

HAM ISI WATHCDOG

Implementation

Jeff Kissel

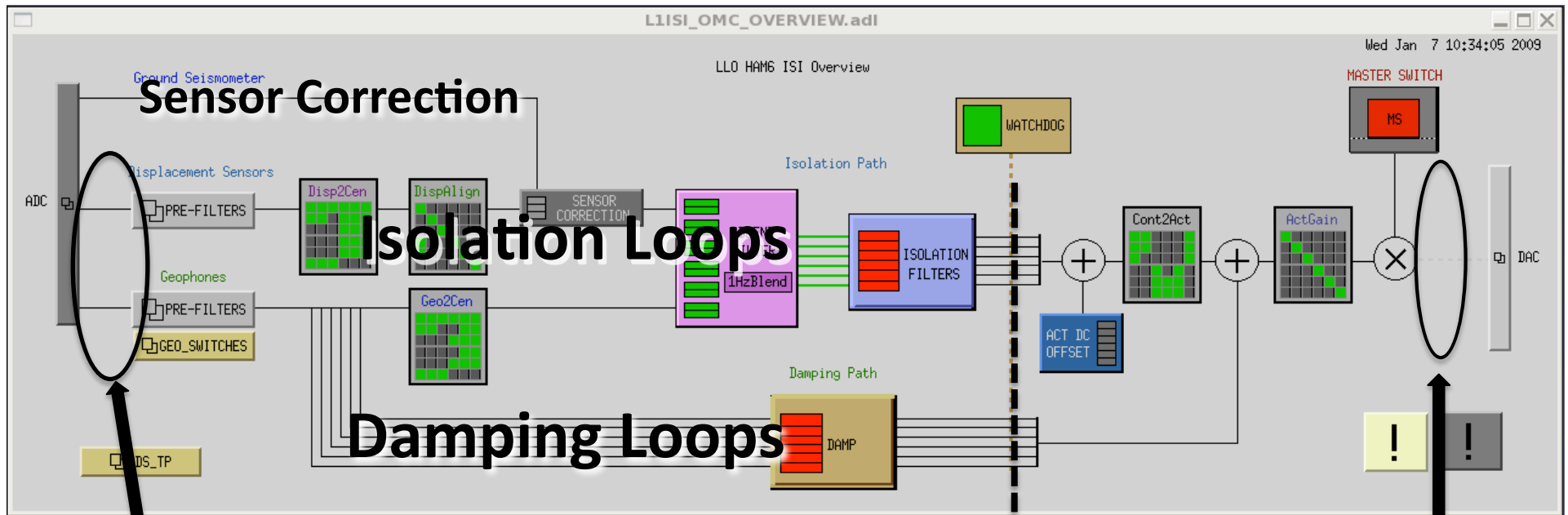
HAM ISI Watchdog Review, 1st Meeting

Jan 13, 2009

HAM ISI WATCHDOG Implementation

- State Diagram and overview
- Simulink Diagram
- Medm interface
- C Code (if desired)
- Auxiliary scripts (if desired)
- Features to be implemented

HAM ISI Watchdog Overview

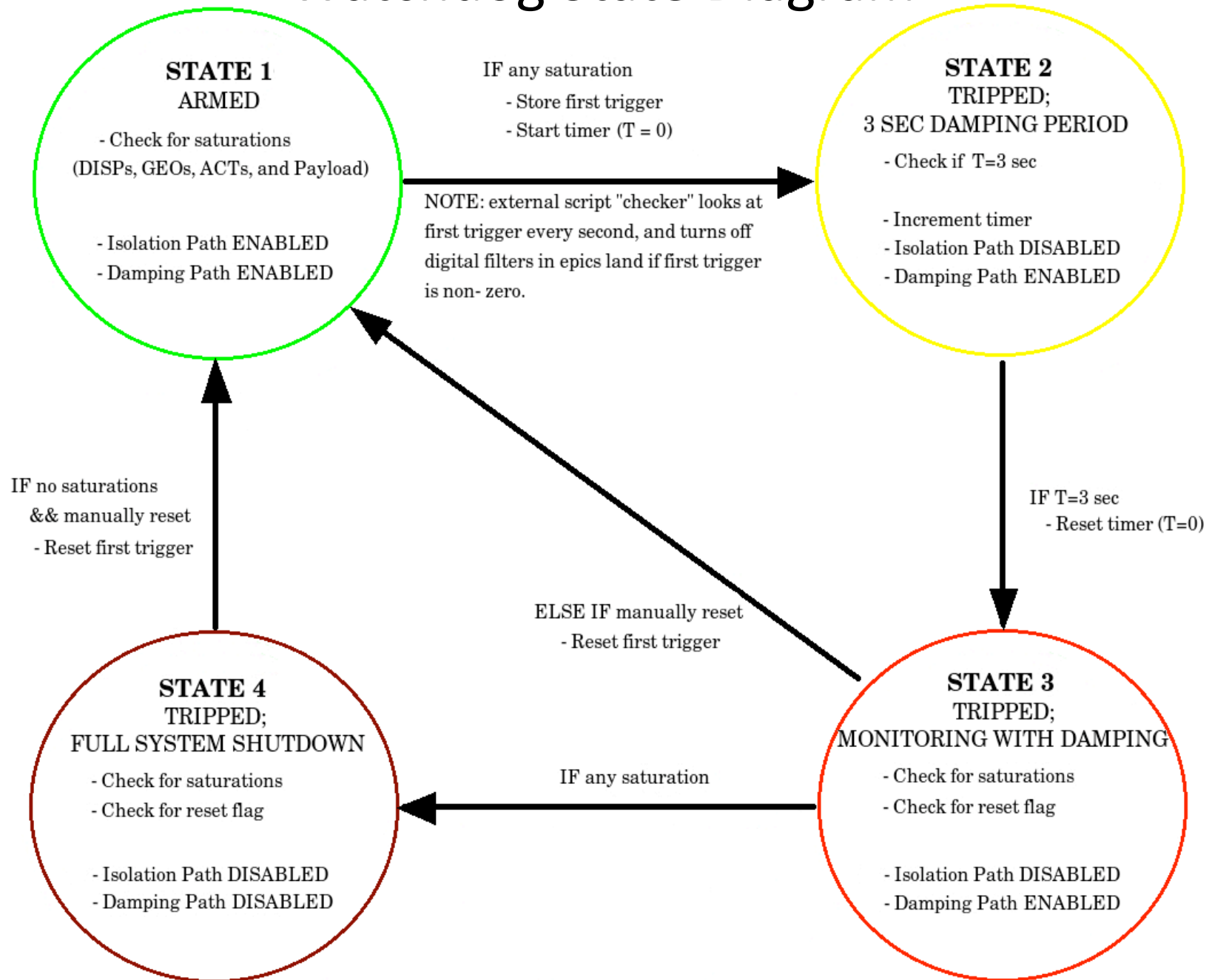


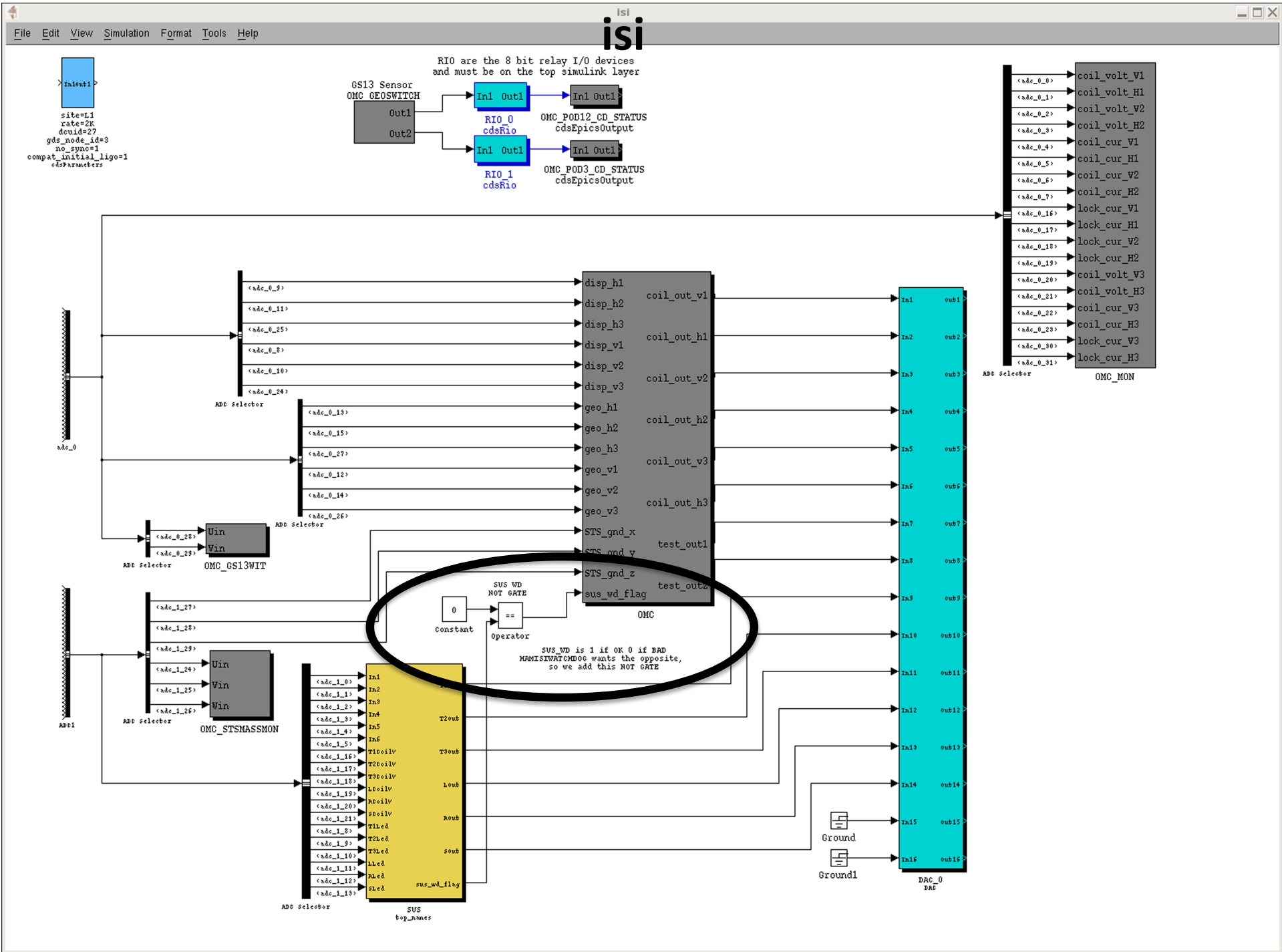
We watch displacement sensors and geophones here

