An Interrupt Timing Simulation

V. D. Jones and R. L. Schwartz
DSN Data Systems Development Section

This article describes a timing simulator written in ANSI FORTRAN. The program was developed to aid in the location of timing anomalies in existing interrupt-driven software and to assist in the design of new real-time programs.

I. Introduction

The timing simulation is designed to simulate the real-time processing of internal and external interrupts of both an asynchronous and synchronous nature as well as a series of subprograms that are to be executed as a result of executing an interrupt processing routine.

In order to make use of the simulation, information must be known as to the exact structure and hierarchy of the processing system. The relative priority as well as the duration and frequency of the different routines must also be known.

The simulation handles virtually any system consisting of a supervisory loop with asynchronous and synchronous interrupts and a subprogram queue structure. When all information concerning the system has been given to the program, the simulation of the interrupt handling will occur. Various statistics and state snapshots are available.

A brief timing study of the effect of a 20-character per second Remote Output Teletype on the Viking Telemetry and Command Processor (TCP) using the XDS 920 computer is provided as an example.

II. Overview

The simulator accepts as input data mnemonics which represent Primary Interrupts (PINs), Clock Interrupts (CLKs), and subprograms in a priority queue (PSQ). Priorities, durations, and interval times (not for all inputs) are given (in number of cycles) and the handling of the interrupts is simulated.

At the beginning of each cycle, every clock is checked to see if its interval has expired.[1] If so, the clock is placed on a push down stack. The primary interrupts are then checked to see if there are any occurrences for that cycle, and, if so, the interrupt is put in the stack. The stack is then sorted according to the priority of each interrupt. At

[1]In actuality, a computation is done to determine the next cycle at which a change occurs, and all necessary counters are incremented to that value, eliminating the execution of needless cycles.

this point, the duration counter of the highest priority interrupt is updated. If this is the first cycle of the routine, then the elements in the PSQ associated with that interrupt are enabled. When the duration is satisfied, the interrupt is wiped off the top of the stack. If there are no interrupts to process during that cycle, then the highest priority subprogram that is enabled is fetched and put in the stack. When no PSQ entry has been enabled, the program loops in the supervisor (MAIN) waiting for an interrupt to occur.

Additional features include the ability for a PSQ element to queue other PSQ elements or to enable clock interrupts, and for a clock to have a controlling envelope.

A watchdog timer interval may be specified in order to monitor the activity of the program. This gives status of all interrupts and subprograms. Following the final time specified for the simulation, various statistics are given concerning the activity of the interrupts during the run.

III. Structural Elements of Simulation

A. Priority Subprogram Queue (PSQ)

The PSQ routines are those which are executed in the base or noninterrupt status of the machine. They exist in a priority queue and are enabled by clocks, primary interrupts, and other PSQ routines. Each PSQ can either be assigned a priority as with the clocks and primary interrupts, or will be assigned a default priority of zero, with its position in the PSQ table (relative to the top) determining its priority with respect to other PSQs. Each PSQ entry is disabled after execution. The "main loop" routine is a special case of the PSQ type. It is always enabled and has the lowest priority, thus it will only be executed when nothing else is active or pending.

B. Clock (CLK)

The clock routines are periodically enabled as by a cyclic interrupt. They are given priorities which determine their relative importance with respect to the other clocks, primary interrupts, and, in some cases, the PSQs. The higher the number, the higher the priority. Once started, a clock is enabled once each time its interval expires. This is subject to conditions specified below in regard to the envelope and clock flags. A clock may have a random or deterministic start time. For a random start time, a random number between 1 and 255 is assigned. The duration of the clock interrupt handling routine is also given. When a clock has been enabled and reaches the top of the stack, it is executed. On the first cycle of execution each CLK enables the PSQs associated with it.

C. Primary Interrupt (PIN)

The PIN routines are asynchronous interrupts whose enabling times are given in the input data. When the PIN occurs, it is enabled, and the first cycle of its operation is used to enable all PSQs associated with it. Since it is asynchronous, no envelope control is required. Each PIN has a priority and duration.

