LIGO Laboratory / LIGO Scientific Collaboration

LIGO- T1500406 *LIGO* August 12th, 2015

SEI WD Saturation Counter Update

Version 3

Hugo Paris, Brian Lantz

Distribution of this document:)
Advanced LIGO Project

This is an internal working note
of the LIGO Laboratory

Table of Contents

Update Summary

Changes listed below apply to the HAM-ISI, BSC-ISI and HEPI master models as well as the related MEDM Watchdog screens. Two changes are suggested:

1) Add a "clear saturations" button which is separate from the "reset Watchdog" button. The "Reset Watchdog" button will be unchanged, and the new "clear saturation" button will just clear the saturation counters.

2) The watchdog is setup so that if the total saturations in the history are greater than the allowed number of saturations then the watchdog will trip. Currently, the history is kept until the watchdog trips or the history is reset. We propose changing the saturation history buffering so that saturations more than 1 hour old (value defined in the simulink model) are automatically cleared. The history will be stored in 60 bins, each 1 minute long. Bins more that 1 hour old are cleared.

Reason for Change(s):
Change #1 is so that we can separate the “reset watchdog” from the “clear old saturations” function.
Change #2 is so we only trip on recent troubles, not a slow accumulation of old trouble. JimW

08/11 Bug fix (SEI alog #813)
The bug:
the Saturations_total = current_buffer + sum_of_history

The current_buffer is updated in real-time and contains the total number of saturations since the current 60 second block started.

history_array is 60 element array which contains the history. At the end of each 60 sec block, the current_buffer is put into 1 element of the history_array. It overwrites the "oldest" element of that array, which is how we drop old saturations.
- AND -
The history array is summed to find the new sum_of_history.
i.e. the sum_of_history is only updated 1 time every 60 seconds.

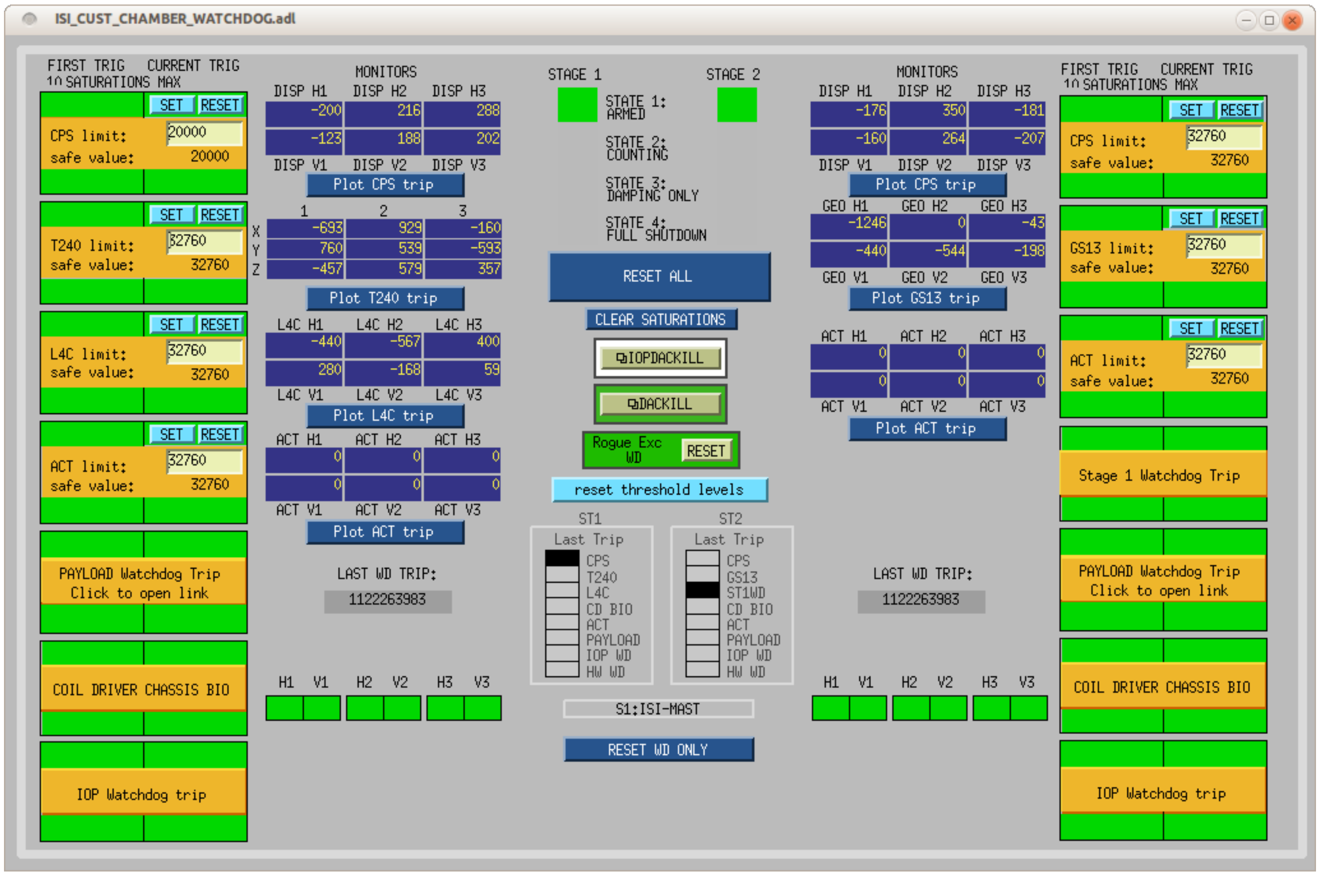
THE BUG -
When you hit reset, it set each element of the history_array = 0 and the current_buffer = 0. BUT NOT the sum_of_history.
THE FIX -
is to also set sum_of_history = 0.

this bug is annoying but not really dangerous. after 60 seconds, the sum_of_history gets recalculated, so the new sum_of_history will be 0 + whatever new saturations arrived between pressing the CLEAR and the end of the 60 second block.

I) Clear Saturations Button

a) Description

A “clear saturations” button was added the the WD screen. This button allows clearing the number of saturations registered on all our devices (CPS, GS13, ACT etc...) without having to reset the WD themselves. Model and MEDM changes are shown below, along with their description.



```
Stage 1 (left)
CPS limit: 20000     safe value: 20000
T240 limit: 32760    safe value: 32760
L4C limit: 32760     safe value: 32760
ACT limit: 32760     safe value: 32760

         DISP H1  DISP H2  DISP H3
           -200      216      288
           -123      188      202
         DISP V1  DISP V2  DISP V3

             1        2        3
X          -693      929     -160
Y           760      539     -593
Z          -457      579      357

         L4C H1   L4C H2   L4C H3
           -440     -567      400
            280     -168       59
         L4C V1   L4C V2   L4C V3

         ACT H1   ACT H2   ACT H3
              0        0        0
              0        0        0
         ACT V1   ACT V2   ACT V3

LAST WD TRIP: 1122263983

Stage 2 (right)
CPS limit: 32760     safe value: 32760
GS13 limit: 32760    safe value: 32760
ACT limit: 32760     safe value: 32760

         DISP H1  DISP H2  DISP H3
           -176      350     -181
           -160      264     -207
         DISP V1  DISP V2  DISP V3

         GEO H1   GEO H2   GEO H3
          -1246        0      -43
           -440     -544     -198
         GEO V1   GEO V2   GEO V3

         ACT H1   ACT H2   ACT H3
              0        0        0
              0        0        0
         ACT V1   ACT V2   ACT V3

LAST WD TRIP: 1122263983
```

Fig.1. BSC-ISI WD screen after update

Fig.1 shows the BSC-ISI WD screen after the update. The “Clear Saturations” button can be found in the center of that screen, right under the “Reset All” button (regular WD reset).