We block loops here

We watch actuators here

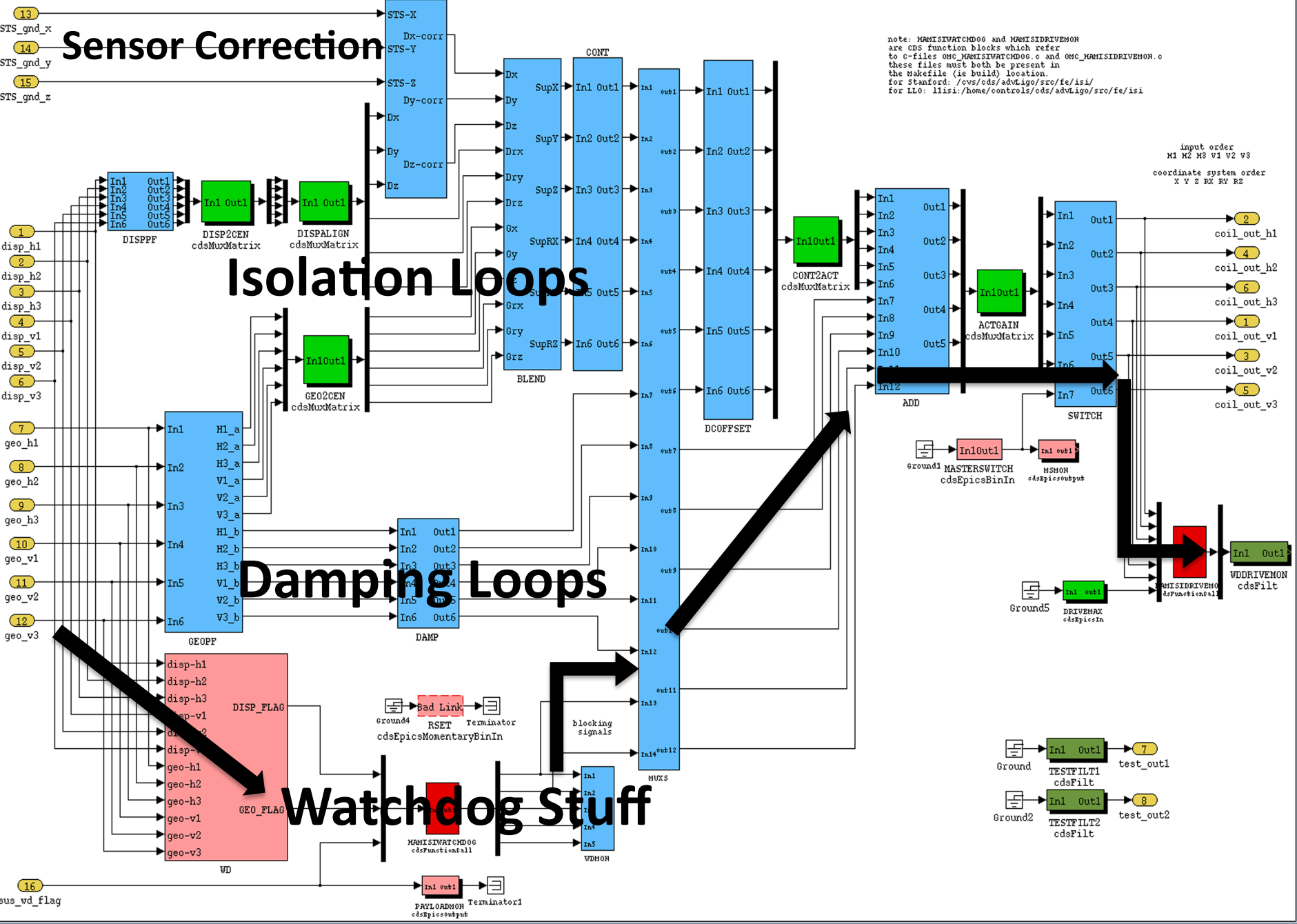
Watchdog State Diagram





isi/OMC

File Edit View Simulation Format Tools Help



Sensor Correction

Isolation Loops

Damping Loops

Watchdog Stuff

note: MAMISWATCHDOG6 and MAMISIDRIVEMON are CDS function blocks which refer to c-files OMC_MAMISWATCHDOG6.c and OMC_MAMISIDRIVEMON.c these files must both be present in the Makefile (ie build) location. for Stanford: /ovs/cds/adVligo/src/fe/isi/ for LL0: /llsi/home/controls/cds/adVligo/src/fe/isi

input order
M1 M2 M3 V1 V2 V3
coordinate system order
X Y Z RX RY RZ

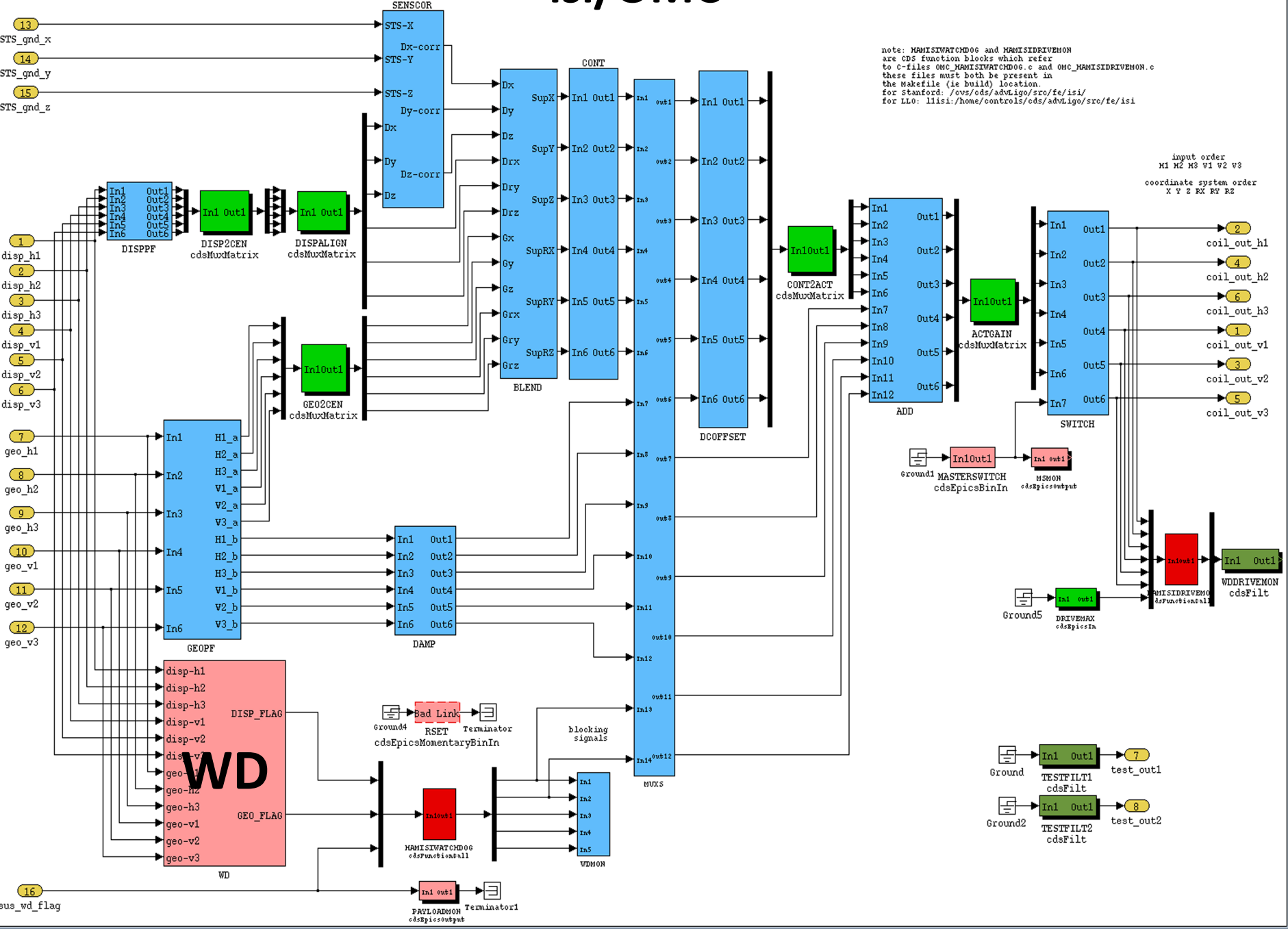
16
sus_wd_flag

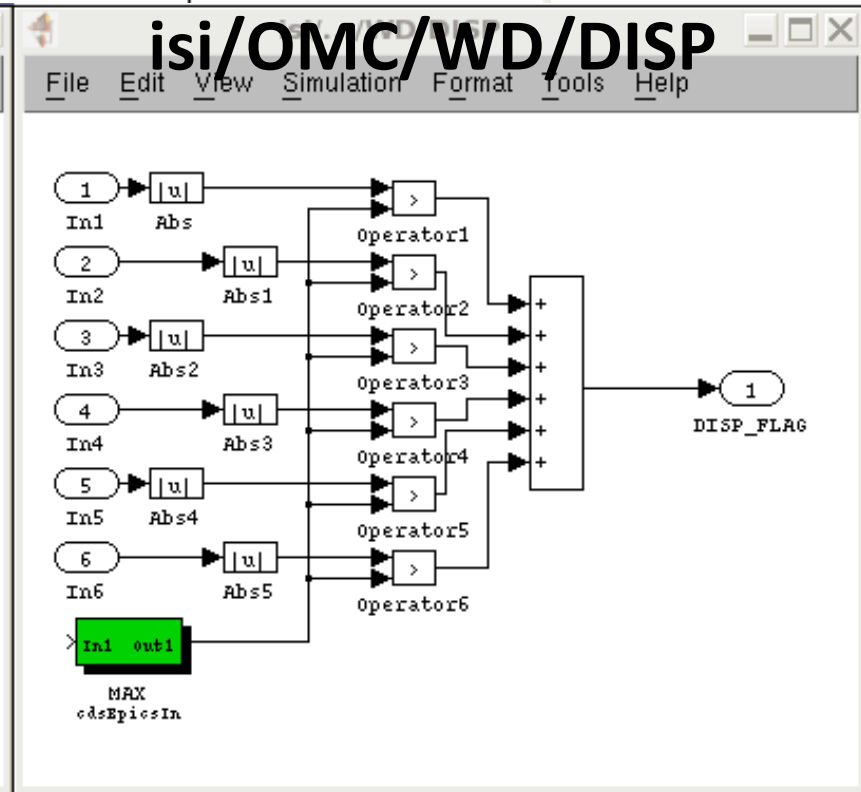
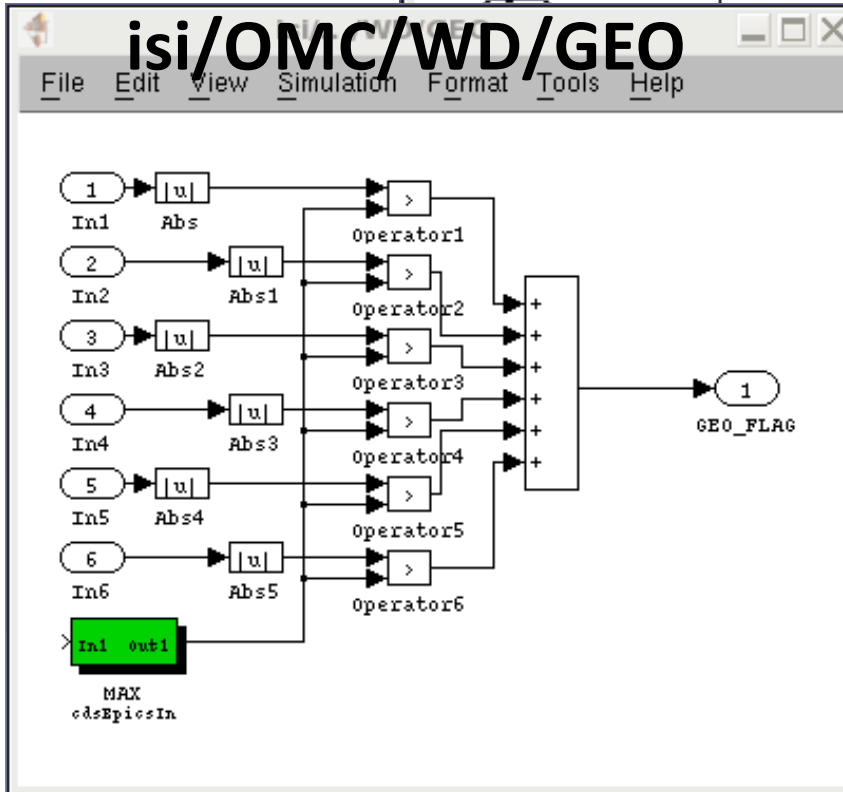
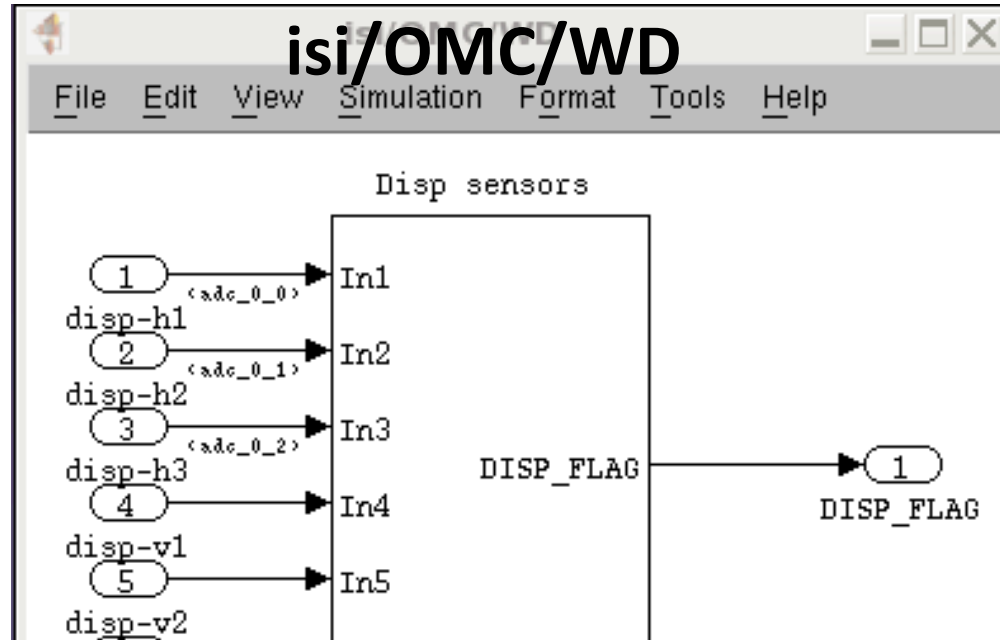
1
coil_out_h1
2
coil_out_h2
3
coil_out_h3
4
coil_out_v1
5
coil_out_v2
6
coil_out_v3

