off.

A 100C which I've tried is noisier than other PCs I've used, a characteristic which may just be an isolated case. Calculator operation with PC power off is the same as when on, as far as PC connection sensing is concerned (i.e. Ops 1-4 load print buffers, Op 7 on a display greater than 19 causes an error condition, and Op 8 initiates a label search to the end of the current partition (but without printing labels)). The 3% execution slowdown (V3N7p6) appears to prevail with PC power either on or off.

Externally, the 100C looks about the same as the other PCs, but a look at the inside, with the top removed, shows quite a few changes. The PC board (here, PC means printed circuit, not print cradle!) shows marked differences in circuitry and components. What appear to be 2 high-power diodes are fastened to 2 much smaller heat sinks, eliminating the 100A's one large one, and a single large capacitor has been replaced by 2 smaller ones. The print assembly appears to be about the same, although the step-moter and transmission are different in appearance, and made by a different manufacturer. (Maybe this accounts for the higher print-noise level). Some components on the PC board nearest the tape supply have been relocated, and a cardboard guard eliminated.

Further comparisons among the various PC-100 models are invited from the membership.

A CROM Op Ind

Dave Leising (890) has found an ML-19 point (SBR 172) to execute an effective Op*53, making it possible to have any of the 40 Op functions executed in the CROM. One that he found interesting with printer connection is Op 8, which produces a somewhat odd mnemonicless listing of RAM code. To see what happens, at turn-on, key 10 Op 17 8 STO 53, write some code at both ends of the partition, press (latch) the TRACE key, and key Pgm 19 SBR 172. The first 2 steps are printed as 000 01 and 001 00 followed by the "real" steps 001-158. Step 159 is printed as a code 43 (RCL), and execution halts following an effective Prt of the 8 in the display. Inspection of the last step confirms the code 43, but a look at step 000 shows a code 85 (+). If during the listing the TRACE key is unlatched, label-search appears to continue without further printing (even labels are ignored). If the TRACE key is again latched before the search ends, printing resumes with the remaining steps.

While the CROM Op*53 executing as Op 8 causes execution to halt at the end of the label search, other Ops let the CROM code continue to execute, resulting in error setting and endless looping, unless several key registers have been appropriately initialized. So be prepared to key RST to halt execution, when necessary. Members are invited to examine the behavior of other CROM-executed Ops, and to share their findings.

TIPS AND MISCELLANY

Printing 12 Places (58/59/PC): Mike Brown (128) has discovered an efficient way to get 12 of the 13 mantissa places printed. He found that in TRACE mode, a RCL Ind on a register whose contents is a 13-place integer causes 12 of those places to be printed! (albeit along with a little garbage and an error condition). Invoke the following sequence by pressing A, with the desired number to be printed in the display register: 000: R/S LA Prt S00 1 EE 50 Prd0 CLR Stflg9 R*0 CLR RST and see the number as displayed printed, followed by a trace of R*0, which includes the unscaled 12 MSDs of the input number. The 1 EE 50 is to assure that the input number is an integer, and may be replaced by a larger or smaller scaling factor, depending upon the expected range of inputs. The RST should be replaced by INV Stflg 9 rtn and the R/S at step 000 deleted, if this routine is to be subroutine callable.

Revealed ROM Constants (V3N12p5): John VanWye (982) suggests that the non-normalized constants which Bill Skillman associates with steps 384-447 are not scaled at all (the machine treats them as fixed rather than floating point (V3N7p1,2)). Thus each 8-step register actually holds a 16-place number, with an implied decimal point between the 2 MSDs. But it's not clear how many of the places were intended to be good. John finds ln 1.0001 at steps 424-431 good to all 16 places, but π and π/2 at steps 496-503 and 488-495 respectively appear to be good to only 13 places.

Friendly Competition (V3N12p6): Maurice reports receiving a "3" routine from Richard G Snow (brother of Robert G (212)) tying my 11-step routine, but here's a 58/59 one from Jared Weinberger (221) which is 2 steps shorter: INV lnx INV lnx INV lnx y^x lnx x^2 = lnx lnx. (Recall that INV lnx counts as only one step).

Bridge Deal (V3N10p2): Bill's program has generated at least 2 opposite reactions: Jared considers it insufficiently random at the end, while Lou finds it worthy of replacing his (V3N2p5).

A CROM Identifier: 58/59 users generally working with 2 or more CROMs may find it helpful to insert a CROM-identifying card in the mag/plastic card display slot to remind them which module is in the machine, especially when a CROM-dependent RAM program without a CROM test is run.

Membership Address Changes: 100: C & C of Denver Rm 285 5440 Roslyn St Denver, CO 80216; 372: 572 John Hancock St Orange Park, FL 32073; 606: 2325 N 87th Way Scottsdale, AZ 85251; 751: 500 Newport Ctr Dr #455 Newport Bch, CA 92660; 786: 2416 K St NW #805 Washington DC 20032; 1056: Street address is 7722 (not 722).

An INV HIR Application: Maurice Swinnen (779) notes that although prefixing a HIR operation with INV to change its function may seem to offer no advantage to straight specification, it can save steps in some cases. For example, since INV HIR 31 is the same as HIR 51, and INV HIR 41 is the same as HIR 61, in cases where all 4 register-arithmetic operations are to be subroutine callable, HIR 31 rtn HIR 41 rtn HIR 51 rtn HIR 61 rtn reduces to INV HIR 31 rtn INV HIR 41 rtn, when absolute subroutine addressing either includes or excludes an INV, as desired.

Print Overlays (PC): Roger Cowell (1010) notes that print patterns can be enhanced by repeated transport of the same tape past the print-heads. Depending on the length of the tape to be reprinted, some manual re-registration (alignment) may be required.