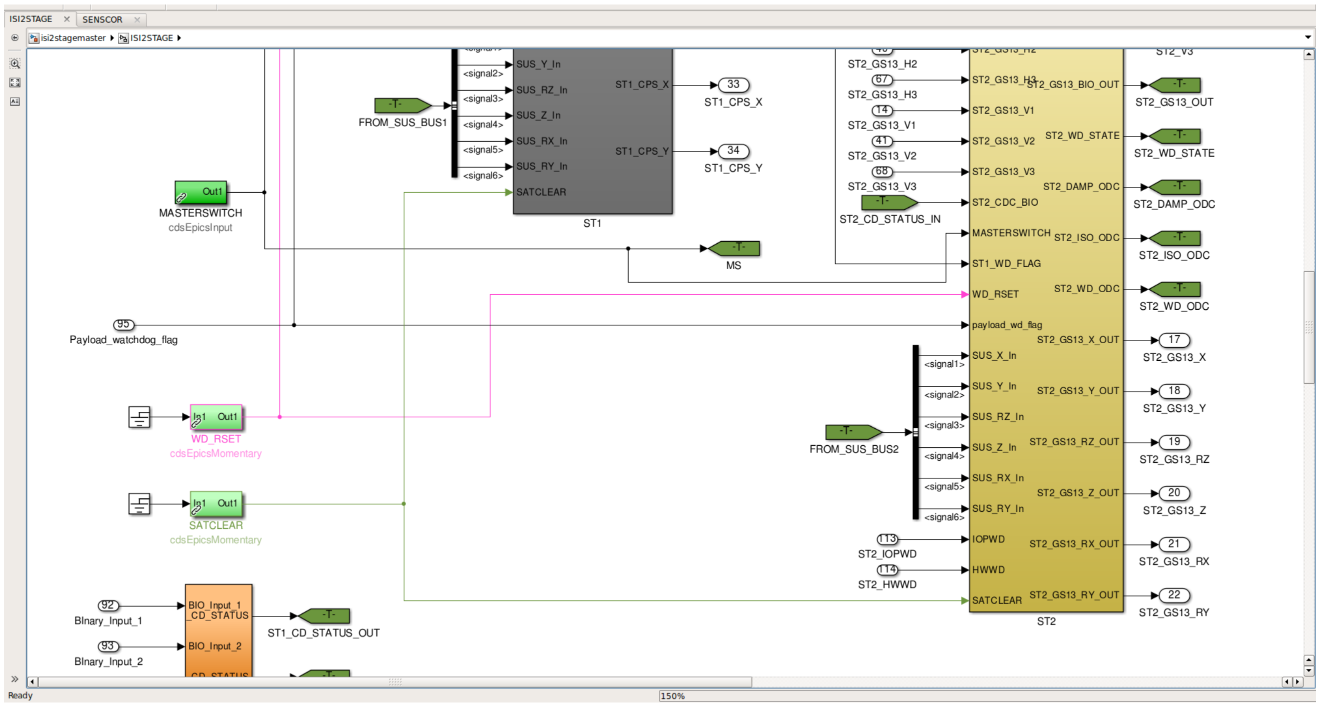


Fig. 2. SATCLEAR in top layer of model

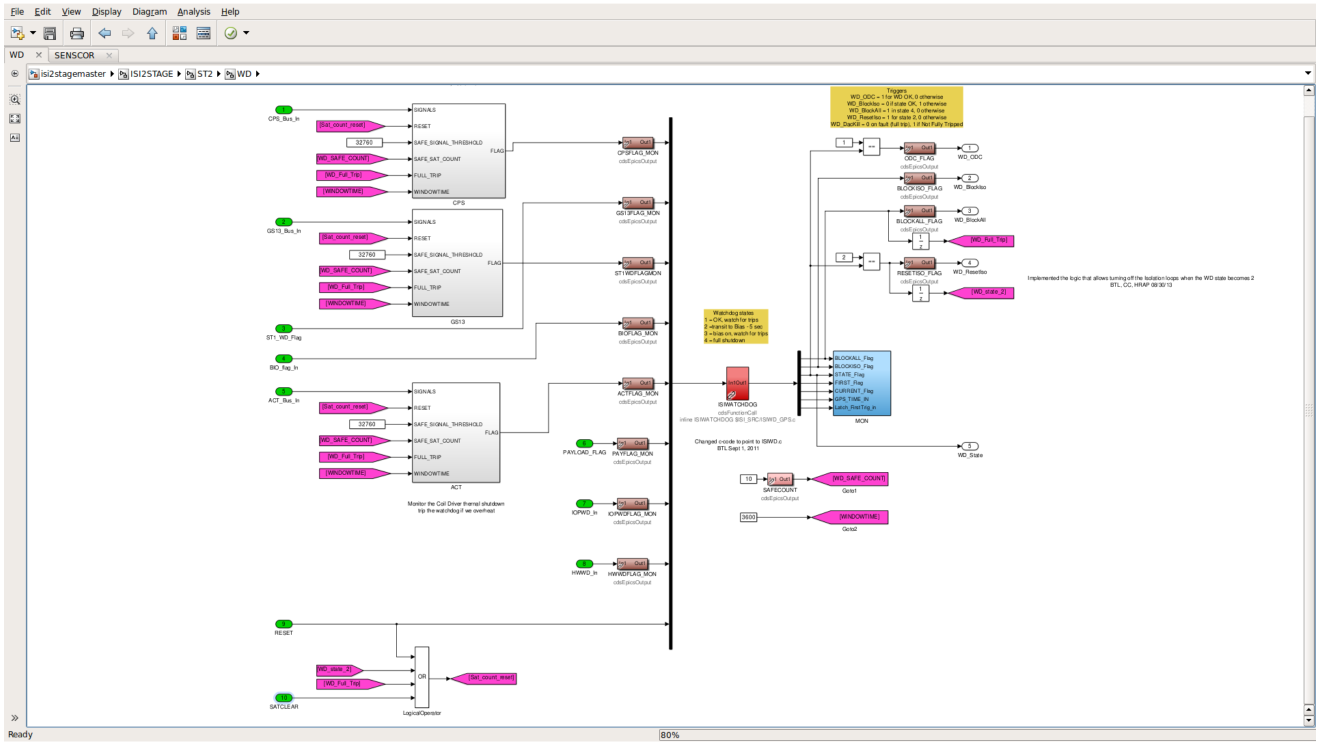


Fig. 3. SATCLEAR signal in the WD block

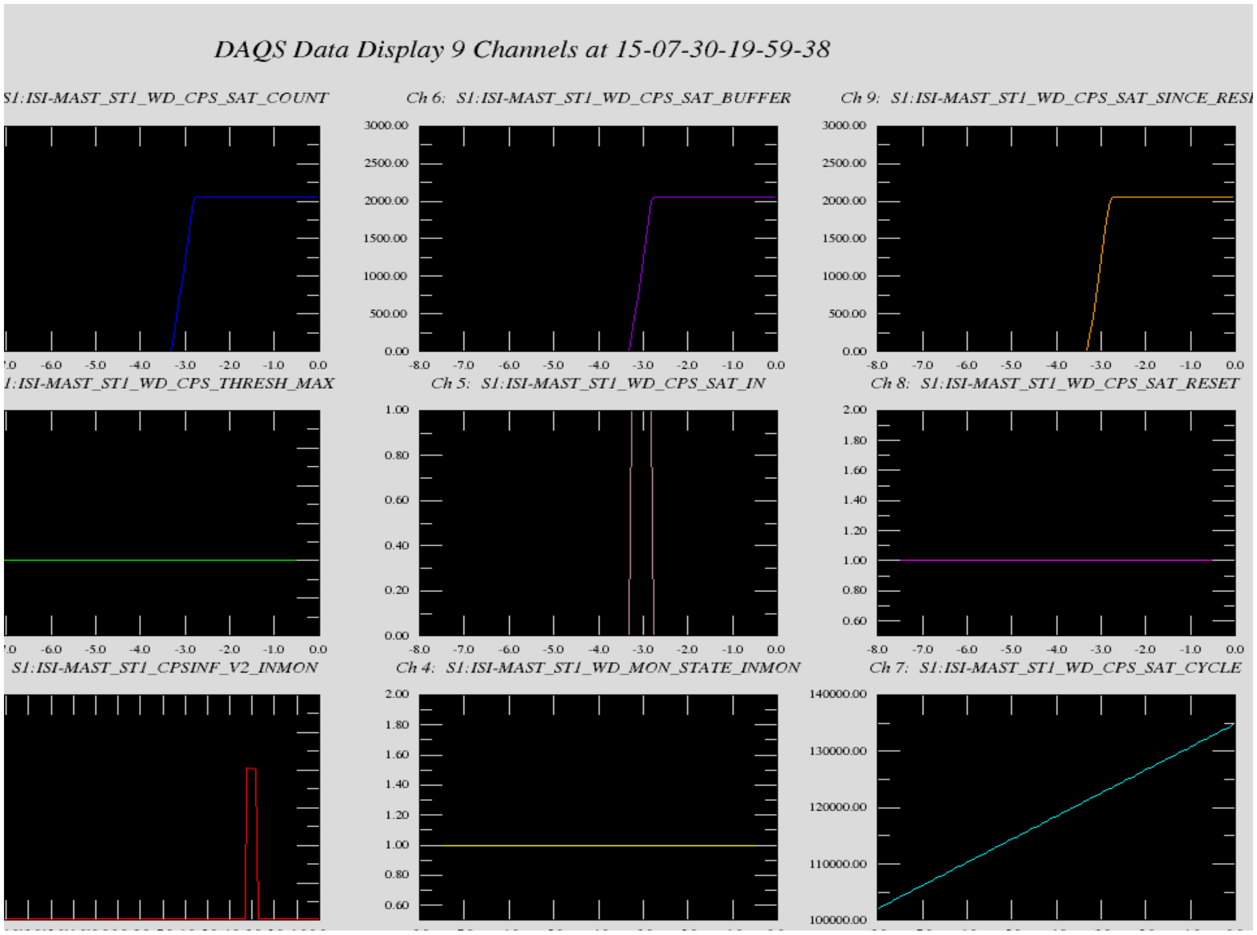
Fig.2-3 show the model changes necessary for the “ Clear Saturations” button to work. Fig.2. Shows the top layer of the BSC-ISI model to which a SATCLEAR EPICS momentary was added. This momentary is the one triggered by the “Clear Saturations” button of the WD screen. Its signal is then sent all the way down the model to the WD block where is is linked to the RESET input of the new WD saturation counter c-code (fig.3).