7
test_out1
8
test_out2

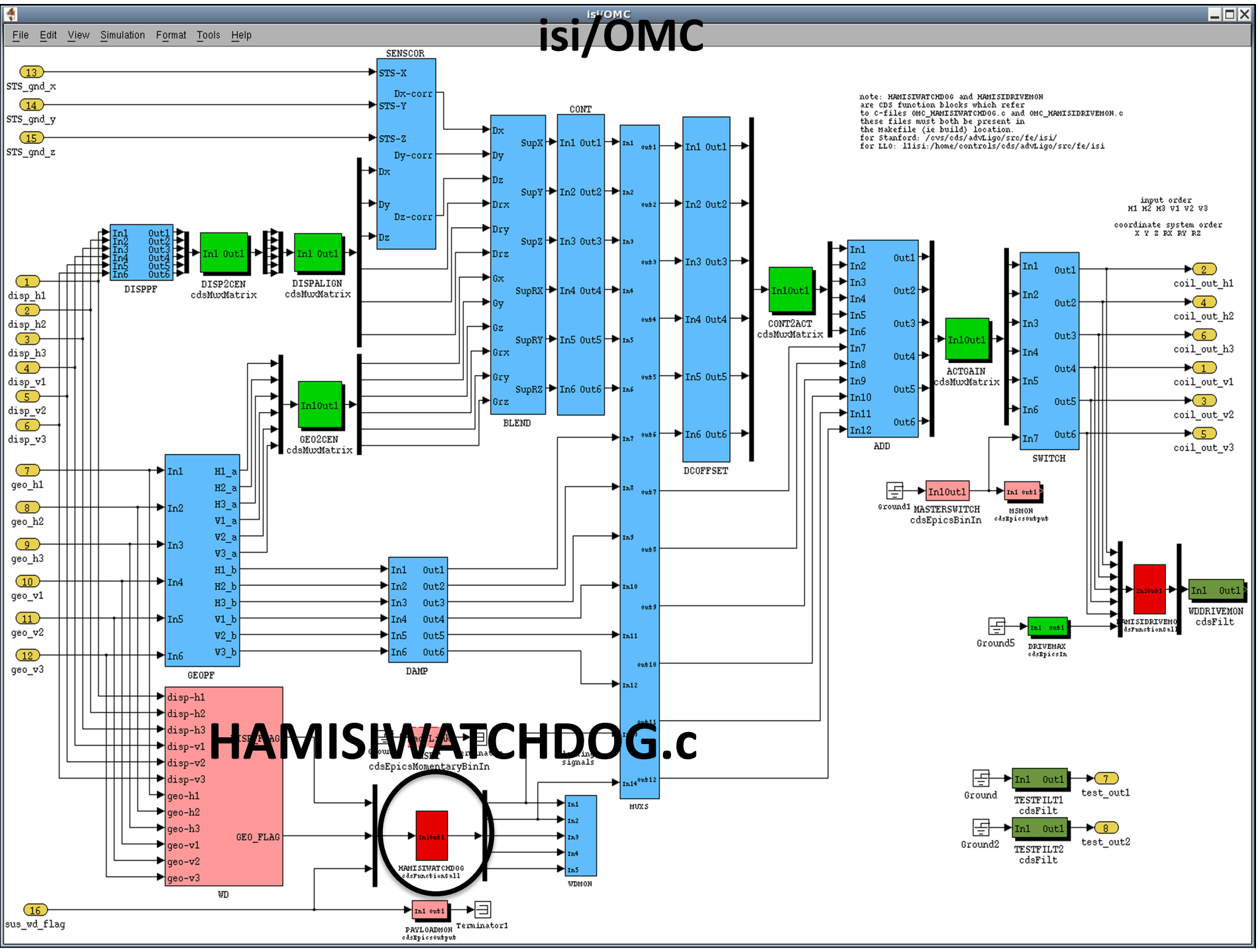
isi/OMC

File Edit View Simulation Format Tools Help





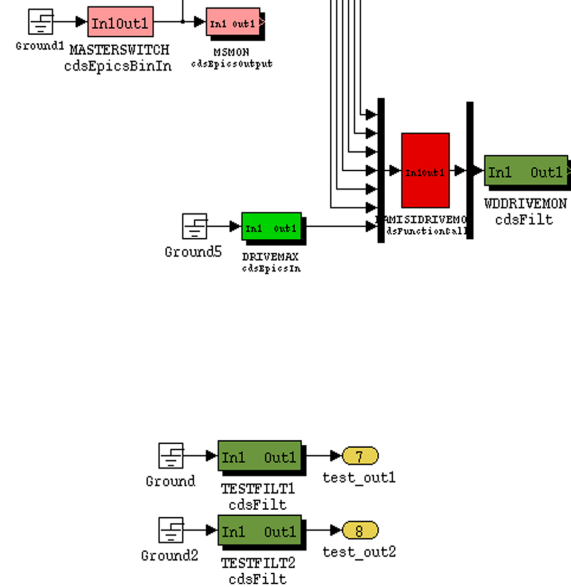
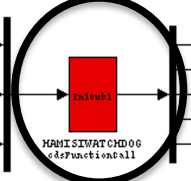
isi/OMC



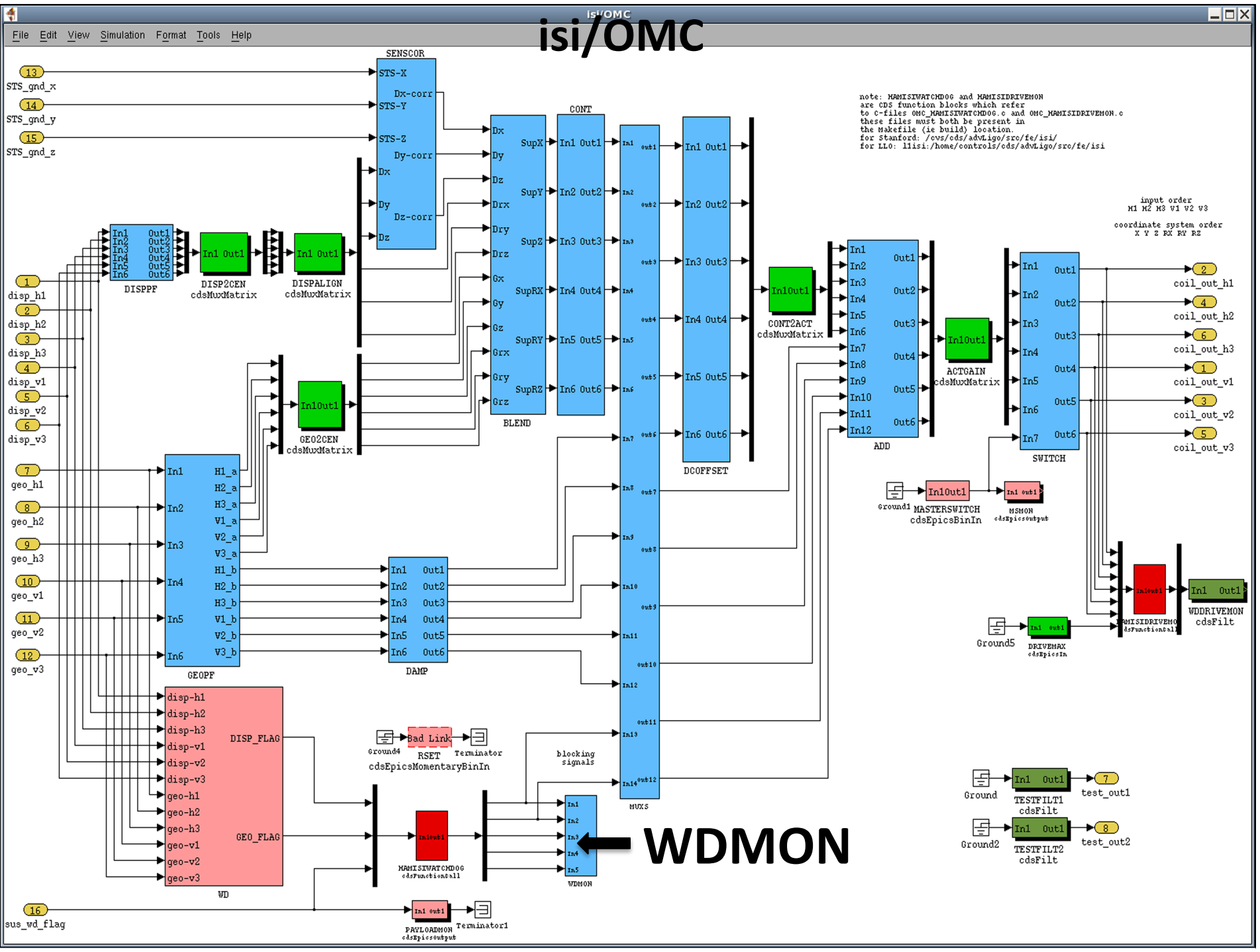
note: HAMISIVATCHDOG and HAMISIDRIVEMON are CDS function blocks which refer to c-files OMC_HAMISIVATCHDOG.c and OMC_HAMISIDRIVEMON.c these files must both be present in the Makefile (ie build) location. for Stanford: /ovs/cds/adVligo/src/fe/isi/ for LL0: /llsi/home/controls/cds/adVligo/src/fe/isi