D. Envelope (ENV)

Envelopes or ENVs serve to inhibit or permit clock interrupts. If a clock interrupt is controlled by an envelope, an interrupt is only recognized when the envelope is high. A clock can be controlled by at most one envelope, but an envelope may control more than one clock. The waveform of the envelope is similar to that of the clocks. It is started at a time either specified as random or deterministic. Each ENV is given a duration and an interval. Thus, if an envelope is turned on at time t_0, it will be on until $t_0 + t_d$. It will be off from $t_0 + t_d$ until $t_0 + t_i$, then on from $t_0 + t_i$ until $t_0 + t_i + t_d$, etc.

E. Clock Control Flag (CCF)

Clock control flags or CCFs are controlled by the completion of PSQs. CCFs exist as a simulation feature to allow disabling and enabling of interrupts by subprograms. Clocks may be controlled by more than one PSQ, and a PSQ may control several clocks.

F. ENV–CCF Interaction

The envelopes (ENVs) and the clock control flags (CCFs) are examined, and, if "true," the appropriate clocks will be enabled. If a clock is disabled when it would normally start as the result of interval expiration, it will not start until the interval has expired (again).

IV. Modified Backus Normal Form (BNF) Description of Simulation Input Data

Input editing is structured to process the mnemonics and associated parameters in accordance with the syntax rules given below. The notation is to be interpreted as follows:

(1) Read the connective ":=" as: "is formed from"

(2) Read the symbol "<X>" as: "the object named X"

(3) $\left\{\cdot\right\}_a^b$ indicates repetition of the set $\left\{\cdot\right\}$ from a to b inclusive

(4) $\left\{\cdot\right\}_a^a$ indicates repetition of the set $\left\{\cdot\right\}$ a times

(5) "|" indicates exclusive or; viz. <X> : = $p\,|\,q\,|\,r$ shortens the sequence given by:

<X> : = p
<X> : = q
<X> : = r

(6) Any string not enclosed in angular brackets (<· · ·>) represent a terminal element, e.g., +, −, RND, ƀ (blank), \$, etc.

<DIGIT> : = 0|1|2|3|4|5|6|7|8|9

<CHAR> : = A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

<DELIMITER> : = ,|ƀ

<TERMINATOR> : = \$ <NOISE>

<NAME> : = <CHAR> $\left\{\text{<CHAR>}\,|\,\text{<DIGIT>}\right\}_0^5$ <DELIMITER>

<NUMBER> : = $\left\{\text{<DIGIT>}\right\}_1^8$ <DELIMITER> (UNLESS NOTED ELSEWHERE)

<NOISE> : = <NAME> | $\left\{\text{<NAME> <DELIMITER>}\right\}_0^\infty$

<NUMERIC DATUM> : = <NOISE> <NUMBER> <NOISE>

<START TIME> : = <NUMBER>|RND <NUMBER ($\leq$255)>|RND

<FLAG> : = +|−

<TFIN> : = <NUMERIC DATUM>

<PSQ> : = <NAME> <NUMBER (DURATION)> <NUMBER (PRIORITY)> $\left\{\text{<NUMBER (NUMBER OF PSQs)>}\left\{\text{ <NAME (PSQ NAME) > }\right\}_{\text{NUMBER PSQs}}^{\text{NUMBER PSQs}}\right\}_0^1$

NOTE: A PSQ may call only those PSQs already input.

<MAIN LOOP> : = <NUMBERIC DATUM>

<WATCHDOG> : = <NUMERIC DATUM>

<CLOCK> : = <NAME> <NUMBER (PRIORITY)> <NUMBER (INTERVAL)> <NUMBER (DURATION)> <NUMBER (NUMBER PSQs ENABLED)> <START TIME> $\left\{\text{<NAME (PSQ NAME)>}\right\}_{\text{NUMBER PSQs}}^{\text{NUMBER PSQs}}$ $\left\{+\ \text{<NUMBER (NUMBER PSQs ENABLED)>}\left\{\text{<NAME (PSQ NAME)>}\right\}_0^{\text{NUMBER PSQs}}\right\}_0^n$

NOTE: "NUMBER OF PSQs ENABLED" reflects the number on that card. Continuation lines contain a separate count. $0 \leq n =$ total number PSQs enabled ≤ 21.