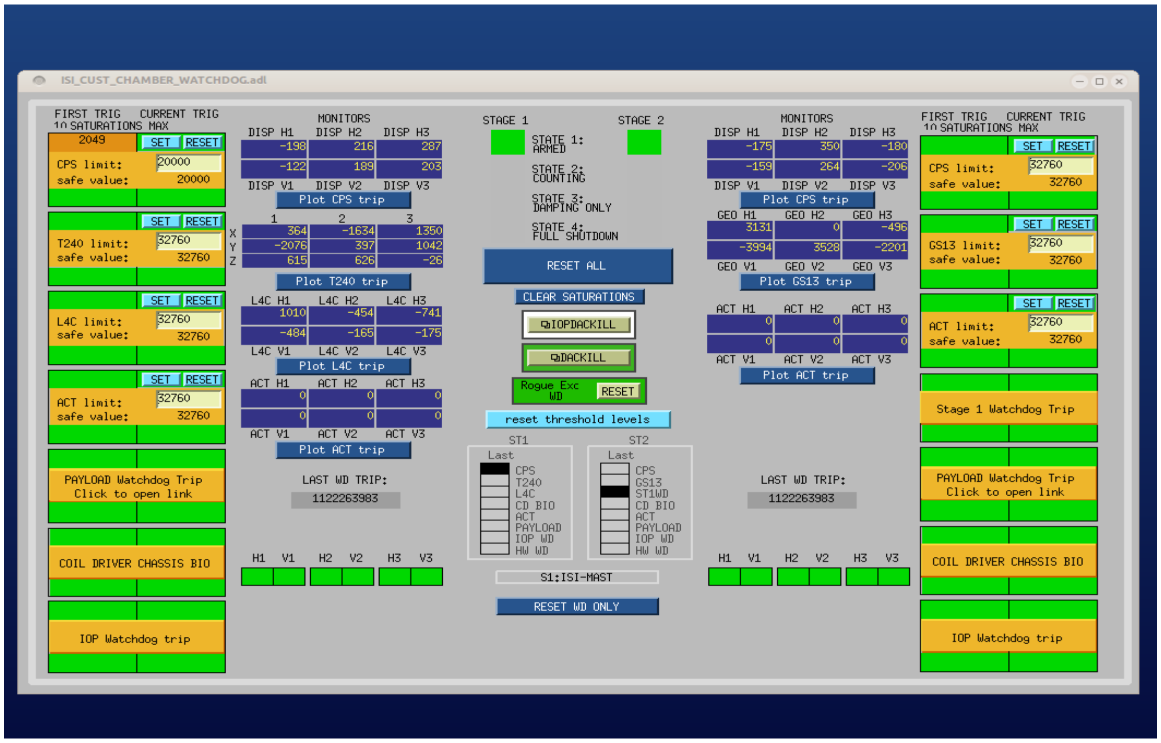
b) Tests Performed

The tests performed to ensure the “Clear Saturations” feature was properly working are listed below with the test result. The tests performed are illustrated in the figures below.

- Apply a number of saturation lower than the saturation counter limit (which was temporarily raised to 3000cts to facilitate the test).
- Make sure the number of saturations rises properly by monitoring them on DV and make sure that this information is properly displayed on the WD screen. (fig. 4-5)
- Use the “Clear Saturations” button to clear those saturations
- Confirm that the saturation counters got cleared by looking at both the WD MEDM screen and a DV session. (fig 6-7)

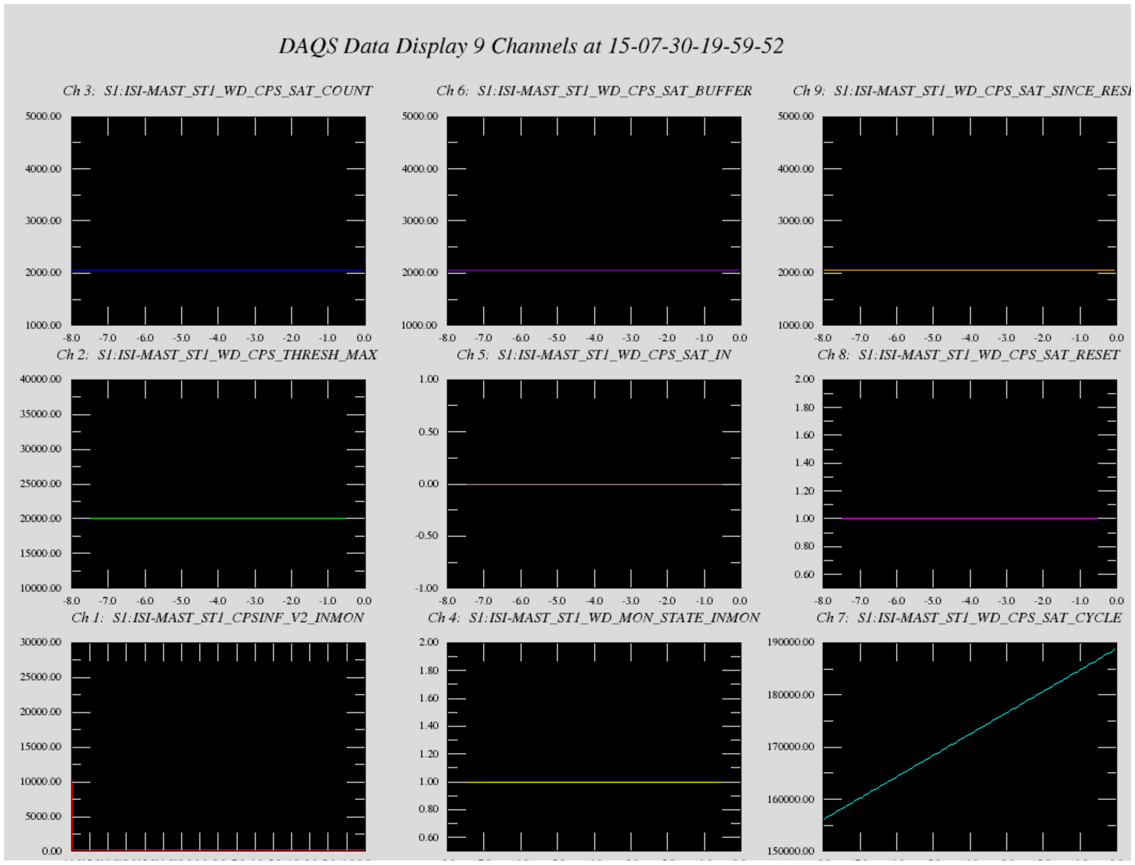
Test Result: Passed.




