527-106

SAX science telemetry allocation

A simulation tool

Prepared by:

L.Chiappetti, IFCTR

Issue 1.0
30 April 1990

Distribution:

ASI	R.C.Butler
SAX EMWG	(TBD for issue 1.0)
AER	SSDCWG members
SSDCWG	all members

Document for which no computer readable source exists any longer

Partial scan of original hardcopy. (available on request) supplied

1. Introduction

The simulation program herein described has been prepared on request of R.C.Butler after the SAX Payload Instrument Data Management Meeting held in Milano on 6 October 1989.

The program itself is an evolution of a previous version (described in an informal note "Considerations about SAX Data Management" by L.Chiappetti, December 1988), but with a quite different aim.

The present program is intended to simulate and test different polling algorithm which could be employed by the SAX OBDH to read source packets produced by the SAX Instruments over the BTB.

Document history

Version	Date issued	Comments and pages affected by change
0.1 draft	3 Nov 89	original issue (all pages)
0.2 draft	22 Feb 90	reissued cover, section 1,2 and 3 entirely, page 4.2 and following, section 7 and added printouts for cases after 1c
0.3 draft	4 Apr 90	reissued cover, this pag. 1-1, pag. 4-3 and following, added pag. 7-7 and printouts for cases after 4c
1.0	30 Apr 90	reissued cover, this pag. 1-1, pag. 4-6 and following, pag. 7-13 and following, and added printouts after case 9d

Please note the version of the program used at the level of draft 0.3 OR LATER has undergone some changes undocumented in section 5 (namely additional info has been added to TAP.SX, described in 4.2.n, the SNAPSHOT feature is now present in debug mode, the details of which, as well as the presence and layout of plot files is now subject to undocumented short notice changes).

2. Logics of the simulation

The simulation program is based on the following logics:

1) It supports an arbitrary *number of experiments*, with a current maximum of 6 (which is also the number of the experiments in the full payload complement on SAX)

2) Each experiment *Field-Of-View* (*FOV*) can be defined as NFI, WFC1 or WFC2 (normally one has of course four NFIs, one WFC1 and one WFC2). Each type of FOV corresponds to an input light curve (namely it is associated to a *light curve file name*).

3) The input light curves are contained in external *light curve files*, which are generated under control of external programs not part of this package. The light curves define the trend of variation of the input (cosmic source) event rate, but not the absolute magnitude of the count rates. In line of principle all experiments defined as NFI share the same light curve file (i.e. the same qualitative trend), while WFC1 and WFC2 may be associated to the same or other files.

4) For each experiment one defines a constant *background level*, and a *scaling factor* which are used to convert the input light curves into true count rates. If *ai* is the content of the i-th bin of the input light curve, the corresponding count rate in cts/s is :

$$ci = B + S\ ai$$

where B is the background and S the scaling factor. Therefore the intensity of the source to be simulated and its spectral characteristics as seen by a given instrument are parametrized only through S (0.1 Crab = 50 S circa).

4a) The light curve information is relevant only for direct mode simulation. Note that light curves can be modelled on coarse time bins (larger than the polling simulation bins). Earth obstruction periods (for all modes) are considered in a coarse way (note that only one WFC could be assumed active at any one time).

5) For each experiment one defines a number of logical *queues* of source packets. At the moment all source packets are assumed to have the same length of 8808 bits. Also each queue (with the exception of the GRBM queue) can contain only 0,1 or 2 packets maximum (but WFCs in Direct mode use 20). Each queue has an associated *mode*.

5a) A *direct mode* (D) queue is characterized only by the *number of bits allocated to one event* (see however 6 below).

5b) An *indirect mode* (I) queue is characterized by two quantities: the *number of bytes allocated to an histogram* (spectrum) and the histogram *accumulation time* (see however 6 below).

3. Assumptions and limitations

3.1 Assumptions built in the program

A number of values are encoded in the program (as PARAMETER statements, which means they could be changed recompiling the program): some of them are truly relevant and shall be verified while others simply introduce practical limitations.

Relevant parameters include :

1) duration of one orbit, currently approximated as 100 minutes

2) length of a source packet (excluding the packet telemetry header, but including any user adminstrative data header) : 8808 bits

3) number of source packets generated at a GRBM trigger : 109 packets

4) number of toggle buffer per queue : 2 buffers (20 for WFC Direct)

5) mass memory size available to net science data : 374 Mbit (where 1 Mbit = 106 bits)

6) limiting bit rate (overall) : 100 kbit/s (1 kbit = 1000 bits)

7) limiting bit rate per each instrument : one sixth of the above figure (i.e. 16.7 kbit/s); not relevant, issue warning only.