<ENVELOPE> : = <NAME> <NUMBER (INTERVAL)> <NUMBER (DURATION)> <NUMBER (NUMBER CLOCKS CONTROLLED)> <START TIME>

$$\left\{ \langle\text{NAME (CLOCK NAME)}\rangle \right\}_{\text{NUMBER CLOCKS CONTROLLED}}^{\text{NUMBER CLOCKS CONTROLLED}}$$

NOTE: $0 \leqq$ NUMBER CLOCKS CONTROLLED $\leqq 25$.

<ISR> : = <NAME> <NUMBER (PRIORITY)> <NUMBER (DURATION)> <NUMBER (NUMBER OF PSQs CALLED)>

$$\left\{ \langle\text{NAME (PSQ NAME)}\rangle \right\}_{0}^{\text{NUMBER OF PSQs CALLED}}$$

<PINTIME> : = $\left\{ \langle\text{NUMERIC DATUM}\rangle \right\}_{0}^{25}$

NOTE: Interrupt times must be in ascending order.

<CLOCK FLAG> : = <NAME (PSQ NAME)> <NUMBER (NUMBER OF CLOCKS CONTROLLED)>

$$\left\{ \langle\text{NAME (CLOCK NAME)}\rangle \ \langle\text{FLAG}\rangle \right\}_{\text{NUMBER CLOCKS CONTROLLED}}^{\text{NUMBER CLOCKS CONTROLLED}}$$

<DATA SETUP> : = $\langle\text{TFIN}\rangle \ \langle\text{TERMINATOR}\rangle \left\{ \langle\text{PSQ}\rangle \right\}_{0}^{30} \langle\text{TERMINATOR}\rangle$

$$\langle\text{MAIN LOOP}\rangle \ \langle\text{WATCHDOG}\rangle \left\{ \langle\text{CLOCK}\rangle \right\}_{0}^{25} \langle\text{TERMINATOR}\rangle$$

$$\left\{ \langle\text{ENVELOPE}\rangle \right\}_{0}^{25} \langle\text{TERMINATOR}\rangle \left\{ \langle\text{ISR}\rangle \ \langle\text{PINTIME}\rangle \right\}_{0}^{25}$$

$$\langle\text{TERMINATOR}\rangle \left\{ \langle\text{CLOCK FLAG}\rangle \right\}_{0}^{25} \langle\text{TERMINATOR}\rangle$$

NOTE: There are no column restrictions. Columns 1 through 80 are scanned on input. Any card with a quote sign (') in column 1 will be interpreted as a comment card.

V. Viking Telemetry and Command Processor Timing Simulation

After determining the program structure, as well as the execution time and frequency (where applicable) of the routines, the input data (Fig. 1) reflect the simulation input with worst case timing estimates. Figure 2 illustrates a snapshot of the simulated system following expiration of total process time.

These results apply to the Viking Model with the following current operations:

(1) 20 characters per second teletype outputting continuous alarms

(2) 33⅓-bit per second Channel 1 telemetry.

(3) 2000-bit per second Channel 2 telemetry.

(4) 16000-bit per second Channel 3 telemetry.

(5) 4-bit per second command rate.

(6) 11-word Mission Definition Table transfer from TCP to DDA (Data Decoder Assembly) every second.

The above system was run for 4 seconds (500,000 cycles) with telemetry acquisition lasting 1.26 seconds (157,584 cycles) and tracking for 3.74 seconds (342,416 cycles).

The worst case analysis yielded the following: During Channel 1 acquisition an overrun of the module status subprogram (MSTAT) queued by the 10-pulse per second interrupt routine occurred 67% of the time. This was due to both the increased execution time of the

Channel 1 trap (C1TRPA) which takes roughly twice as long during acquisition and to the processing of GCF blocks and the message processor done by the GCFBLK and MSGPRC subprograms. Despite the MSTAT overrun, status was still completed approximately every 315 milliseconds.

During tracking the overrun of MSTAT was less severe, with an overrun only occurring 25% of the time yielding a completed module status update approximately every 130 milliseconds.