ISI_CUST_CHAMBER_WATCHDOG.adl

```
FIRST TRIG   CURRENT TRIG
10 SATURATIONS MAX
2049            SET RESET
CPS limit:      20000
safe value:     20000
                SET RESET
T240 limit:     32760
safe value:     32760
                SET RESET
L4C limit:      32760
safe value:     32760
                SET RESET
ACT limit:      32760
safe value:     32760
PAYLOAD Watchdog Trip
 Click to open link
COIL DRIVER CHASSIS BIO
IOP Watchdog trip

          MONITORS
  DISP H1   DISP H2   DISP H3
     -198       216       287
     -122       189       203
  DISP V1   DISP V2   DISP V3
        Plot CPS trip
      1         2         3
X     364     -1634      1350
Y   -2076       397      1042
Z     615       626       -26
        Plot T240 trip
  L4C H1    L4C H2    L4C H3
    1010      -454      -741
    -484      -165      -175
  L4C V1    L4C V2    L4C V3
        Plot L4C trip
  ACT H1    ACT H2    ACT H3
       0         0         0
       0         0         0
  ACT V1    ACT V2    ACT V3
        Plot ACT trip
      LAST WD TRIP:
        1122263983
H1  V1  H2  V2  H3  V3

STAGE 1        STAGE 2
   STATE 1:
   ARMED
   STATE 2:
   COUNTING
   STATE 3:
   DAMPING ONLY
   STATE 4:
   FULL SHUTDOWN
      RESET ALL
  CLEAR SATURATIONS
     IOPDACKILL
     DACKILL
  Rogue Exc WD   RESET
 reset threshold levels
   ST1            ST2
 Last           Last
   CPS            CPS
   T240           GS13
   L4C            ST1WD
   CD BIO         CD BIO
   ACT            ACT
   PAYLOAD        PAYLOAD
   IOP WD         IOP WD
   HW WD          HW WD
     S1:ISI-MAST
   RESET WD ONLY

          MONITORS
  DISP H1   DISP H2   DISP H3
     -175       350      -180
     -159       264      -206
  DISP V1   DISP V2   DISP V3
        Plot CPS trip
  GEO H1    GEO H2    GEO H3
    3131         0      -496
   -3994      3528     -2201
  GEO V1    GEO V2    GEO V3
        Plot GS13 trip
  ACT H1    ACT H2    ACT H3
       0         0         0
       0         0         0
  ACT V1    ACT V2    ACT V3
        Plot ACT trip
      LAST WD TRIP:
        1122263983
H1  V1  H2  V2  H3  V3

FIRST TRIG   CURRENT TRIG
10 SATURATIONS MAX
                SET RESET
CPS limit:      32760
safe value:     32760
                SET RESET
GS13 limit:     32760
safe value:     32760
                SET RESET
ACT limit:      32760
safe value:     32760
Stage 1 Watchdog Trip
PAYLOAD Watchdog Trip
 Click to open link
COIL DRIVER CHASSIS BIO
IOP Watchdog trip
```

Fig.4. “Saturation Clear” test sequence part 1



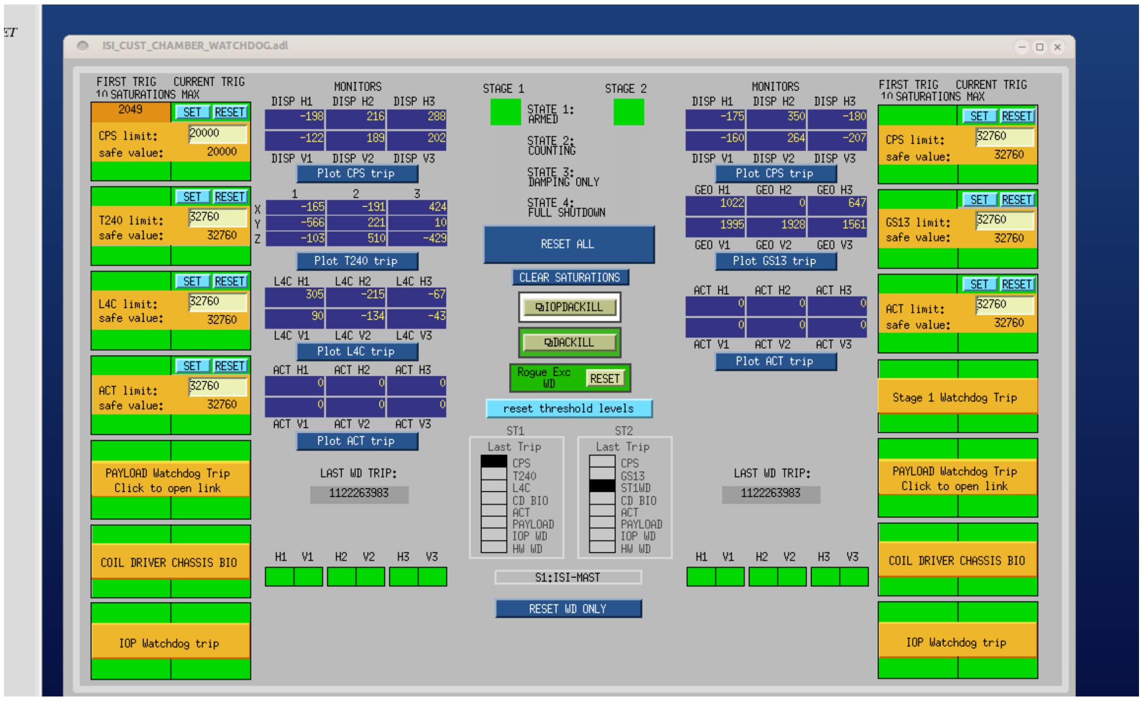
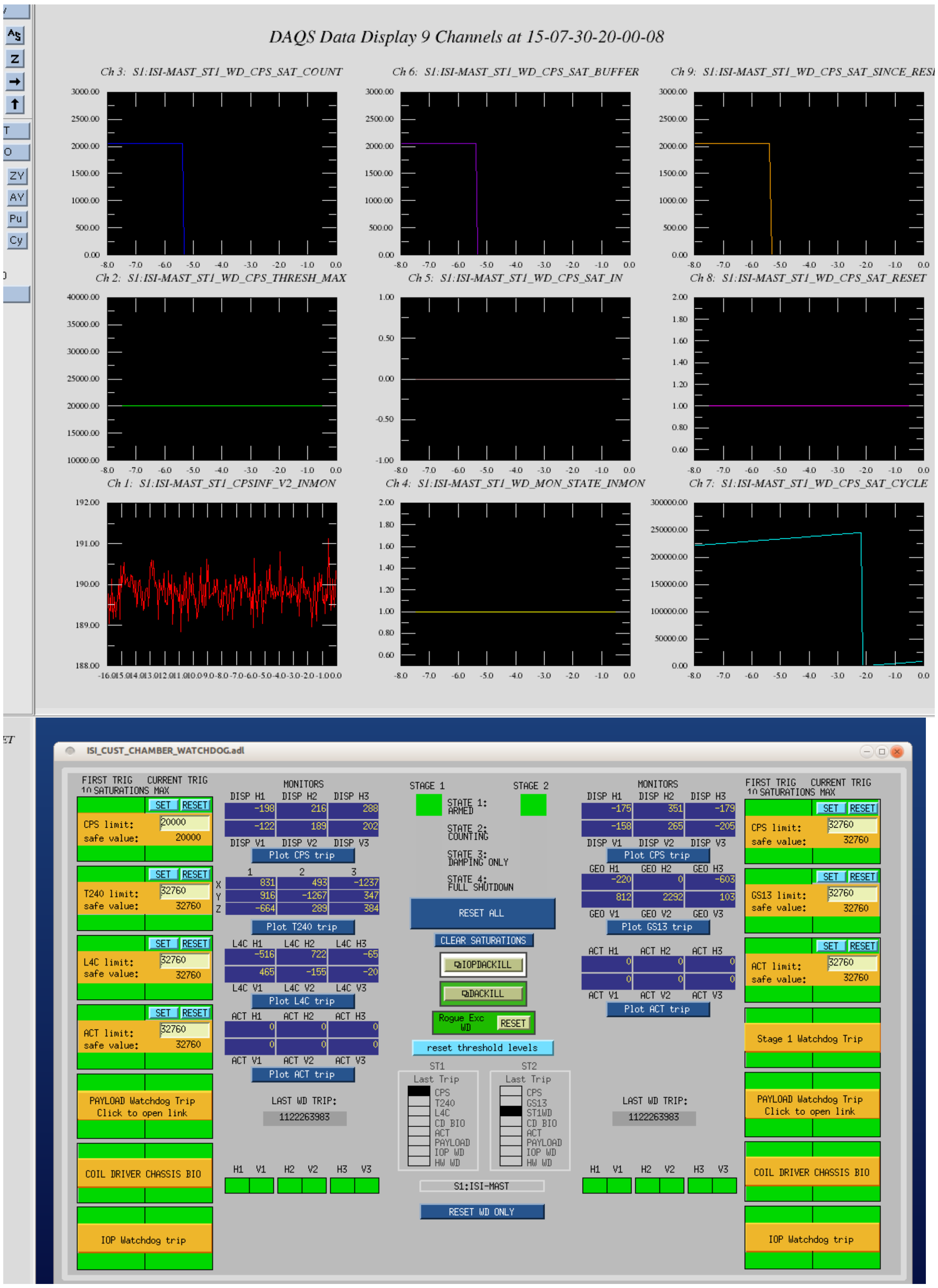

```
ISI_CUST_CHAMBER_WATCHDOG.adl

FIRST TRIG   CURRENT TRIG
10 SATURATIONS MAX
2049
CPS limit:   20000
safe value:  20000

T240 limit:  32760
safe value:  32760

L4C limit:   32760
safe value:  32760

ACT limit:   32760
safe value:  32760

PAYLOAD Watchdog Trip
Click to open link

COIL DRIVER CHASSIS BIO

IOP Watchdog trip

MONITORS
DISP H1  DISP H2  DISP H3
  -198      216      288
  -122      189      202
DISP V1  DISP V2  DISP V3
Plot CPS trip
      1        2        3
X  -165     -191      424
Y  -566      221       10
Z  -103      510     -429
Plot T240 trip
L4C H1   L4C H2   L4C H3
   305     -215      -67
    90     -134      -43
L4C V1   L4C V2   L4C V3
Plot L4C trip
ACT H1   ACT H2   ACT H3
     0        0        0
     0        0        0
ACT V1   ACT V2   ACT V3
Plot ACT trip

LAST WD TRIP:
1122263983

H1 V1 H2 V2 H3 V3

STAGE 1        STAGE 2
STATE 1: ARMED
STATE 2: COUNTING
STATE 3: DAMPING ONLY
STATE 4: FULL SHUTDOWN
RESET ALL
CLEAR SATURATIONS
IOPDACKILL
DACKILL
Rogue Exc WD   RESET
reset threshold levels
ST1 Last Trip: CPS T240 L4C CD BIO ACT PAYLOAD IOP WD HW WD
ST2 Last Trip: CPS GS13 ST1WD CD BIO ACT PAYLOAD IOP WD HW WD
S1:ISI-MAST
RESET WD ONLY

MONITORS
DISP H1  DISP H2  DISP H3
  -175      350     -180
  -160      264     -207
DISP V1  DISP V2  DISP V3
Plot CPS trip
GEO H1   GEO H2   GEO H3
  1022        0      647
  1995     1928     1561
GEO V1   GEO V2   GEO V3
Plot GS13 trip
ACT H1   ACT H2   ACT H3
     0        0        0
     0        0        0
ACT V1   ACT V2   ACT V3
Plot ACT trip

LAST WD TRIP:
1122263983

H1 V1 H2 V2 H3 V3

FIRST TRIG   CURRENT TRIG
10 SATURATIONS MAX
CPS limit:   32760
safe value:  32760

GS13 limit:  32760
safe value:  32760

ACT limit:   32760
safe value:  32760

Stage 1 Watchdog Trip

PAYLOAD Watchdog Trip
Click to open link

COIL DRIVER CHASSIS BIO

IOP Watchdog trip
```