input order
M1 M2 M3 V1 V2 V3
coordinate system order
X Y Z RX RY RZ

HAMISIVATCHDOG.c



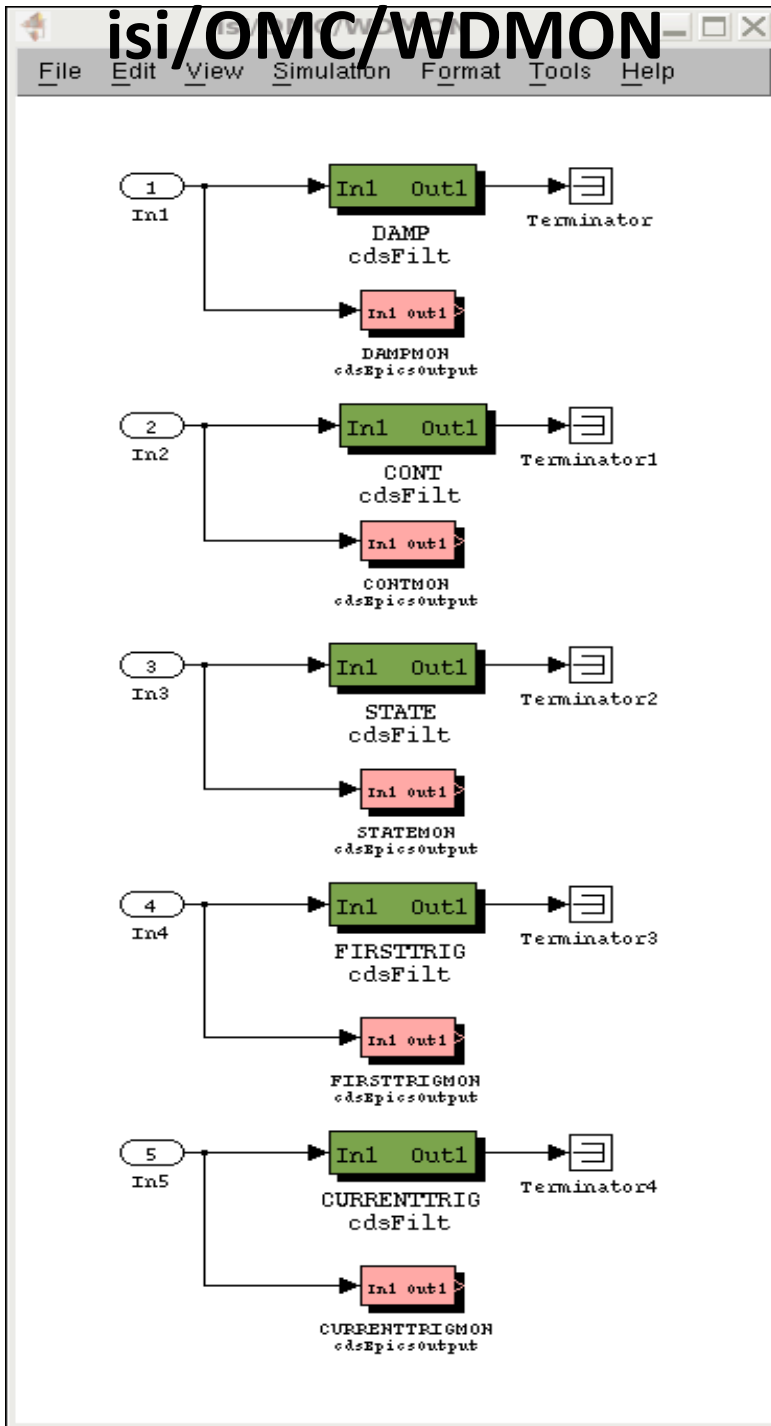
isi/OMC



note: HAMISIVATCMD06 and WDDRIVEMON are CDS function blocks which refer to c-files OMC_HAMISIVATCMD06.c and OMC_WDDRIVEMON.c these files must both be present in the Makefile (ie build) location. for Stanford: /ovs/cds/adVligo/src/fe/isi/ for LL0: /isi/home/controls/cds/adVligo/src/fe/isi

