for PPC T.

PSEUDO-RANDOM NUMBER GENERATOR TEST PROGRAM

Judging by the number of programs for generating psuedo-random numbers that have appeared in the PPC Journal, there is considerable interest in generating reasonably random numbers. If we could access the clock in the 41C and measure the time between key entries, we would have a true random number generator (RNG), since the clock runs several orders of magnitude faster than controllable human reaction time. Since no one has yet figured out how to do that, we must settle for psuedo-random number generators (PRNGs). All PRNG programs are not equally good, and some are very poor, especially if you take many outputs from a single seed value.

A true RNG has two important properties. First, the probability of any output is equal to that of any other output, so as a series gets longer the proportion of any particular output to the total output approaches 1/C, where C is the number of output choices of the RNG. Second, there is no sequential dependency between numbers, that is, the probability of any number following some previous output number or sequence of such numbers is equal to that of any other number following that previous output number or sequence of numbers. This also means that no matter what output numbers have already been generated, they give you no useful way of predicting what the next outputs will be.

The adequacy of a RNG or PRNG can be evaluated statistically by testing a large sample of outputs for equiprobability and lack of sequential dependency. A common way of doing this testing is to count up not only single outputs (Ø, 1s, 2s, etc., called singlets), but also sequential outputs (Ø followed by Ø, Ø followed by 1, Ø followed by 2, etc., called doublets). Sequential output testing usually includes doublets, and, if there is a theoretical reason to suspect higher order sequential dependencies. triplets, quadruplets, etc.

The following test program was designed to check the PRNG subroutine used in an ESP test program (V8N6P62), where nonpredictability is extremely important. It requires a printer and extended memory. By putting in your own PRNG subroutine as step Ø56 and calling it LBL Ø5, substituting for steps Ø56 through Ø66 in the present program, you can test it. The test program assumes a PRNG integer output between Ø and 9, a scaling factor stored in register Ø8, and a seed stored in register Ø9. LBL Ø1 prompts for a seed for the PRNG and stores it in register Ø9. I use the time, to the nearest second, as a convenient way of getting a starting seed that is beyond my control, entering the seconds as a decimal.

Detailed operation notes are listed with the program. Briefly, the TESTRNG program starts with housekeeping chores of clearing registers and flags, then prompts for the total number

of outputs, trials, wanted from the PRNG (TOTAL N?) in this run, for the number of choices (2 to 1Ø) the PRNG is to have, whether you want the raw PRNG output printed (enter "N" for no, otherwise just press R/S), and whether you want the PRNG output accumulated for later statistical analysis (ANALYZE?) (again enter "N" for no, otherwise R/S).

A note on limitations. A singlet analysis can be carried out if your PRNG output is Ø to 9, but TESTRNG can carry out a doublet analysis only if there are no more than 5 choices (outputs Ø to 4). Doublet analysis will be automatically skipped if C > 5. This is due to the difficulty of addressing enough registers: as it is, registers ØØ through 6Ø (size Ø61) are used. In general I suspect that if your PRNG rountine shows no singlet or doublet biases for outputs Ø through 4 it probably doesn't have them for outputs 5 through 9, but don't count on it. I would be happy if some one worked out how to analyze doublet frequencies from Ø through 9.

The TESTRNG program then prompts for a time or other seed number (SEED?), following which it accesses the PRNG subroutine until it has collected N outputs. This can take a while. For my particular PRNG subroutine, it takes about 4 minutes to collect 1ØØ outputs. When N outputs are present the total number of outputs to date are printed ($\Sigma\Sigma$ =) and a beep signals that a new seed is wanted. Total outputs equal N on the first run of TESTRNG. This feature exists because I usually use my PRNG in the ESP test program for runs of 25 trials or less and then enter a new time seed for each run to assure better randomicity. Your N should be the usual number of outputs you use at a time from your PRNG, or an even larger sample.