Fig.5. “Saturation Clear” test sequence part 2


DAQS Data Display 9 Channels at 15-07-30-20-00-08
Ch 3: S1:ISI-MAST_ST1_WD_CPS_SAT_COUNT
Ch 6: S1:ISI-MAST_ST1_WD_CPS_SAT_BUFFER
Ch 9: S1:ISI-MAST_ST1_WD_CPS_SAT_SINCE_RESI
Ch 2: S1:ISI-MAST_ST1_WD_CPS_THRESH_MAX
Ch 5: S1:ISI-MAST_ST1_WD_CPS_SAT_IN
Ch 8: S1:ISI-MAST_ST1_WD_CPS_SAT_RESET
Ch 1: S1:ISI-MAST_ST1_CPSINF_V2_INMON
Ch 4: S1:ISI-MAST_ST1_WD_MON_STATE_INMON
Ch 7: S1:ISI-MAST_ST1_WD_CPS_SAT_CYCLE
ISI_CUST_CHAMBER_WATCHDOG.adl

FIRST TRIG CURRENT TRIG
10 SATURATIONS MAX
SET RESET
CPS limit: 20000
safe value: 20000
SET RESET
T240 limit: 32760
safe value: 32760
SET RESET
L4C limit: 32760
safe value: 32760
SET RESET
ACT limit: 32760
safe value: 32760
PAYLOAD Watchdog Trip
Click to open link
COIL DRIVER CHASSIS BIO
IOP Watchdog trip

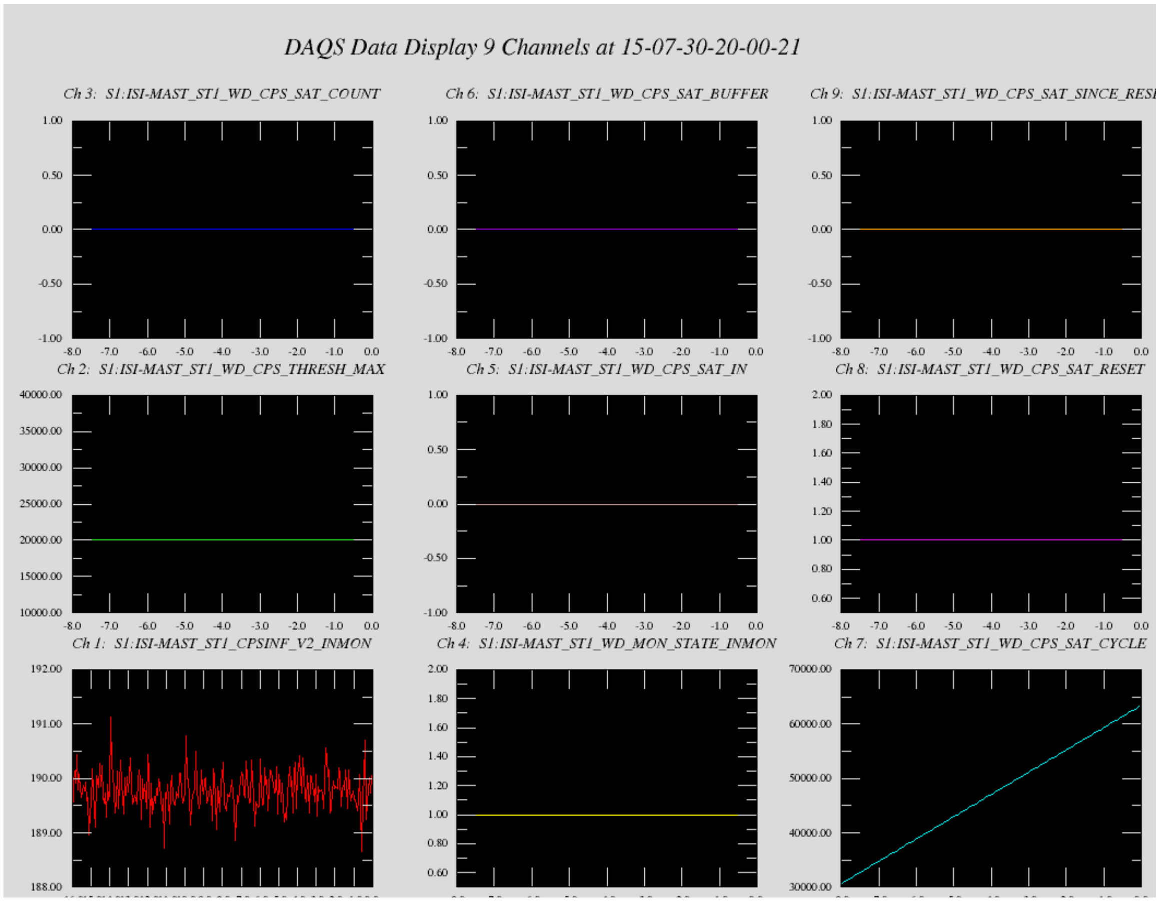
MONITORS
DISP H1 DISP H2 DISP H3
-198 216 288
-122 189 202
DISP V1 DISP V2 DISP V3
Plot CPS trip
1 2 3
X 831 493 -1237
Y 916 -1267 347
Z -664 289 384
Plot T240 trip
L4C H1 L4C H2 L4C H3
-516 722 -65
465 -155 -20
L4C V1 L4C V2 L4C V3
Plot L4C trip
ACT H1 ACT H2 ACT H3
0 0 0
0 0 0
ACT V1 ACT V2 ACT V3
Plot ACT trip
LAST WD TRIP:
1122263983
H1 V1 H2 V2 H3 V3