input order
M1 M2 M3 V1 V2 V3
coordinate system order
X Y Z RX RY RZ

WDMON



L1:ISI-OMC_WDMON_DAMPMON

L1:ISI-OMC_WDMON_CONTMON

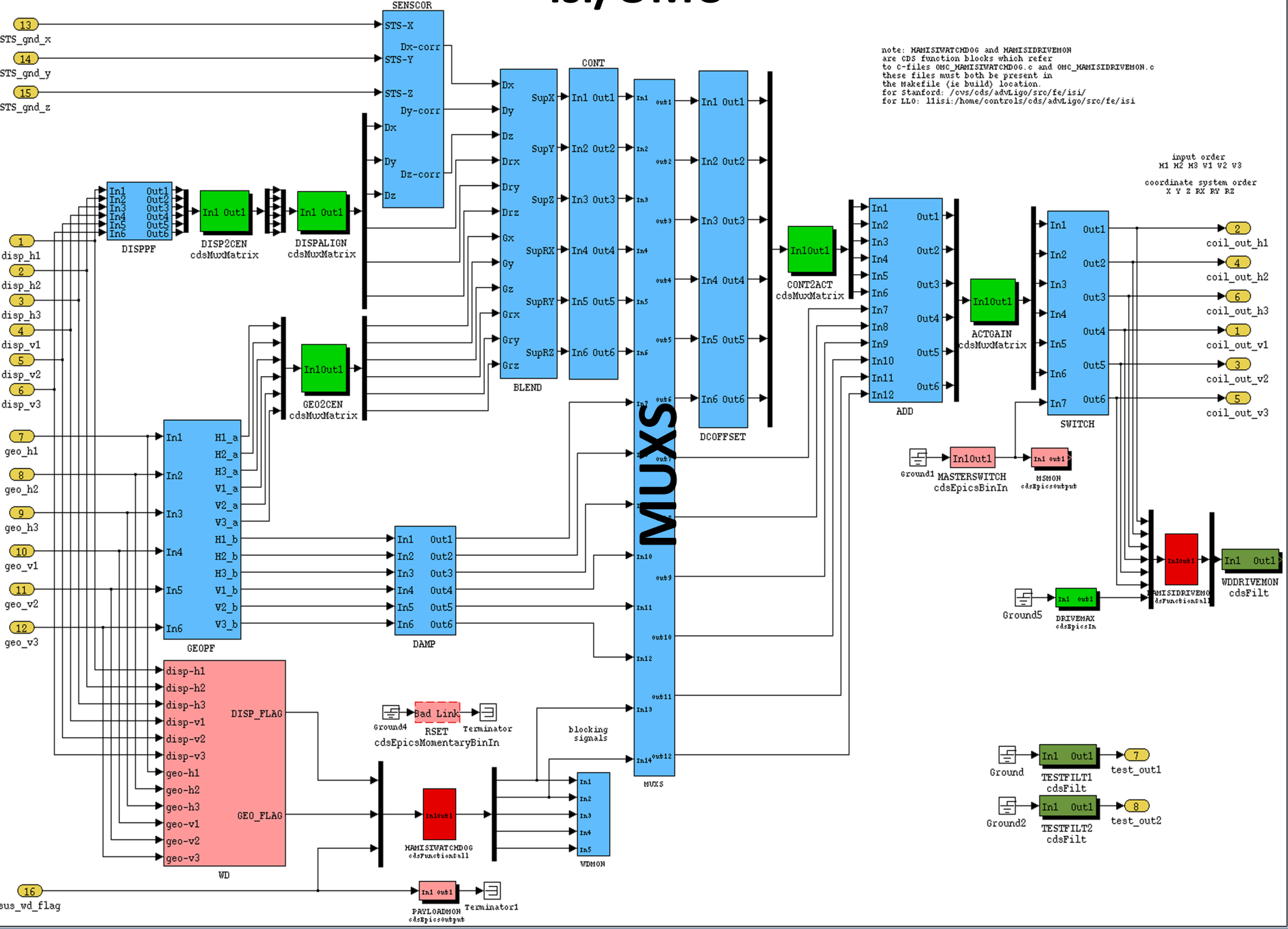
L1:ISI-OMC_WDMON_STATEMON

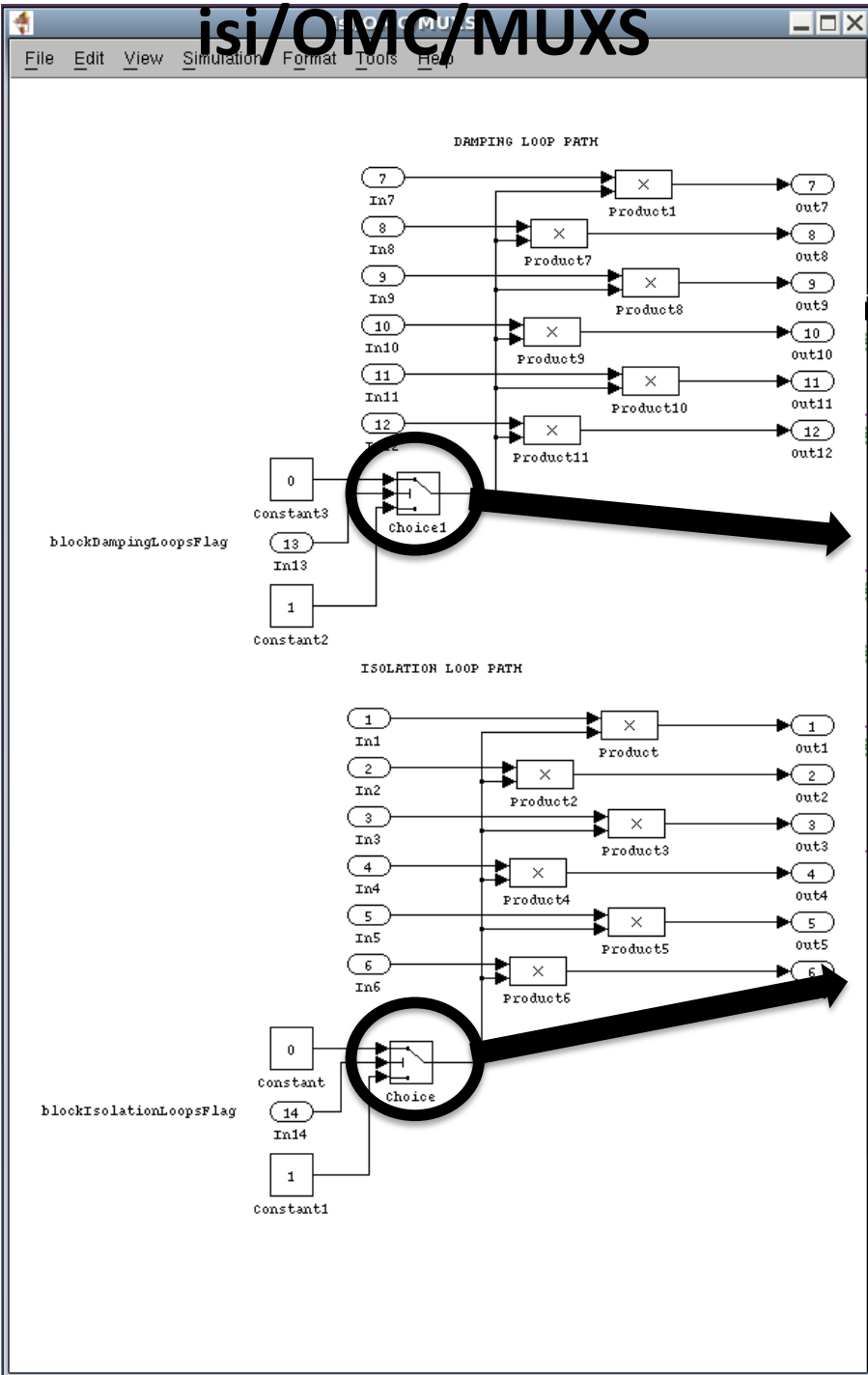
L1:ISI-OMC_WDMON_FIRSTTRIGMON

L1:ISI-OMC_WDMON_CURRENTTRIGMON

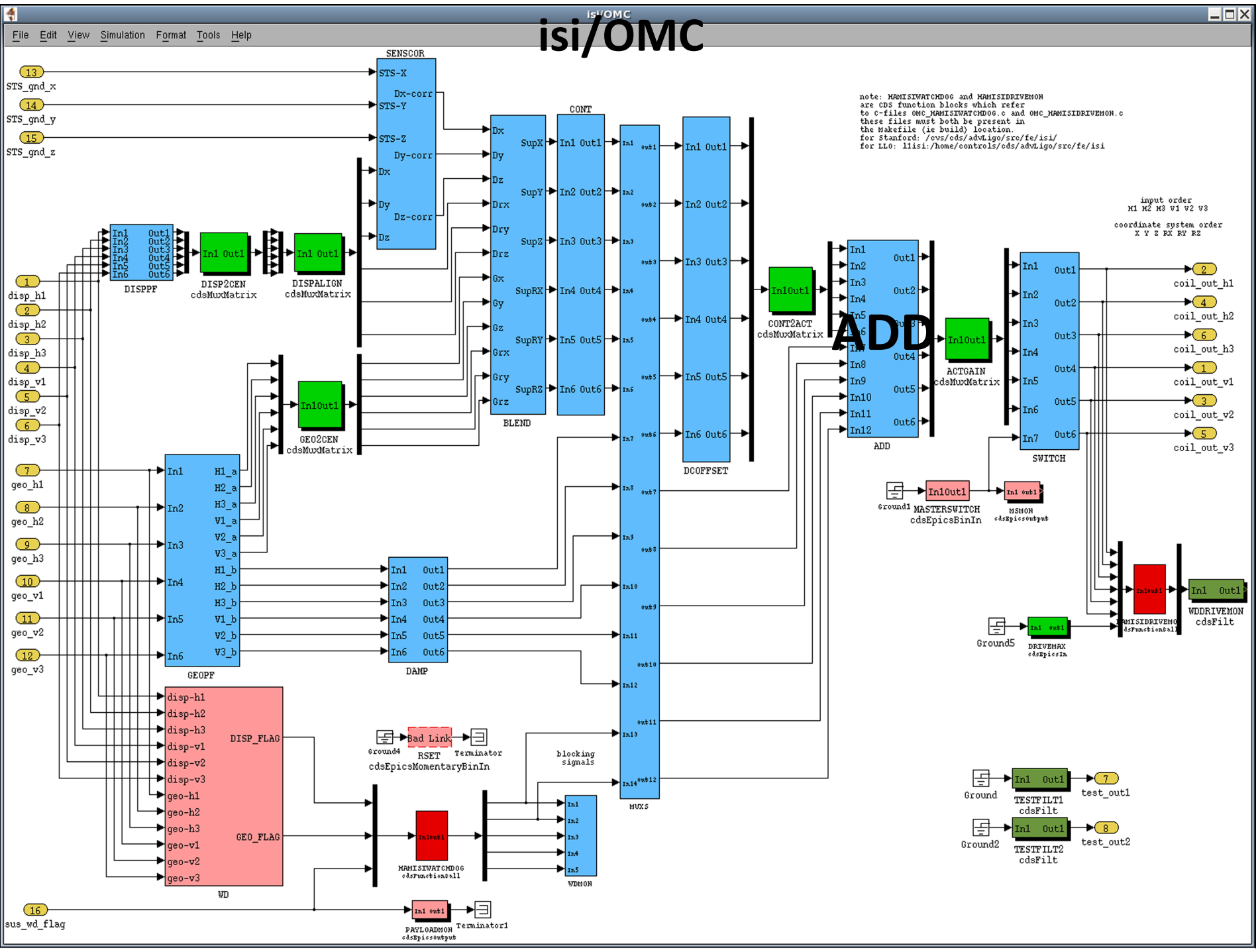
isi/OMC

File Edit View Simulation Format Tools Help





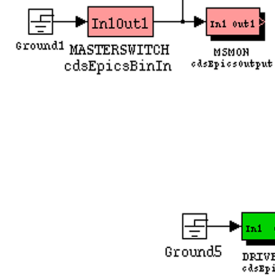
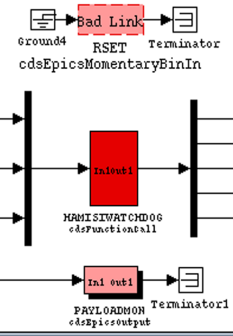
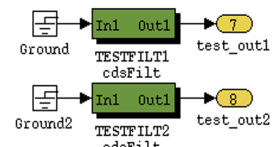
isi/OMC



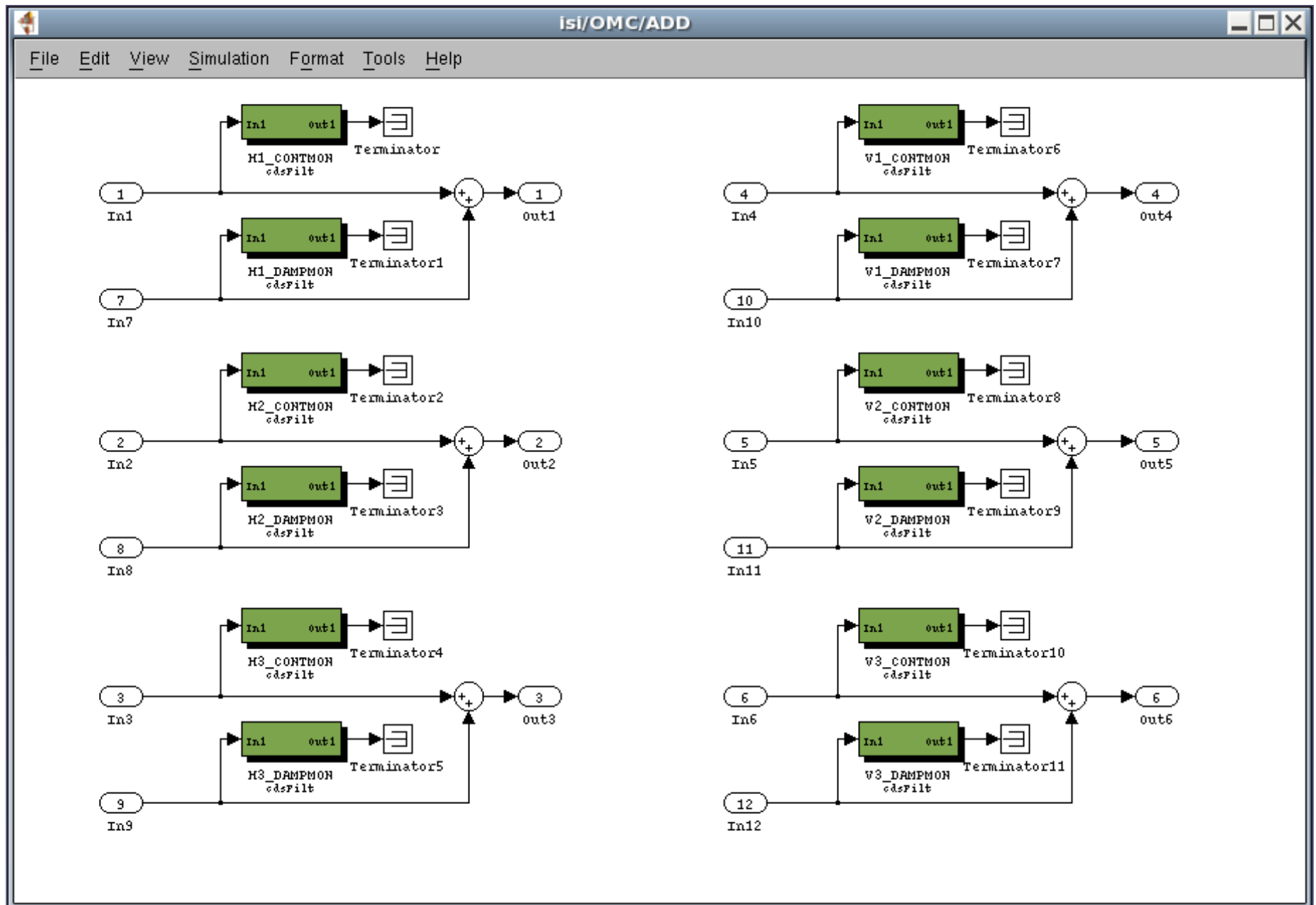
note: MAMISIVATCHD06 and MAMISIDRIVEMON are CDS function blocks which refer to c-files OMC_MAMISIVATCHD06.c and OMC_MAMISIDRIVEMON.c these files must both be present in the Makefile (ie build) location. for Stanford: /ovs/cds/adVligo/src/fe/isi/ for LL0: /llisi/home/controls/cds/adVligo/src/fe/isi