At this point you can enter a new time seed and get N more outputs from the PRNG. For analysis purposes, these outputs are accumulated with the preceeding batches of N outputs. For example, if you used a binary PRNG and in 1ØØ trials had 48 Øs and 52 1s, then ran another 1ØØ trials with 46 Øs and 54 1s, the storage registers would cumulate 94 Øs and 1Ø6 1s.

If you elected to store data for analysis, at the end of any run of N trials you can XEQ "CHI" to start analysis of the cumulated results. This analysis does not affect data storage registers, so you can cumulate more data after such an analysis. CHI is a Chi-square statistical test at the singlet and (if C $\leq$ 5) the doublet levels. The resulting values of Chi-square can be looked up in the table included here, for various Cs. If the value you obtain is less than the tabulated value, it would occur by chance alone 94 times in 1ØØ trials, assuming your PRNG output is indeed random, i.e., your PRNG passed the test! This is a simplified overview of the statistical testing, of course, and those who have very stringent requirements for randomicity should consult appropriate statistics texts. Note also that if the expected value of any particular PRNG output is less than 5, the

Chi-square test is generally not valid: the TESTRNG program will automatically print "E < 5" to alert you when this happens.

Here's an example of TESTRNG's operation. Run 2Ø trials of a 3-choice PRNG and analyze, using the current PRNG subroutine. The time seed is 935.54. The PRNG raw output printed is

```
2. 1. 2. Ø. 1. 2.
1. 1. 1. Ø. 1. Ø.
1. 2. 1. 2. Ø. 2.
2. 1.
```

(If anyone knows how to suppress the printing of the decimal point, which takes up useless space here, I would like to hear from you). Upon executing "CHI" the printer gives us

```
4. 9. 7.
S CHI SQ = 1.9ØØ
```

indicating that Ø was generated 4 times, 1 was generated 9 times, and 2 was generated 7 times. The Chi-square analysis uses the formula

$$\text{Chi-square} = \sum \frac{(O-E)^2}{E}$$

For each possible output category (Ø, 1, or 2 in this case) the observed number of appearances of that output (O) has the expected number of appearances (E, 2Ø/3 in this case) subtracted from it, the result is squared and then divided by E, and the results are summated for all possible outputs.

The doublet analysis subroutine then prints out

```
Ø. 3. 1.
2. 2. 4.
2. 4. 1.
D CHI SQ = 7.Ø53
  E < 5
```

indicating that an output of Ø was followed by Ø zero times, Ø was followed by 1 three times, 1 was followed by Ø two times, etc. Chi-square is computed by the same formula, but note that we only have 19 doublets in 2Ø trials, and we sum over 9 output categories. Because the expected frequencies in each category were less than 5, "E < 5" is printed out as a warning that this particular analysis is not valid.

Now we need some devoted member to check every PRNG program published to date and tabulate under what conditions it is or isn't adequately random.

Charles T. Tart (7268)