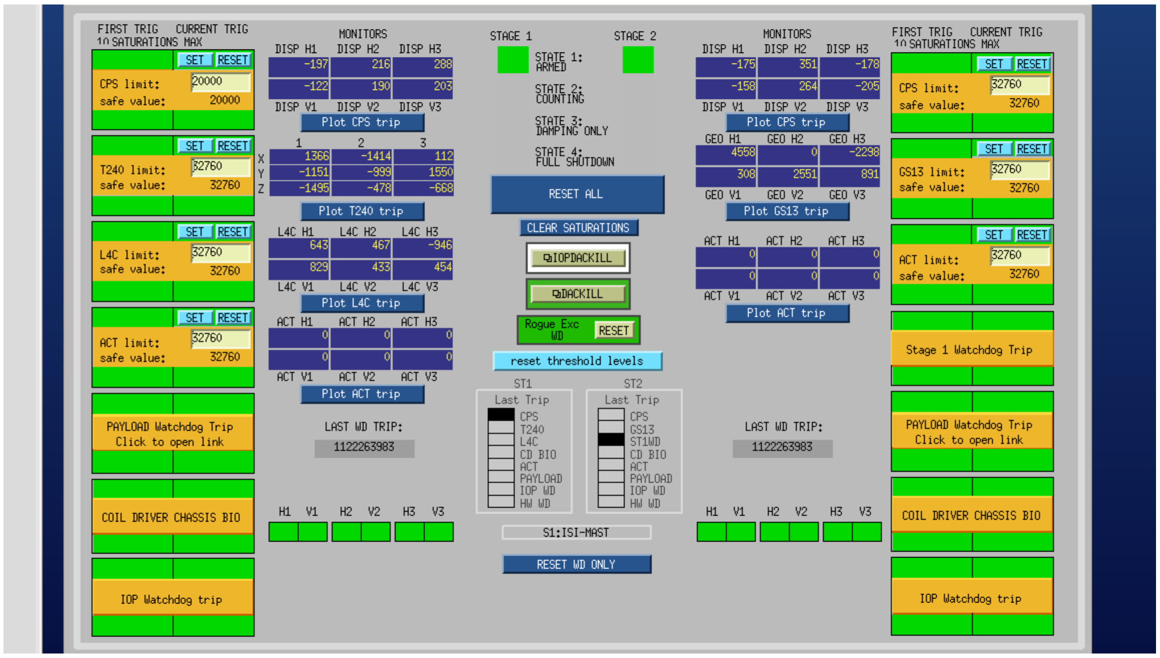
STAGE 1 STAGE 2
STATE 1: ARMED
STATE 2: COUNTING
STATE 3: DAMPING ONLY
STATE 4: FULL SHUTDOWN
RESET ALL
CLEAR SATURATIONS
IOPDACKILL
DACKILL
Rogue Exc WD RESET
reset threshold levels
ST1 ST2
Last Trip: CPS, T240, L4C, CD BIO, ACT, PAYLOAD, IOP WD, HW WD
Last Trip: CPS, GS13, ST1WD, CD BIO, ACT, PAYLOAD, IOP WD, HW WD
S1:ISI-MAST
RESET WD ONLY

MONITORS
DISP H1 DISP H2 DISP H3
-175 351 -179
-158 265 -205
DISP V1 DISP V2 DISP V3
Plot CPS trip
GEO H1 GEO H2 GEO H3
-220 0 -603
812 2292 103
GEO V1 GEO V2 GEO V3
Plot GS13 trip
ACT H1 ACT H2 ACT H3
0 0 0
0 0 0
ACT V1 ACT V2 ACT V3
Plot ACT trip
LAST WD TRIP:
1122263983
H1 V1 H2 V2 H3 V3

FIRST TRIG CURRENT TRIG
10 SATURATIONS MAX
SET RESET
CPS limit: 32760
safe value: 32760
SET RESET
GS13 limit: 32760
safe value: 32760
SET RESET
ACT limit: 32760
safe value: 32760
Stage 1 Watchdog Trip
PAYLOAD Watchdog Trip
Click to open link
COIL DRIVER CHASSIS BIO
IOP Watchdog trip

Fig.6. “Saturation Clear” test sequence part 3




FIRST TRIG CURRENT TRIG
10 SATURATIONS MAX

SET RESET
CPS limit: 20000
safe value: 20000

SET RESET
T240 limit: 32760
safe value: 32760

SET RESET
L4C limit: 32760
safe value: 32760

SET RESET
ACT limit: 32760
safe value: 32760

PAYLOAD Watchdog Trip
Click to open link

COIL DRIVER CHASSIS BIO

IOP Watchdog trip

MONITORS

	DISP H1	DISP H2	DISP H3
	-197	216	288
	-122	190	203
	DISP V1	DISP V2	DISP V3

Plot CPS trip

	1	2	3
X	1366	-1414	112
Y	-1151	-999	1550
Z	-1495	-478	-668

Plot T240 trip

L4C H1	L4C H2	L4C H3
643	467	-946
829	433	454
L4C V1	L4C V2	L4C V3

Plot L4C trip

ACT H1	ACT H2	ACT H3
0	0	0
0	0	0
ACT V1	ACT V2	ACT V3

Plot ACT trip

LAST WD TRIP:
1122263983

H1 V1 H2 V2 H3 V3

STAGE 1 STAGE 2
STATE 1: ARMED
STATE 2: COUNTING
STATE 3: DAMPING ONLY
STATE 4: FULL SHUTDOWN

RESET ALL
CLEAR SATURATIONS
IOPDACKILL
DACKILL
Rogue Exc WD RESET
reset threshold levels

ST1 Last Trip: CPS, T240, L4C, CD BIO, ACT, PAYLOAD, IOP WD, HW WD
ST2 Last Trip: CPS, GS13, ST1WD, CD BIO, ACT, PAYLOAD, IOP WD, HW WD

S1:ISI-MAST
RESET WD ONLY

MONITORS

DISP H1	DISP H2	DISP H3
-175	351	-178
-158	264	-205
DISP V1	DISP V2	DISP V3

Plot CPS trip

GEO H1	GEO H2	GEO H3
4558	0	-2298
308	2551	891
GEO V1	GEO V2	GEO V3

Plot GS13 trip

ACT H1	ACT H2	ACT H3
0	0	0
0	0	0
ACT V1	ACT V2	ACT V3

Plot ACT trip

LAST WD TRIP:
1122263983

H1 V1 H2 V2 H3 V3

FIRST TRIG CURRENT TRIG
10 SATURATIONS MAX

SET RESET
CPS limit: 32760
safe value: 32760

SET RESET
GS13 limit: 32760
safe value: 32760

SET RESET
ACT limit: 32760
safe value: 32760

Stage 1 Watchdog Trip

PAYLOAD Watchdog Trip
Click to open link

COIL DRIVER CHASSIS BIO

IOP Watchdog trip

Fig.7. “Saturation Clear” test sequence part 4

II) Saturation Counters

a) Description

The watchdog is setup so that if the total saturations in the history are greater than the allowed number of saturations then the watchdog will trip. Currently, the history is kept until the watchdog trips or the history is reset. We propose changing the saturation history buffering so that saturations more than 1 hour old (value defined in the simulink model) are automatically cleared. The history will be stored in 60 bins, each 1 minute long. Bins more that 1 hour old are cleared.

The saturation counters are now handled by a new piece of c-code called WD_SATCOUNT.c. This code presented in *II) b) c-code*, can be found under $userapps/isi/common/src/.

The design of WD_SATCOUNT.c is inspired by circular buffers, also called ring buffers. The idea of such a system is that a circular window is used to store buffered data, The data is stored in subdivisions of the main window called bins, which get replaced, once the main window is full, starting from the oldest.

In the illustration below (Fig.8), the *main window* is the full circle, its subdivisions are the *bins* and the purple content is pasted from the buffer into the bins, oldest bins getting replaced first, as the main window fills up.