input order
M1 M2 M3 V1 V2 V3
coordinate system order
X Y Z RX RY RZ

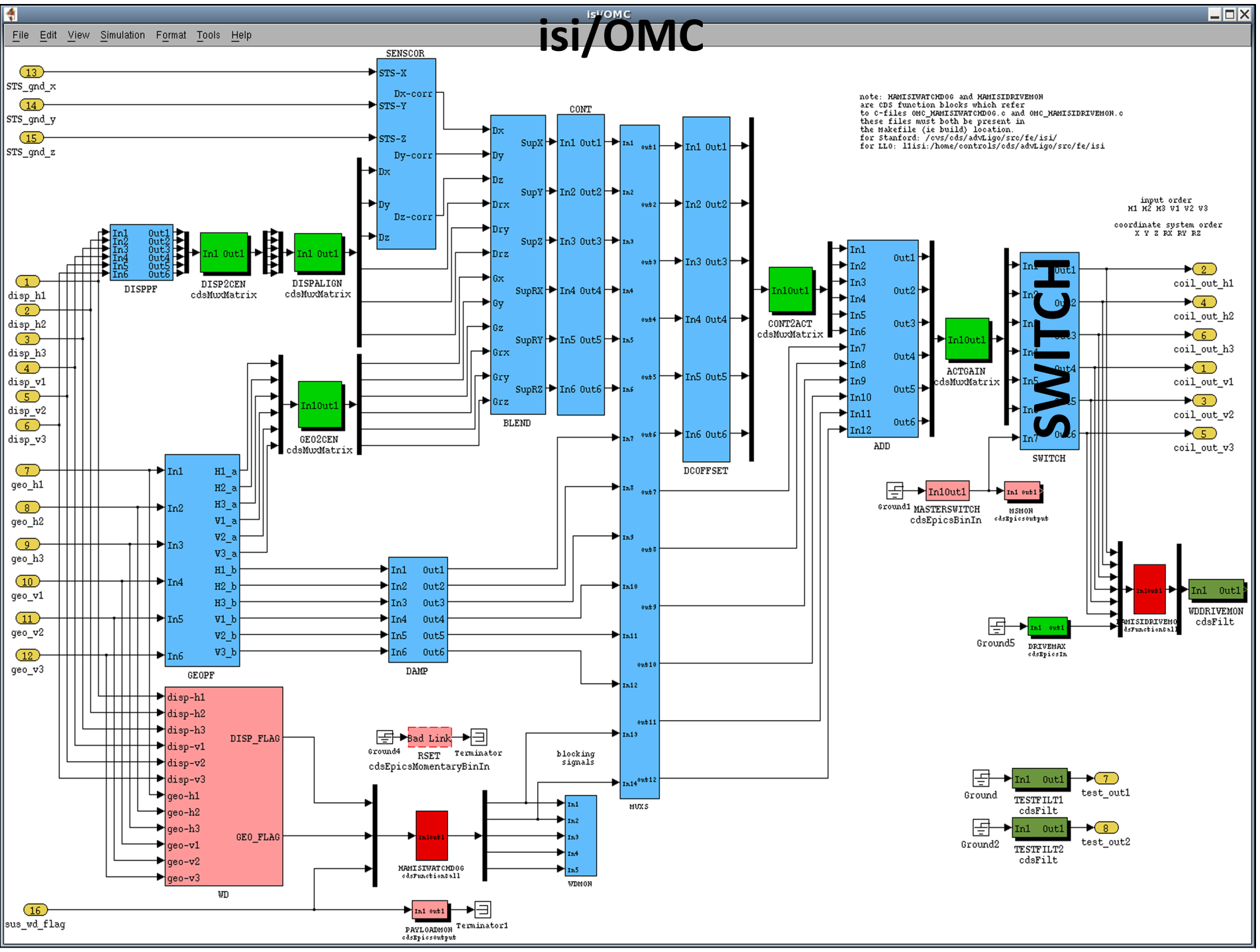
ADD



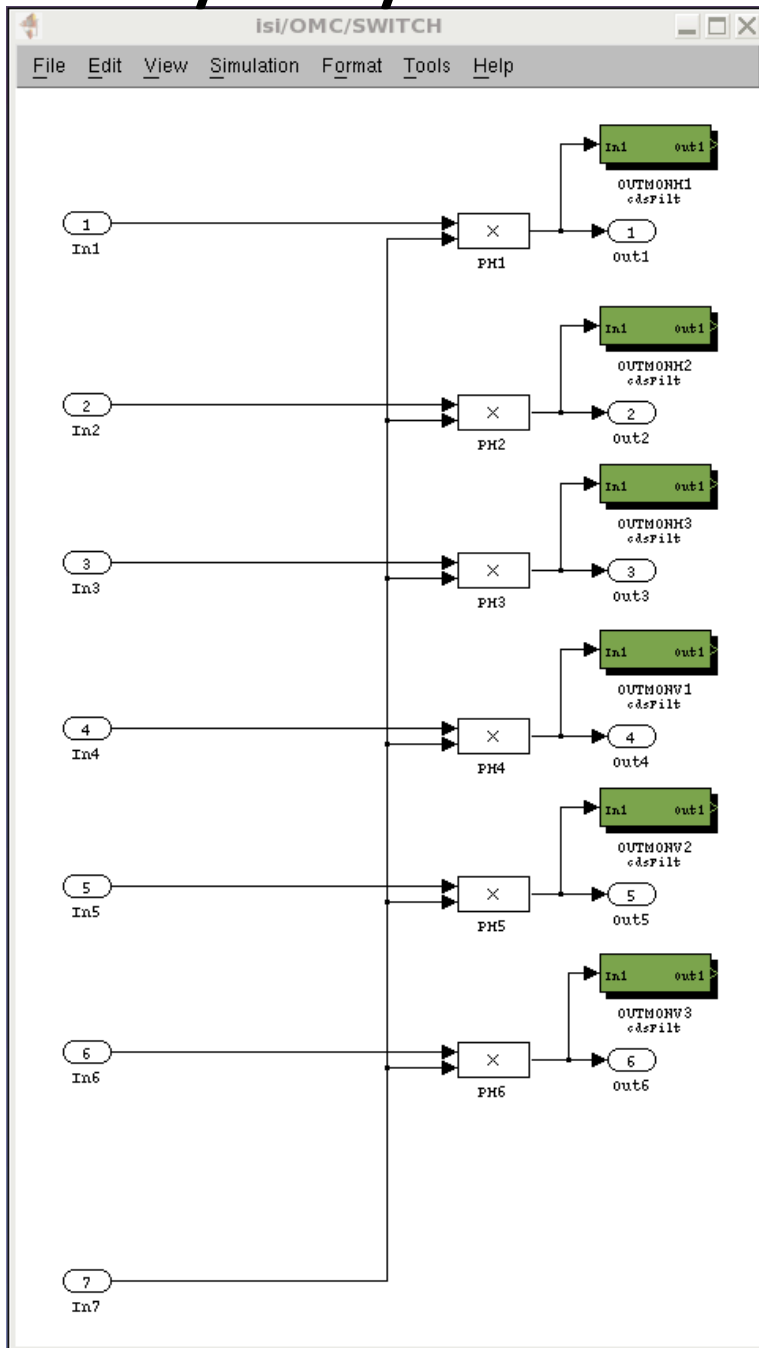
isi/OMC/ADD



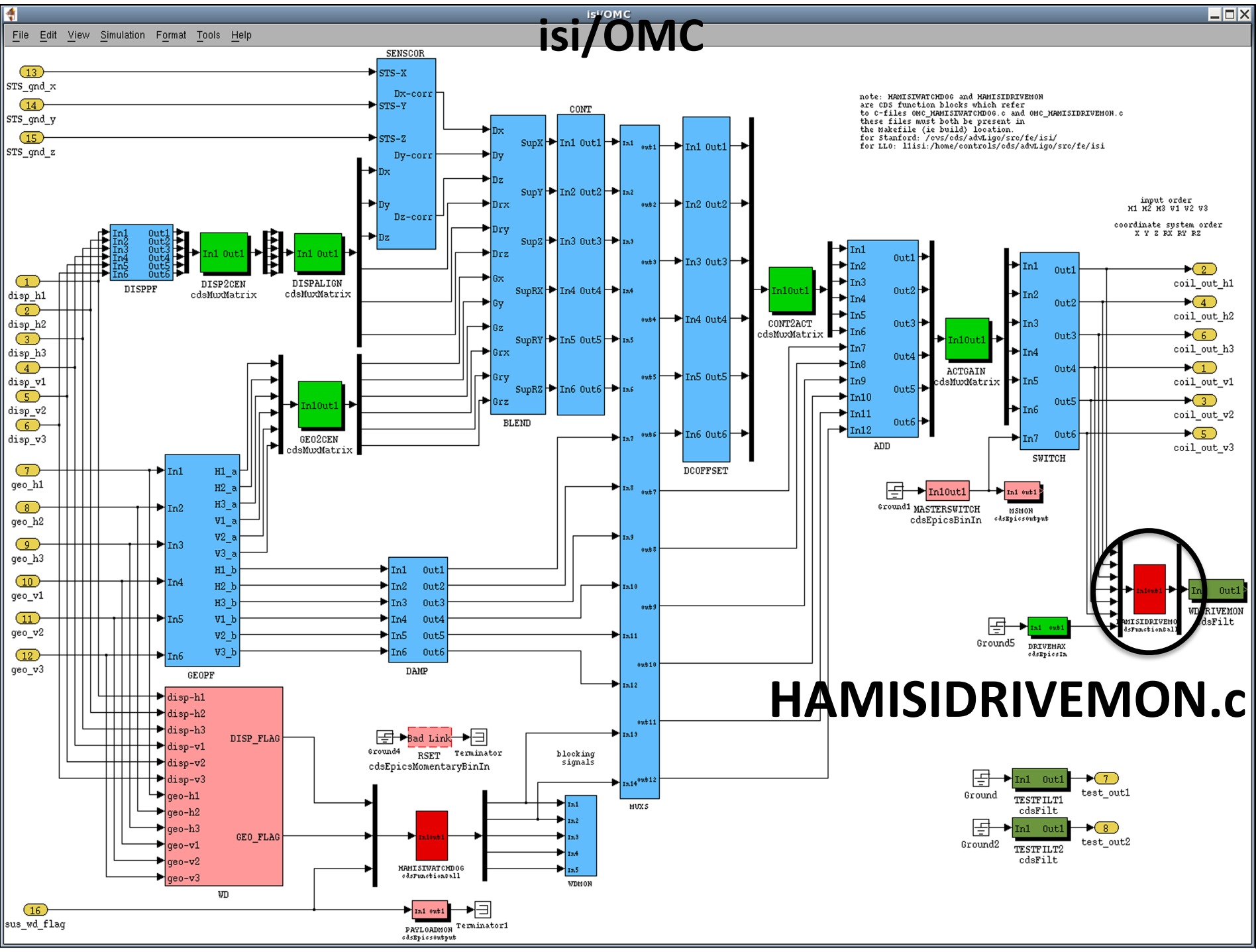
isi/OMC



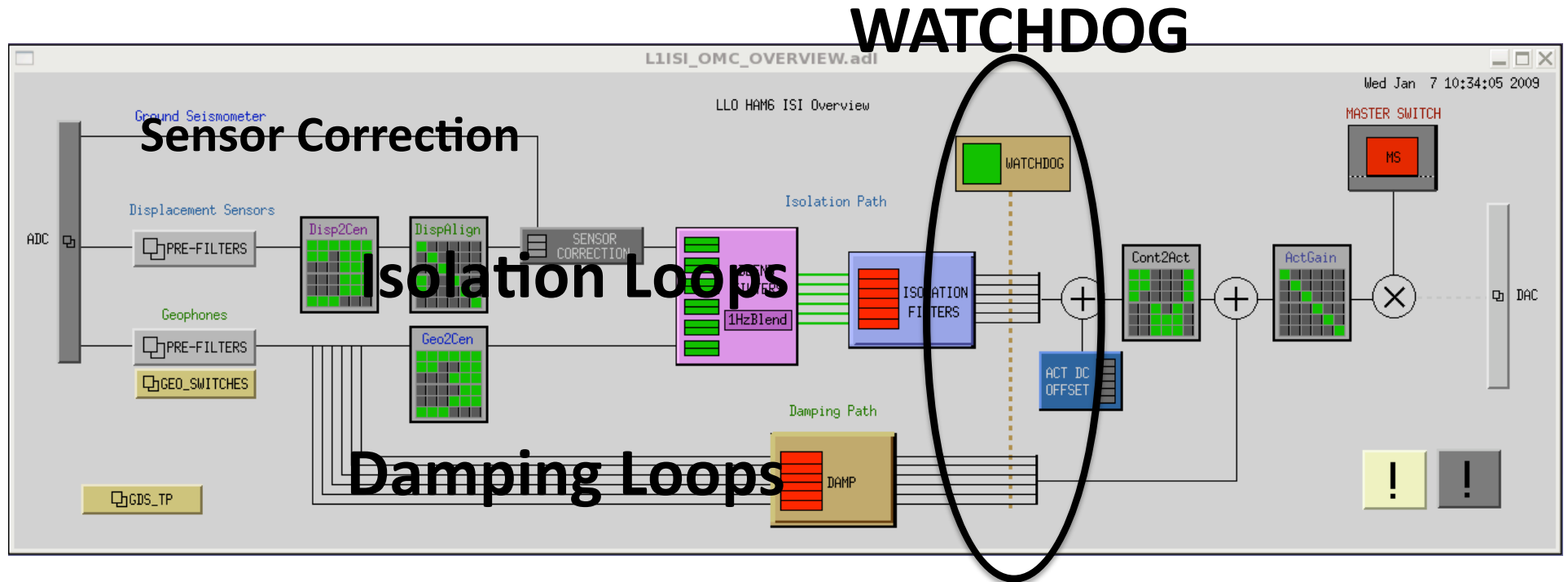
isi/OMC/SWITCH



isi/OMC



HAM ISI Overview MEDM Screen



HAM ISI Watchdog MEDM Screen

The screenshot displays a MEDM screen titled "LIISI_CUST_OMC_CTRL SWITCH_BUTTONS.adl". It features a table for limits and a section for watchdog states.