Number of Choices	Chi-Square Value	
	Singlet	Doublet
2	3.841	9.488
3	5.991	16.919
4	7.815	26.296
5	9.488	37.652
6	11.Ø7Ø	
7	12.592	
8	14.Ø67	
9	16.919	
10	18.3Ø7	

```
Ø1 LBL "TESTRNG"
   Ø2 CF ØØ          Clear flags, store loop control #
   Ø3 CF Ø1            for register clear.
   Ø4 CF Ø2
   Ø5 CF Ø3
   Ø6 Ø.Ø59Ø1
   Ø7 STO 6Ø
   Ø8 LBL Ø2         Loop for storing Øs in register
   Ø9 Ø                ØØ through 59.
1Ø STO IND 6Ø
   11 ISG 6Ø
   12 GTO Ø2
   13 FIX Ø
14 "TOTAL N?"
   15 PROMPT
   16 STO Ø5
   17 LBL Ø7         Prompts for number of outputs of
18 "CHOICES?"          PRNG, choices, C.
   19 PROMPT
   2Ø STO Ø8
   21 1              Computes and stores C-1 for later
   22 -                computational ease.
   23 STO 46
   24 1Ø             Limits C to maximum of 1Ø.
   25 RCL Ø8
   26 X>Y?
   27 GTO Ø7
   28 5              Test:  C > 5?  If so, no doublet analysis data
   29 RCL Ø8            will be stored.
   3Ø X>Y?
   31 SF Ø2
   32 "N"            Should raw PRNG ouput be printed?  Enter "N"
   33 ASTO Y            if not, R/S if you want it.
   34 AON
35 "PRINT RAW?"
   36 PROMPT
   37 AOFF
   38 ASTO X
   39 X=Y?
   4Ø SF ØØ
```

```
   41 "N"          Should singlet and doublet data be stored for
   42 ASTO Y         analysis?  Enter "N" if not, R/S if you
   43 AON            want it.
44 "ANALYZE?"
   45 PROMPT
   46 AOFF
   47 ASTO X
   48 X=Y?
   49 SF Ø1
   5Ø LBL Ø1       Prompts for a seed value for the PRNG.
   51 TONE 6
   52 "SEED?"
   53 PROMPT
   54 LN
   55 ABS
   56 STO Ø9
   57 LBL Ø5       Pseudo-Random Number Generator, PRNG routine.
   58 PI             Your routine should be entered here.  This
   59 RCL Ø9         one takes transformed seed from reg. Ø9,
   6Ø +              adds pi, raises sum to 5th power and
   61 5              stores fractional part in reg. Ø9.
   62 Y↑X            Lines 65-67 scale result to range of C
   63 FRC            and take integer.
   64 STO Ø9
   65 RCL Ø8
   66 *
   67 INT
   68 STO 16
   69 FS? ØØ       Test:  accumulate PRNG output for later
   7Ø GTO Ø4         analysis?
   71 LBL Ø3       Accumulate PRNG output in print buffer,
   72 ACX            with spaces, for later printing.
   73 1
   74 SKPCHR
   75 LBL Ø4       Increment trials counter (reg. Ø6)
   76 1              by one.
   77 ST+ Ø6
   78 FS? Ø1       Test: analysis wanted?
   79 GTO Ø6
   8Ø RCL 16       Add 5Ø to PRNG output number to get control
   81 5Ø             number so proper singlet count register
   82 +              will be incremented.  Increment by one.
   83 STO 17
   84 1
85 ST+ IND 17
   86 FS? Ø2       Test: doublet analysis OK?
   87 GTO 11
```

```
 88 RCL Ø6        Test: first trial of run?  If so, skip
 89 1               lines 91-96 to increment a doublet
 9Ø X=Y?            register.
 91 GTO Ø6
 92 RCL 18        Add 1Øx previous PRNG output to current
 93 RCL 16          PRNG ouput to determine doublet storage
 94 +               register number.
 95 STO 19
 96 1             Increment appropriate doublet register
 97 ST+ IND 19      by one.
 98 LBL Ø6        Multiply current PRNG output by 1Ø, store,
 99 RCL 16          use for doublet increment addressing on
 1ØØ 1Ø             next trial.
 1Ø1 *
 1Ø2  STO  18

 1Ø3 LBL 11       Test: end of run?  If not, activate another
 1Ø4 RCL Ø6         PRNG output.
 1Ø5 RCL Ø5
 1Ø6 X>Y?
 1Ø7 GTO Ø5
 1Ø8 PRBUF        Print accumulated PRNG output.  Add trials
 1Ø9 RCL Ø6         of current run to grand trials counter.
 11Ø ST+ Ø7