Program built-in limits include :

8) maximum number of data points in input light curves : such to cover 2 orbits at 1 second resolution. However it is noted that a coarser resolution (e.g. 60 s) is generally sufficient.

9) maximum number of queues for all experiments together : 30 queues

3.2 Assumptions under user control

A number of other assumptions regarding the instrument behaviour are under user control, as they can be entered in the parameter files. It is expected that such values will be supplied by the Experiment Manager Working Group or by the Payload Manager.

A particular case is represented by the *quantum* time, which is nominally a free parameter, however its use is conditioned by the particular polling algorithm used. The following assumptions are made:

the time necessary to transmit a packet over the BTB is 17.7 ms (therefore the polling time is not meaningful if it is larger than this)

the time necessary for an IB interrogation is 10 cycles of 64 μs (i.e. 0.64 ms), hence this is the minimum sensible polling time

4. Algorithms

4.1 Queue-filling algorithms

The following mechanisms are used in the simulation program to fill the different type of queues :

1) for D-queues

from the number of bits/event e and the header size of h bytes, the number of events/packet is computed once for ever :

$$n = (8808-8h)/e$$

then during each quantum-cycle at time τ_j the event rate is derived from the input light curve as c_i taking into account the coarser binning :

$$i : \quad t_i < \tau_j < t_{i+1}$$

the event number in the current quantum is $c_i \Delta\tau$ (computed as a REAL number), which is added to a counter. When this counter exceeds n, the queue is incremented by 1 packet and the counter is reset.

2) for I- and S-queues

from the spectrum size of s bytes and the header size of h bytes, the number of spectra in a packet is computed once for ever :

$$N = (8808-8h)/8s$$

then during each quantum-cycle a time counter is incremented by $\Delta\tau$; when the time counter (REAL) exceeds the accumulation time for 1 spectrum, a spectrum counter is incremented, and the time counter is reset. When the spectrum counter exceeds N, the queue is incremented by 1 packet and the spectrum counter is reset.

The above if the spectrum size is less than the packet size (the remaining space is wasted). If the spectrum size is bigger than the packet size, it may require P packets (again with some wastage at the end). In this case the spectrum counter is unused, and when the time counter exceeds the accumulation time, the queue is incremented directly by P packets.

3) for H-queues

from the bit rate b and the header size of h bytes, the time taken to fill one packet is computed :

$$T = (8808-8h)/b$$

then during each quantum-cycle a time counter is incremented by $\Delta\tau$; when such time counter (REAL) exceeds T, the queue is incremented by 1 packet and the time counter is reset.

5. Usage of the program

The program is currently implemented on the IFCTR HP-1000 under RTE-6 VM. It is written in HP FTN7X Fortran, including in the i/o statements some (control) escape sequences for HP-terminals and for the HP Laserjet printer.

The program is controlled by a number of parameter files. The main (command) file is called *TAP.SX::X2* and should be filled according to the following template:

```
Nominal all-direct POLL1 scheme          * Run identification 40 chars
0.0175                                   * time quantum in seconds
6                                        * Number of experiments (1-6)
LCU.NF                                   * NFI  light curve file
LCU.W1                                   * WFC1 light curve file
LCU.W1                                   * WFC2 light curve file
-1,-1                                    * NFI  start end Earth obstruction MIN
-1,-1                                    * WFC1 start end Earth obstruction MIN
-1,-1                                    * WFC2 start end Earth obstruction MIN
MEC.SX                                   * experiment file name
LEC.SX                                   * experiment file name
HPG.SX                                   * experiment file name
PDS.SX                                   * experiment file name
WF1.SX                                   * experiment file name
WF2.SX                                   * experiment file name
---------------------------------------- * ---------------------------------
Asterisk shall be in column 41           * lines below --- are never read
```

note that it is compulsory to specify a run identification, the quantum time $\Delta\tau$, the number of experiment (maximum 6) E, the names of the three light curve files (which may also be all equal), the start and end of the obstructed period in each orbit (in minutes, e.g. 30,40 if the source is occulted from minute 30 to minute 40, or 80,10 if occulted from minute 80 to minute 10 of next orbit; if no obstruction desired code **-1,-1**).

only the next E lines following the obstruction codes are then read, any remaining line is ignored.

The experiment parameter files, named in the E final lines in the main command file, are read in turn. They are arranged according to the following template :

```
Medium Energy Concentrator               * Exp identification 40 chars
NFI                                      * FOV : NFI or WFC1 or WFC2
1.0                                      * light curve scaling factor
50.0                                     * background (cts/s)
4                                        * number of queues
                space for queue related info described below

---------------------------------------- * ---------------------------------
Asterisk shall be in column 41           * lines below --- are never read
```