An overrun of MSTAT was not caused to any large extent by the addition of the 20-cps teletype but was caused by the increased activity of the Channel 1 trap during acquisition and the long execution time for MSGPRC and GCFBLK (approximately 120 milliseconds).

If the module status is desired more frequently than every 315 milliseconds during acquisition and every 130 milliseconds during tracking, it is suggested that the MSGPRC and GCFBLK subprograms be segmented into four smaller blocks to allow the executive to check for enabled subprograms of higher priority between segment execution.

VI. Program Usage

The program (written in ANSI FORTRAN) takes approximately 14K of memory (7K for instruction bank, 7K for data bank). See the Timing Simulation Descriptive Document (Ref. 1) for a complete description including operating instructions for the UNIVAC 1108 and the XEROX SIGMA 5, and a description of how to construct a simulation given an interrupt processing system.

Reference

1. Schwartz, R., *An Interrupt Timing Simulation*, IOM 3384-74-013, Mar. 8, 1974 (JPL internal document).

Bibliography

Jones, V. D., "Telemetry and Command Multiple-Mission Software (Model A)," in *The Deep Space Network Progress Report*, Technical Report 32-1526, Vol. XVIII, pp. 163–166, Jet Propulsion Laboratory, Pasadena, Calif., Dec. 15, 1973.

Schwartz, R., *Viking Remote Output Teletype Timing Analysis*, IOM 3384-74-012, Mar. 1, 1974 (JPL internal document).

```
'INPUT DATA FOR VIKING TELEMETRY AND COMMAND SIMULATION
'-------------------------------------------------------
'
'
TFIN 500000
$
'
'
'PSQ   DURATION
'-------------
C2FRM  438
C1FRM  538
S1STAT 20
B1STAT 20
S2STAT 20
TMAGC  125
MSTAT  500
SSCON  500
SSCON1 663
DSGEN  250
DDCON  250
DDCON1 413
SNCAL  163
LOMSG  50
BDCON  125
BDCON1 288
C1CON  62
GCFBLK 16625
GCF2   14375
MSGPRC 13413
TYPEIN 15375
TYPMEM 50
TVER   250
$
'
'
MAIN   3125
WTCHDG 500000
'CLKS  PRIO   INT  DUR  #  STRT        PSQS ENABLED
'-------------------------------------------------
PPS1   10    125000 125  6   5          TMAGC SSCON DDCON BDCON C1CON MSGPRC
BITSRT 18    31250  725  0   16384
SYMIR  17    31250  1438 0   20290
CMAEOW 16    750000 13   0   10
'      THIS IS FOR 2000 BPS
I2DDA1 11    58500  125  0   4754
I2DDA2 12    14624  125  0   5676
I1DDA1 13    58500  1200 1   9081       C2FRM
'      CHANNEL 3 16000 BPS HIGH RATE
I1DDA2 14    11364  125  0   12382
PPS10  15    12500  375  1   5          MSTAT
C1TRPA 22    938    425  0   1
C1TRPT 22    938    250  0   1
C1TRP1 22    157658 10   1   204500     C1FRM
SEC30  10 3750000 10    4 3750005       SSCON1 DDCON1 BDCON1 DSGEN
                             + 3        S1STAT S2STAT B1STAT
HSII   25    625    50   0   256

HSIIG  25    625000 10   1   31526      GCFBLK
HSOI   24    625    50   0   1
'      ROINT 20  CPS
ROINT  11    6250   100  0   5
$
'
'
'ENV   INT            DUR     #    STRT          CLOCKS CONTROLLED
'-----------------------------------------------------------------
ENV1   3825100        3625100  1      5            PPS1
ENV3   500000         157584   1      1            C1TRPA
ENV4   500000         500000   2    157600        C1TRPT HSOI
$
'
'
'PIN   PRIO DUR    #    PSQS ENABLED
'----------------------------------
EORS   26   20     3    TYPEIN TYPMEM TVER
13767 4283230
CSU    27   100    1    GCF2
29767 6404000
$
$
'
'
'EOD
```