FIRST TRIG	CURRENT TRIG
DISP limit:	20000.0
GEO limit:	20000.0
ACT limit:	30000.0
PAYLOAD limit:	1000

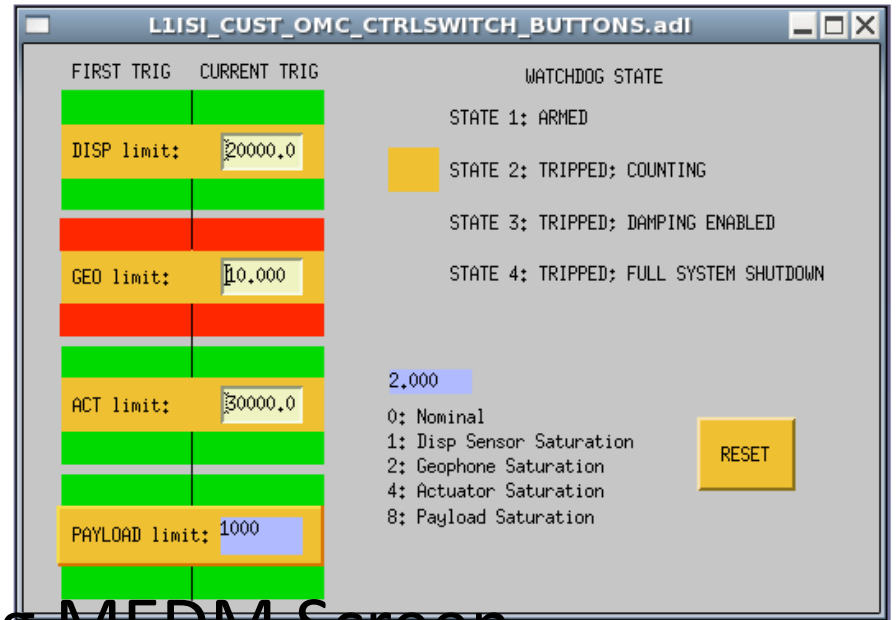
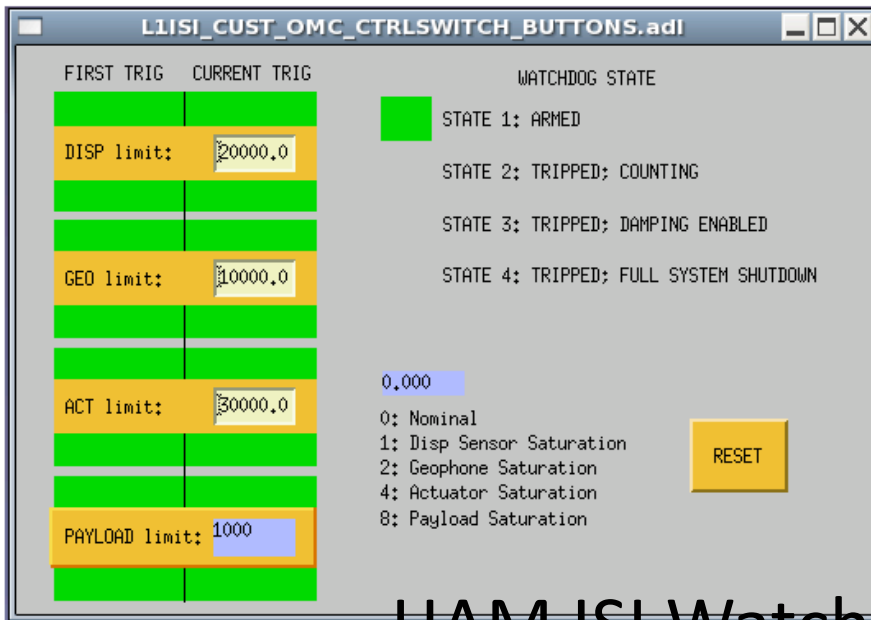
WATCHDOG STATE

- STATE 1: ARMED
- STATE 2: TRIPPED; COUNTING
- STATE 3: TRIPPED; DAMPING ENABLED
- STATE 4: TRIPPED; FULL SYSTEM SHUTDOWN

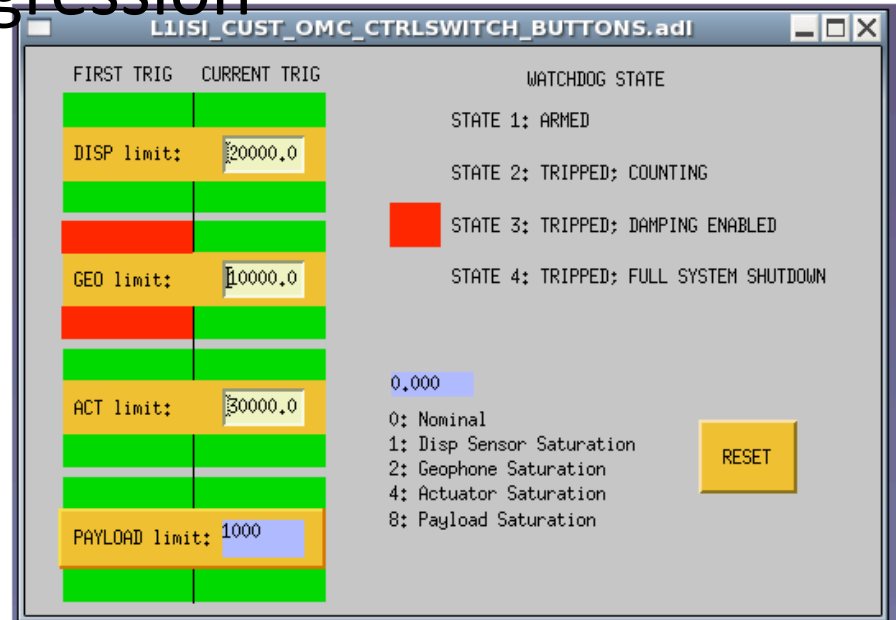
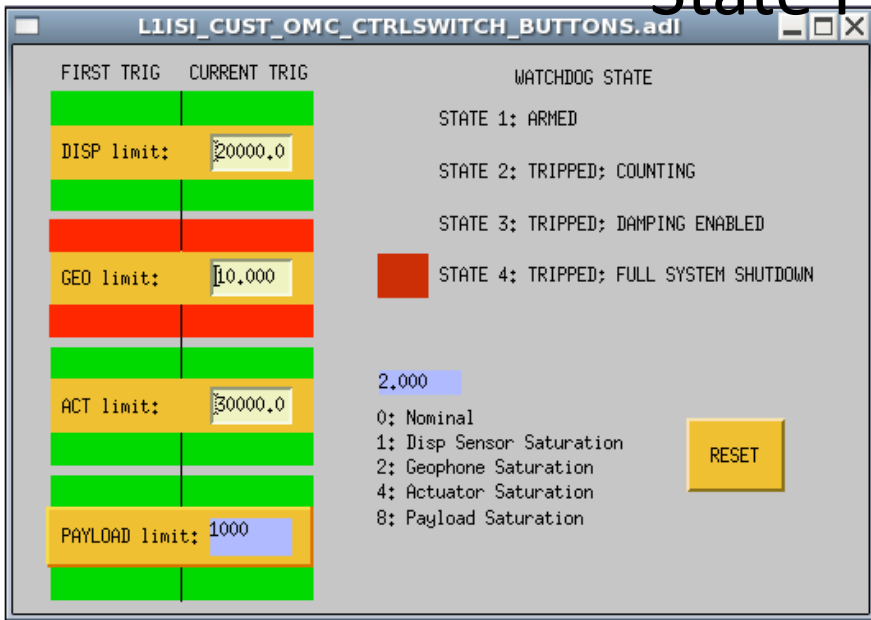
0.000

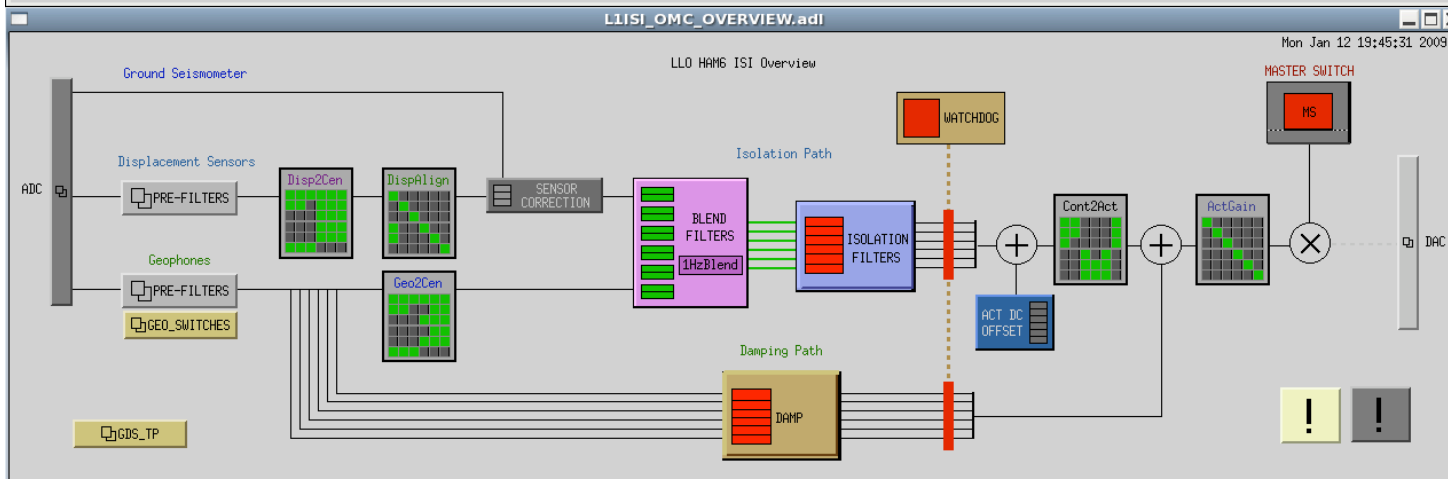
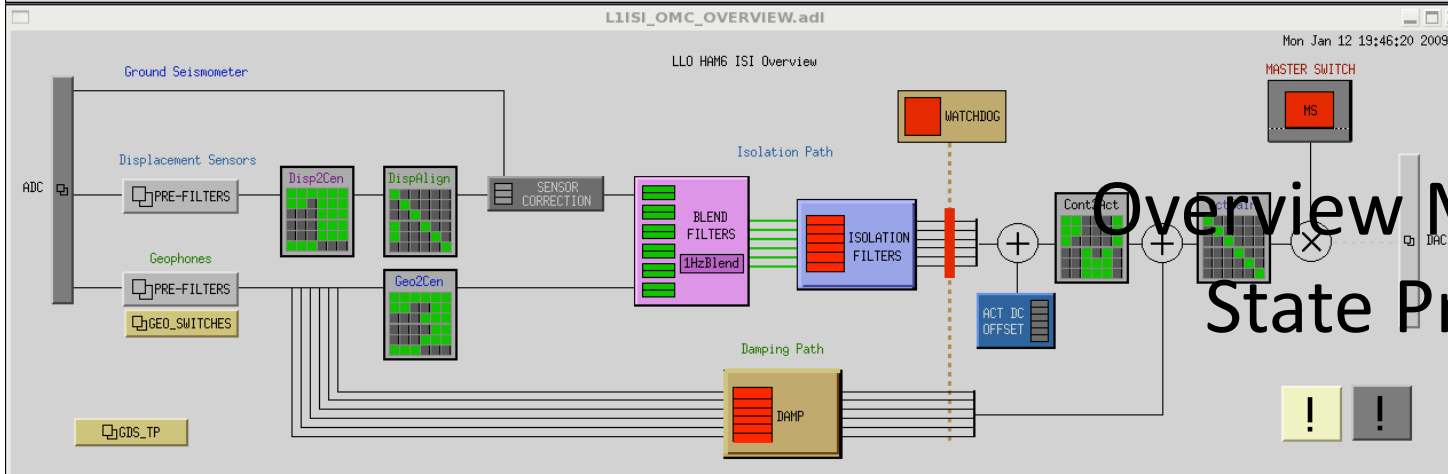
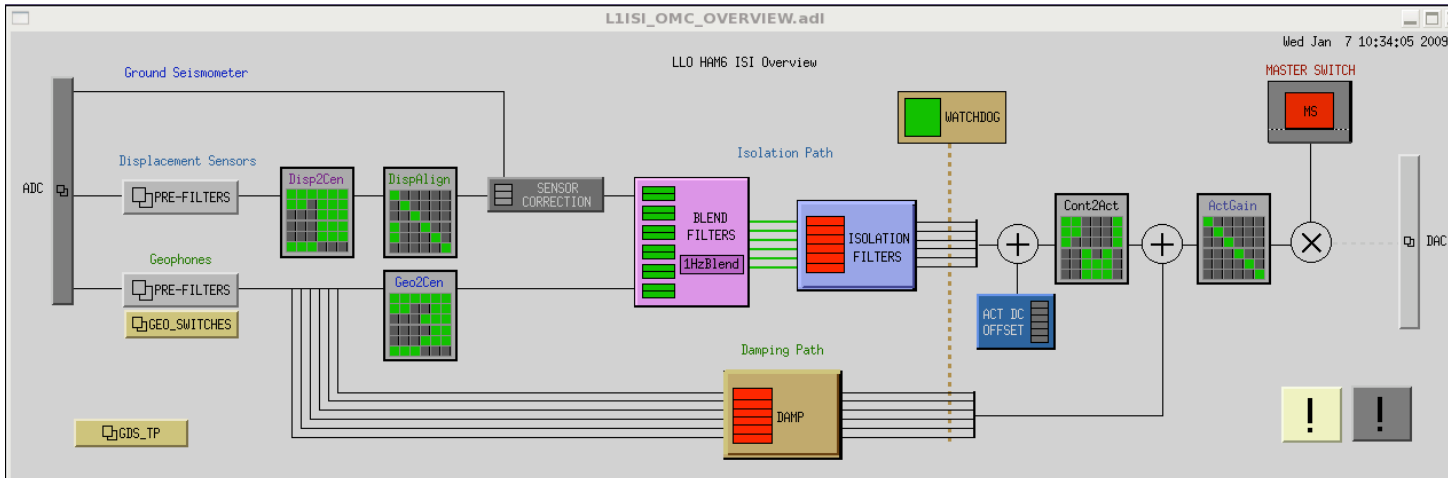
0: Nominal
1: Disp Sensor Saturation
2: Geophone Saturation
4: Actuator Saturation
8: Payload Saturation