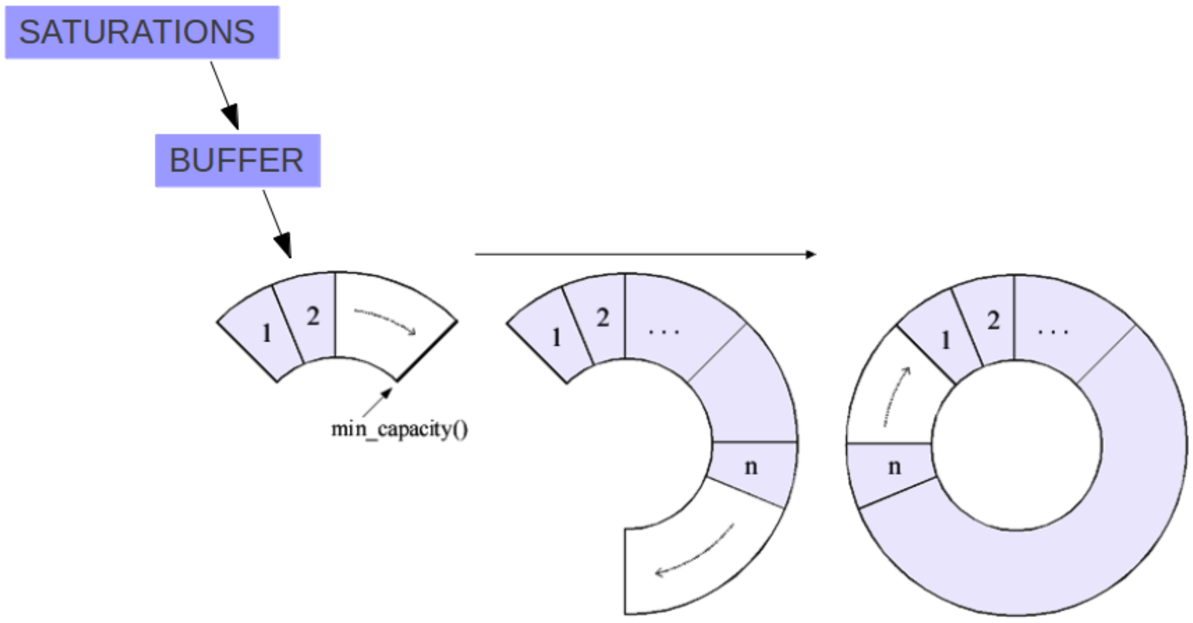


Fig.8 Circular buffer illustration

b) c-code (as of Aug 12th 2015)

```
/* WD_SATCOUNT.c Function: WD_SATCOUNT.c
 *
 * SATCOUNT is a WD saturation counter that only accounts for saturations
 * that occured during a given time window. The way saturations are
 * stored in this time window is inspired by ring buffers. Saturations
 * are stored in the limited number of bins the main time window is made of.
 * Bins get replaced, oldest first, when the main time window is full.
 *
 * Inputs:
 *--------
 * (1) int SATURATIONS: wd saturations signal
 *      0: no saturation
 *      1+: Number of saturations registered during the current FE cycle
 * (2) int WINDOWTIME: total time observed by the main time window (in seconds)
 * (3) int RESET: reset signal
 *             0: Clear the saturation counter
 *             1: Leave the saturation counter
 *
 * Outputs:
 * -------
 * (1) int SATURATIONS_TOTAL: total number of saturations to be compared
 *  against the SAFE_SAT_COUNT
 * (2) int BUFFER: current number of saturations in the buffer                  - FOR TESTING
 * (3) int CYCLE: current FE cycle number, modulo the number of cycles in a bin. - FOR TESTING
 * (4) int RESET: reset request received.                                        - FOR TESTING
 * (5) int SAT_SINCE_RESET total number of saturations since last reset          - FOR LONG TERM
MONITORING
 *
 * Authors: HRAP, BTL
 * July 2015
 *
 * See ECR E1500325 / integration issue 1086
 *
 * Aug 15, 2015  BTL removed test cases from end and fixed the CLEAR_BUG
 * BTL fix RESET behavior so that history clear works.
 * bug was that the precomputed history sum (BIN_SAT_TOTAL) was not being set to 0
 */


#define MODEL_RATE FE_RATE // makes a constant out of the model rate
#define BINS 60 //number of bins within time window

void ISISATCOUNT(double *argin, int nargin, double *argout, int nargout){
    int SATURATION_IN = argin[0];
    int WINDOWTIME    = argin[1];
    int RESET         = argin[2];

// ---------------------Initialisations------------------------------------

    //  Note: as of now there is no regulation to insure that the TIMEWINDOW / BINS doesn't have a remainer
    int WINDOW_CYCLES, NUM_BINCYCLES;
    WINDOW_CYCLES    =   WINDOWTIME * MODEL_RATE; // window duration in FE cycles
    NUM_BINCYCLES    =   WINDOW_CYCLES / BINS; // number of cycles per bin

    // CYCLE keeps track of the FE cycles
    // using "Static int" allows keeping value between FE cycles
    static int CYCLE = 0;

    // Initialize WINDOW with empty bins
      // "static int WINDOW[BINS]={0};" Oly works if BIN is a constant (define).
```

```
        // It wouldn't work otherwise because
            // when the compilers sees your variable declaration
            // interpreter returns: "variable-sized object may not be initialized"
            static int WINDOW[BINS]={0};

            // Total number of saturations since last reset
            static int SAT_SINCE_RESET = 0;

    // Initialize:
    //                  - BUFFER: Buffer whithin which saturations are summed into bins
    //                  - AVAILABLE_BIN: Available bin (oldest one) in the WINDOW array
    //      - BIN_SAT_TOTAL: sum of the bin history
     //                 - SATURATIONS_TOTAL: bin history + number of sat. in current BUFFER - main output
    static int BUFFER             = 0;
    static int AVAILABLE_BIN      = 0;
    static int BIN_SAT_TOTAL      = 0;
    int SATURATIONS_TOTAL         = 0;
    int i                         = 0; // simple counter

// ---------------------Computation----------------------------------

    // Apply RESET:
    if (RESET == 0)
    {
        BUFFER = 0; // clear buffer
        SAT_SINCE_RESET = 0; // clear the number of saturations since last reset.
        SATURATIONS_TOTAL = 0;
        BIN_SAT_TOTAL = 0;   // this fixes the initial 'not clearing consistantly' bug.
        for(i = 0; i < BINS; i++)
        {
            WINDOW[i] = 0 ; // clear window
        }
     }

    // Store the number of saturations in the buffer, and the number of saturations
    // since last reset, at every sample
    BUFFER          = BUFFER            + SATURATION_IN ;
    SAT_SINCE_RESET = SAT_SINCE_RESET   + SATURATION_IN ;

    // Update the cycle number
    CYCLE++;

    // If new bin ready
    if (CYCLE >= NUM_BINCYCLES)
    {
        // Store buffer in next available bin in the window
        WINDOW[AVAILABLE_BIN] = BUFFER;

        // Update AVAILABLE_BIN location in WINDOW
        AVAILABLE_BIN++;

        // Limit AVAILABLE_BIN to the number of bins avaialble in WINDOW
        if (AVAILABLE_BIN >= BINS)
        {
            AVAILABLE_BIN = 0 ;
        }

        // Clear BUFFER
        BUFFER = 0;

        // start counting CYCLES over
        CYCLE = 0;
        // SUM the saturations in the whole window
        BIN_SAT_TOTAL = 0; // initialize sum
        for(i = 0; i < BINS; i++)
        {
            BIN_SAT_TOTAL = BIN_SAT_TOTAL + WINDOW[i] ; // sum of the saturations in the whole window
        }

    }


    // SUM the current amount of saturations
    // (SATURATIONS_TOTAL is the output of the c-code block)
    SATURATIONS_TOTAL = BIN_SAT_TOTAL + BUFFER;


    // output the total number of saturations
    argout[0] = SATURATIONS_TOTAL;
    argout[1] = BUFFER;
    argout[2] = CYCLE;
    argout[3] = RESET;
    argout[4] = SAT_SINCE_RESET;

    return;

}
```

c) Model implementation

A WINDOWTIME flag (bottom right) goes into the CPS block (Fig.9). CPS block details are shown in Fig.10. Note the WINDOWTIME and RESET inputs to the c-code and the c-code's outputs (Fig. 10). The SAFECOUNT is recorded in an epics output for each device (CPS, GS13, ACT, etc...).