Fig. 1. Sample input data for Viking telemetry and command simulation

```
          ******END OF CYCLE   500000******


          STACK
     PRIORITY      NAME        STACK TIME
                           DELTA      TOTAL
       22          C1TRPT    205    85433
       -1          MAIN     3107    88540


          CLOCK
 PRIOR      INTV       TIME  E/D     DUR.        T   START  COUNT     ENV NO.     NAME
  10      125000     124995   0       125        0       5      4           1     PPS1        1.
  18       31250      14866   0       725        0   16384     16           0     BITSRT      2.
  17       31250      10960   0      1438        0   20290     16           0     SYMIR       3.
  16      750000     499990   0        13        0      10      1           0     CMAEOW      4.
  11       58500      27246   0       125        0    4754      9           0     I2DDA1      5.
  12       14624      11732   0       125        0    5676     34           0     I2DDA2      6.
  13       58500      22919   0      1200        0    9081      9           0     I1DDA1      7.
  14       11364      10330   0       125        0   12382     43           0     I1DDA2      8.
  15       12500      12495   0       375        0       5     40           0     PPS10       9.
  22         938         45   0       425        0       1    169           2     C1TRPA     10.
  22         938         45  -1       250       45       1    365           3     C1TRPT     11.
  22      157658     137842   0        10        0  204500      2           0     C1TRP1     12.
  10     3750000          0   0        10        0 3750005      0           0     SEC30      13.
  25         625        369   0        50        0     256    800           0     HSII       14.
  25      625000     468474   0        10        0   31526      1           0     HSIIG      15.
  24         625        624   0        50        0       1    547           3     HSOI       16.
  11        6250       6245   0       100        0       5     80           0     ROINT      17.


          ENVELOPE DATA
          ENV     INTERVAL       TIME       DUR        T          START     ENV E/D
 1.     ENV1       3825100     499994   3625100   499994             5      -1
 2.     ENV3        500000     499998    157584        0             1       0
 3.     ENV4        500000     342399    500000   342399        157600      -1


        CLOCK FLAGS
 CLK   1.  2.  3.  4.  5.  6.  7.  8.  9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25.

 FLAG   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 STATE  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0


          ISR
 PRIOR    COUNT        DUR.        T  NAME
  26         1          20         0  EORS
  27         1         100         0  CSU


          PSQ
 PRIOR  E/D COUNT    DUR.      T
  0      0     9     438      0  C2FRM       1.
  0      0     2     538      0  C1FRM       2.
  0      0     0      20      0  S1STAT      3.
  0      0     0      20      0  B1STAT      4.
  0      0     0      20      0  S2STAT      5.
  0      0     4     125      0  TMAGC       6.
  0      0    25     500      0  MSTAT       7.
  0      0     4     500      0  SSCON       8.
  0      0     0     663      0  SSCON1      9.
  0      0     0     250      0  DSGEN      10.
  0      0     4     250      0  DDCON      11.
  0      0     0     413      0  DDCON1     12.
  0      0     0     163      0  SNCAL      13.
  0      0     0      50      0  LOMSG      14.
  0      0     4     125      0  BDCON      15.
  0      0     0     288      0  BDCON1     16.
  0      0     4      62      0  C1CON      17.
  0      0     1   16625      0  GCFBLK     18.
  0      0     1   14375      0  GCF2       19.
  0      0     4   13413      0  MSGPRC     20.
  0      0     1   15375      0  TYPEIN     21.
  0      0     1      50      0  TYPMEM     22.
  0      0     1     250      0  TVER       23.
 -1      0    22    3125     18  MAIN       24.


 ******TFINAL SIGNALED******
 SUBPROGRAM MSTAT  OVERUN  37.50 % OF TIME


 CLOCK STATISTICS
   NAME      MEAN TIME BEFORE  COMPLETION

  PPS1                    2111.
  BITSRT                  1375.
  SYMIR                   2689.
  CMAEOW                   481.
  I2DDA1                   441.
  I2DDA2                   613.
  I1DDA1                  3217.
  I1DDA2                   506.
  PPS10                    693.
  C1TRPA                   451.
  C1TRPT                   300.
  C1TRP1                   207.
  HSII                      50.
  HSIIG                     41.
  HSOI                      50.
  ROINT                    781.
```

Fig. 2. Sample output of Viking telemetry and command model