 111 RCL Ø7       Print grand total of PRNG trials to date.
 112 "ΣΣ = "
 113 ARCL X
 114 AVIEW
 115 Ø            Reset trials counter to Ø.
 116 STO Ø6
 117 GTO Ø1
118 LBL "CHI"                 CHI-SQUARE ANALYSES
 119 Ø
 120 STO 26       Clear Chi-square total registers.
 121 STO 35
 122 5Ø           Compute control number for indirect RCL of
 123 STO 27         singlet registers 5Ø through 5Ø+(C-1).
 124 RCL 46
 125 +
 126 1 E3         # = (5Ø+ [C-1] )  + .ØØØØ1
 127 /                      1,ØØØ
 128 1 E-5
 129 +             = 5Ø.ØØ(C-1)Ø1
 13Ø ST+ 27
 131 STO 28
 132 RCL Ø7       Computed expected singlet frequency, E,
 133 RCL Ø8         where
 134 /                      E=N/C
 135 STO 25
```

```
136 5           Test: E<5?  If so, SF Ø3.
137 X>Y?
138 SF Ø3
-------------------------------------------------
139 LBL Ø8      Accumulate frequencies of various singlets
14Ø RCL IND 27    for later printing.
141 ACX
142 1
143 SKPCHR
-------------------------------------------------
144 RCL IND 27  Compute singlet Chi-square
145 RCL 25
146 -             Chi-square = Σ (O-E)^2 / E
147 X↑2
148 RCL 25
149 /               O = observed frequency, each output
15Ø ST+ 26          E = expected frequency of each PRNG output
151 ISG 27
152 GTO Ø8
-------------------------------------------------
153 FIX 3
154 PRBUF       Print singlet Chi-square.
155 RCL 26
156 "S CHI SQ= "
157 ARCL X
158 AVIEW
159 PSE
16Ø CLA
161 FIX Ø
-------------------------------------------------
162 FS? Ø3      Test: E<5?  If so, print "E<5".
163 XEQ 13
-------------------------------------------------
164 CF Ø3
165 FS? Ø2      Test: doublet analysis OK?
166 GTO 12
-------------------------------------------------
167 RCL Ø7      Compute reduced N for doublet analysis
168 ENTER↑        (one trial lost on each run).
169 ENTER↑
17Ø RCL Ø5
171 /
172 -
-------------------------------------------------
173 RCL Ø8      Compute expected doublet frequency, E.
174 X↑2
175 /
176 STO 25
-------------------------------------------------
177 5           Test: E<5?  If so, SF Ø3.
178 X>Y?
179 SF Ø3
-------------------------------------------------
```

```
   18Ø RCL 46        Compute control number for indirect RCL
   181 1 E3            of doublet registers ØØ through Ø(C-1),
   182 /               1Ø through 1(C-1), etc.
   183 1 E-5
   184 +
   185 STO 27
   186 STO 29
   187 LBL Ø9        Nested loops for computing Chi-square
   188 LBL 1Ø          values and printing doublet frequency
   189 FIX Ø           table.  LBL 1Ø moves across a row of
19Ø RCL IND 27         the table, LBL Ø9 moves down to
   191 ACX             the next row.  Lines 2Ø4 to 2Ø9 change
   192 1               the loop control value to step to the next
   193 SKPCHR          row.
194 RCL IND 27
   195 RCL 25
   196 -
   197 X↑2
   198 RCL 25
   199 /
   2ØØ ST+ 35
   2Ø1 ISG 27
   2Ø2 GTO 1Ø
   2Ø3 PRBUF
   2Ø4 1Ø
   2Ø5 RCL Ø8
   2Ø6 -
   2Ø7 .Ø1
   2Ø8 +
   2Ø9 ST+ 27
   21Ø ISG 29
   211 GTO Ø9
   212 FIX 3         Print doublet Chi-square
   213 RCL 35
214 "D CHI SQ= "
   215 ARCL X
   216 AVIEW
   217 PSE
   218 CLA
   219 FIX Ø
   22Ø FS? Ø3        Test: E<5?  If so, print "E<5".
   221 XEQ 13
   222 CF Ø3         Go to seed prompt routine to generate
   223 GTO Ø1          another run.
   224 LBL 12        Print "No DBLT ANAL" if C>5.
225 "NO DBLT ANAL"
   226 AVIEW
   227 PSE
   228 GTO Ø1
```

```
229 LBL 13
23Ø SF 12
231 "E<5"
232 AVIEW
233 CF 12
234 RTN
235 END
```

Print "E<5" to warn that Chi-square analysis probably not valid.