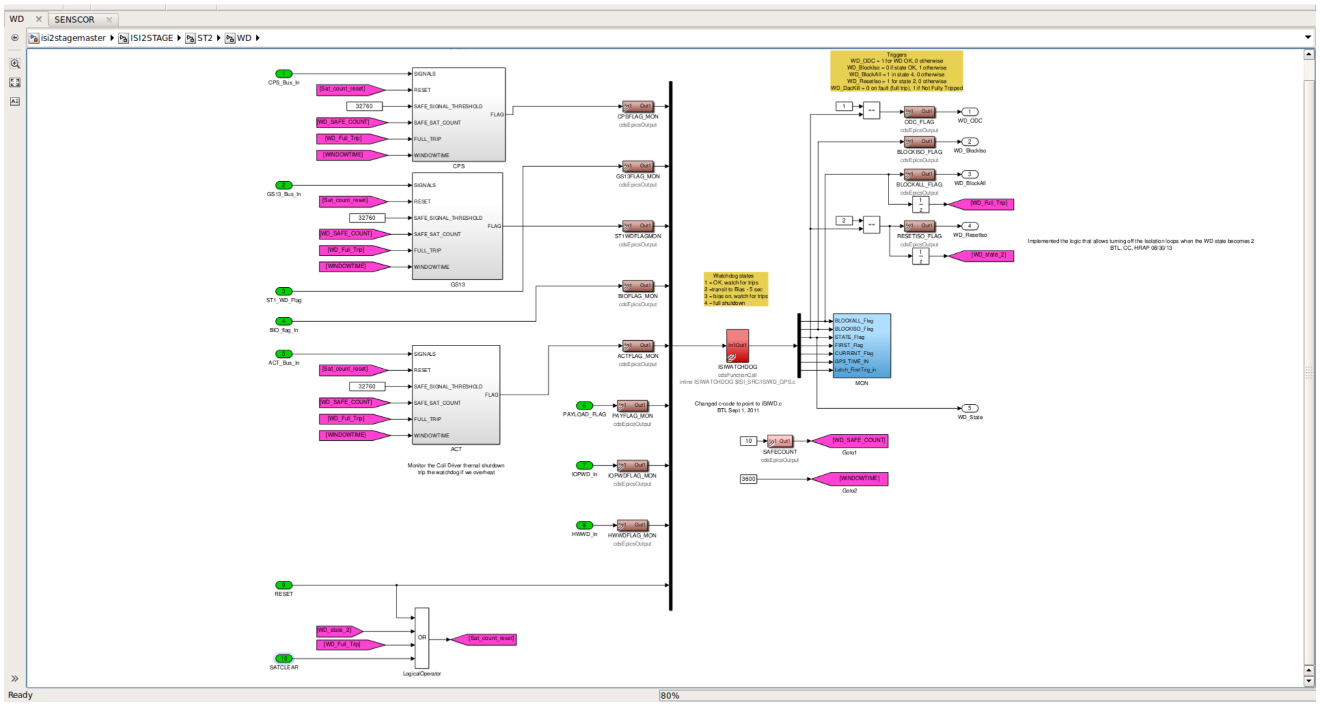


Fig. 9 BSC-ISI ST2 WD block.

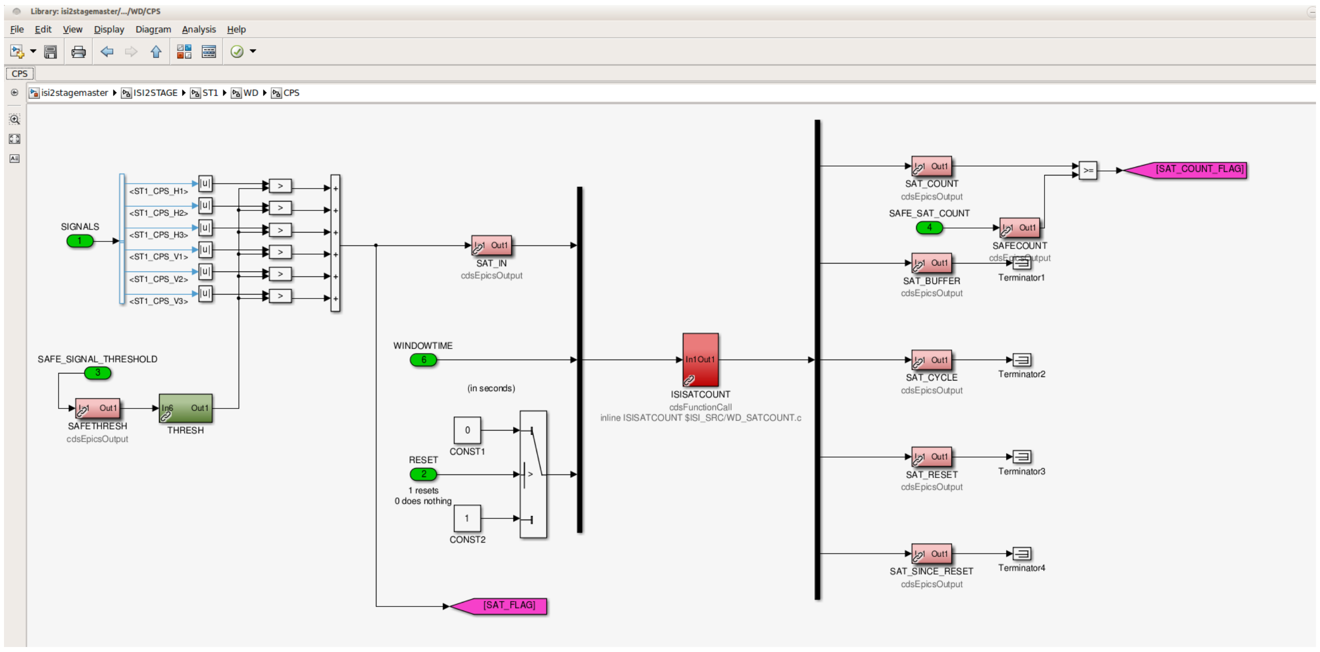


Fig. 10 BSC-ISI ST2 CPS WD block

d) Tests Performed

- Preliminary: saturation counter limit is temporarily set at 3000cts for this test
- Make sure model compiles and starts properly
- Saturate the CPS and make sure the WD trip when CPS saturation counters go over the saturation counter limit
- Clear WD and saturate the CPS again, but this time keep the number of saturations below the saturation counter limit .
- Make sure the number of saturations is properly displayed on WD MEDM screen
- Make sure that the “Saturation Clear” button works
- Make sure that uncleared saturations get automatically cleared once the main window gets renewed (3600s on final design)
- Switched the main window time to a short non-integer (20.1s). The cycle counter looks a bit strange. It doesn't always go to 0, or to its max value before/after being reset. However:
    - The WD trip properly when there is too many saturations
    - The saturations get accounted for properly
    - The WD_RESET and SATCLEAR buttons work properly
    - the WINDOWTIME is hardcoded in the mastermodel and should never be changed from 3600s without SEI approval

Note: the CPU max went up by about 10 after the model update.

- Make sure that the SAFECOUNT values can be monitored:

```
controls@ligo-ops2:~$ chndump | grep SAFECOUNT
S1:HPI-HAMX_WD_SAFECOUNT          1   0     0   4     16   0
S1:ISI-HAMX_WD_ACT_SAFECOUNT      1   0     0   4     16   0
S1:ISI-HAMX_WD_CPS_SAFECOUNT      1   0     0   4     16   0
S1:ISI-HAMX_WD_GS13_SAFECOUNT     1   0     0   4     16   0
S1:ISI-HAMX_WD_L4C_SAFECOUNT      1   0     0   4     16   0
S1:ISI-HAMX_WD_SAFECOUNT          1   0     0   4     16   0
S1:ISI-ITMX_ST1_WD_SAFECOUNT      1   0     0   4     16   0
S1:ISI-ITMX_ST2_WD_SAFECOUNT      1   0     0   4     16   0
S1:ISI-MAST_ST1_WD_ACT_SAFECOUNT  1   0     0   4     16   0
S1:ISI-MAST_ST1_WD_CPS_SAFECOUNT  1   0     0   4     16   0
S1:ISI-MAST_ST1_WD_L4C_SAFECOUNT  1   0     0   4     16   0
```

```
S1:ISI-MAST_ST1_WD_SAFECOUNT       1   0     0   4     16   0
S1:ISI-MAST_ST1_WD_T240_SAFECOUNT   1   0     0   4     16   0
S1:ISI-MAST_ST2_WD_ACT_SAFECOUNT   1   0     0   4     16   0
S1:ISI-MAST_ST2_WD_CPS_SAFECOUNT   1   0     0   4     16   0
S1:ISI-MAST_ST2_WD_GS13_SAFECOUNT   1   0     0   4     16   0
S1:ISI-MAST_ST2_WD_SAFECOUNT       1   0     0   4     16   0
```

Test Result: Passed.