RESET



HAM ISI Watchdog MEDM Screen State Progression





Overview MEDM Screen
State Progression

OMC_HAMISIWATCHDOG.c

```
1 /*
2 * OMC_HAMISIWATCHDOG.c
3 *
4 * HAMISIWATCHDOG is a state machine that watches the displacement
5 * sensor, geophone, actuator, and payload watchdog flags, and blocks
6 * the isolation loop path and/or damping loop path according to the
7 * flags.
8 *
9 *
10 * INPUTS:
11 * argin[0] = displacement sensor saturation flag
12 * argin[1] = geophone sensor saturation flag
13 * argin[2] = payload saturation flag
14 * global variables
15 *     actLevel      = actuator saturation flag (from OMC_HAMISIDRIVEMON.c)
16 *     pLocalEpics  = variable in control of CDS_EPICS (from EPICS database)
17 *
18 * OUTPUTS:
19 * argout[0] = flag to block damping loop path
20 * argout[1] = flag to block isolation loop path
21 * argout[2] = state of watchdog
22 * argout[3] = 4 character bit-field state after the first trigger
23 *             (1 = displacement sensors; 2 = geophones;
24 *             4 = actuators; 8 = payload;)
25 * argout[4] = triggers;
26 *
27 * Written by Jeff Kissel and Tobin Fricke
28 * Nov 26 2008
29 * $Id$
30 */
```

OMC_HAMISIWATCHDOG.c

```
31
32 void OMC_HAMISIWATCHDOG(double *argin, int nargin, double *argout, int nargs) {
33
34     static int state = 4;           // Start in STATE 4 (FULL SYSTEM SHUTDOWN)
35     static int firstTrigger = 0;    // Start with no indication of triggers
36     static int cycleClock = 0;      // Used to count cycles when in state 2.
37     //const int state2HoldCycles = 3 * FE_RATE; // Wait for [seconds] X (cycles/second)
38     const int state2HoldCycles = 3 * 2048; // Wait for [seconds] X (cycles/second)
39
40     int blockIsolationLoopsFlag = 1; // Start script with isolation path BLOCKED
41     int blockDampingLoopsFlag = 1;   // Start script with damping path BLOCKED
42
43
44     // Read inputs
45     int dispTriggered = argin[0];     // Displacement Sensor Flag
46     int geoTriggered = argin[1];     // Geophone flag
47     int payloadTriggered = argin[2]; // Payload flag
48     int actTriggered = actFlag;      // Actuator flag, defined in OMC_HAMISIDRIVEMON
49
50     // Check and record the RESET button value
51     int resetFlag = pLocalEpics->isi.OMC_RSET; // Epics variable for reset button
52     pLocalEpics->isi.OMC_RSET = 0;           // Set reset button epics variable to 0
53
54     // Check for triggers - make a 4 character bitfield so we can tell which triggers triggered
55     // The !! (double inverse) syntax guarantees that the given value is 0 or 1.
56     // The << is a bitwise shift left, i.e. 1 << 2 == 4 and 1 << 3 == 8.
57     // So, if the displacement sensors and actuators trigger, triggers = 0101;
58     int triggers =
59     (!!dispTriggered) | (!!geoTriggered << 1) | (!!actTriggered << 2) | (!!payloadTriggered << 3);
60
```


OMC_HAMISIWATCHDOG.c

```
61 // State transitions
62 switch (state) {
63 case 1: // To STATE 1 (ARMED)
64     if (triggers) {
65         firstTrigger = triggers; // Record the bit-field state after the first trigger
66         cycleClock = 0;
67         state = 2;
68     }
69     break;
70
71 case 2: // To STATE 2 (TRIGGERED; HOLDING 3 SECONDS WITH DAMPING ENABLED)
72     if (cycleClock >= state2HoldCycles) {
73         cycleClock = 0;
74         state = 3;
75         // If three seconds haven't passed yet, we just stay here in state 2.
76     }
77     break;
78
79 case 3: // To STATE 3 (TRIGGERED; MONITORING WITH DAMPING ENABLED)
80     if (triggers)
81         state = 4;
82     else if (resetFlag) {
83         firstTrigger = 0; // Reset first trigger
84         state = 1;
85     }
86     break;
87
88 default:
89 case 4: // To STATE 4 (TRIGGERED; FULL SHUTDOWN)
90     if (!triggers && resetFlag) {
91         firstTrigger = 0; // Reset first trigger
92         state = 1;
93     }
94     break;
95 }
```

OMC_HAMISIWATCHDOG.c

```
97 // State actions
98 switch (state) {
99 case 1: // STATE 1 (ARMED)
100     blockIsolationLoopsFlag = 0; // Leave isolation loop path open
101     blockDampingLoopsFlag = 0; // Leave damping loop path open
102     break;
103
104 case 2: // STATE 2 (TRIGGERED; HOLDING 3 SECONDS WITH DAMPING ENABLED)
105     blockIsolationLoopsFlag = 1; // Block isolation loops path
106     blockDampingLoopsFlag = 0; // Leave damping loops path open
107     cycleClock = cycleClock + 1; // Keep counting, it hasn't been long enough yet
108     break;
109
110 case 3: // STATE 3 (TRIGGERED; MONITORING WITH DAMPING ENABLED)
111     blockIsolationLoopsFlag = 1; // Block isolation loops path
112     blockDampingLoopsFlag = 0; // Leave damping loops path open
113     break;
114
115 default:
116 case 4: // STATE 4 (TRIGGERED; FULL SHUTDOWN)
117     blockIsolationLoopsFlag = 1; // Block isolation loops path
118     blockDampingLoopsFlag = 1; // Block damping loop path
119     break;
120
121 }
122
123 // Output
124 argout[0] = blockDampingLoopsFlag;
125 argout[1] = blockIsolationLoopsFlag;
126 argout[2] = state;
127 argout[3] = firstTrigger;
128 argout[4] = triggers;
129
130 return;
131 }
```

OMC_HAMISIDRIVEMON.c

```
1 /*
2  *   OMC_HAMISIDRIVEMON.c
3  *
4  *   INPUTS
5  *   mux3[0] = digital output to H1 actuator
6  *   mux3[1] = digital output to H2 actuator
7  *   mux3[2] = digital output to H3 actuator
8  *   mux3[3] = digital output to V1 actuator
9  *   mux3[4] = digital output to V2 actuator
10 *   mux3[5] = digital output to V3 actuator
11 *   mux3[6] = user defined threshold (DRIVEMAX)
12 *
13 *   OUTPUTS:
14 *   demux4[0] = saturation flag
15 *
16 *   actFlag (nominally 0) is set to 1 if any digital
17 *   signal exceeds user defined threshold. actFlag is used
18 *   for a flag read by OMC_HAMISIWATCHDOG.c.
19 *
20 *   Written by Jeff Kissel
21 *   Nov 17 2008
22 *   $Id$
23 */
```

OMC_HAMISIDRIVEMON.c

```
24
25 static int actFlag = 0; // must be static to persist between cycles,
26                       // in case WATCHDOG is called before DRIVEMON
27
28 void OMC_HAMISIDRIVEMON(double *argin, int nargin, double *argout, int nargsout) {
29
30     int numActuators = 6; // all but the last input
31     int userThreshold = argin[6]; // the last input
32
33     actFlag = 0; // no saturations yet
34
35     // look for saturations
36     int ii;
37     for (ii = 0; ii < numActuators; ii++)
38         // Set actuator flag to either
39         actFlag |= (argin[ii] >= userThreshold);
40
41     argout[0] = actFlag; //Send flag value to output
42 }
```

Auxiliary Scripts

(Not on front end)

- Checker – checks once a second whether FIRSTTRIG as non-zero value. If FIRSTTRIG is non-zero, calls ctrlDOWN
- ctrlDOWN – Turns OFF isolation loops immediately
 - Sets the gain ramp time to ZERO.
 - Turns the output OFF.
 - Turns the input OFF.
 - Turns the isolation loop boost filter OFF.
 - Turns the isolation loop gain to ZERO.
 - Clears the filter bank history.
 - Sets to the gain ramp time to five seconds.
- chk_daemon – cron job run every minute, makes sure checker is running.

Features to be Implemented

- “Heart beat” blinker on MEDM overview for checker script
- Auxiliary scripts to be moved onto dedicated script machine
- Triggers off of ground STS-2 saturation
- Operator alarm/beep after triggered
- Info for site overview screen
- Anything else?