GLAST The Gamma Ray Large Area Space Telescope

GLAST LAT ACD ELECTRONICS SUBSYSTEM TECHNICAL DOCUMENT	Document # ACD-PROC-000051	Date Effective 12-16-02
	Prepared by: D. A. Sheppard	Supersedes None
Document Title **GLAST LAT ACD FREE Comprehensive Performance Test**		

Gamma-ray Large Area Space Telescope (GLAST)
Large Area Telescope (LAT)
Anti-Coincidence Detector (ACD)

FREE (Front End & Event Electronics)

Electrical Integration Procedure (EIP)
and
Comprehensive Performance Test (CPT)

DRAFT

FREE Comprehensive Performance Test
Table of Contents

Document Change Log

Revision	Date	Change Description	Prepared by
Initial Release		Initial Release	Dave Sheppard

1.0 Scope and Purpose of this Procedure

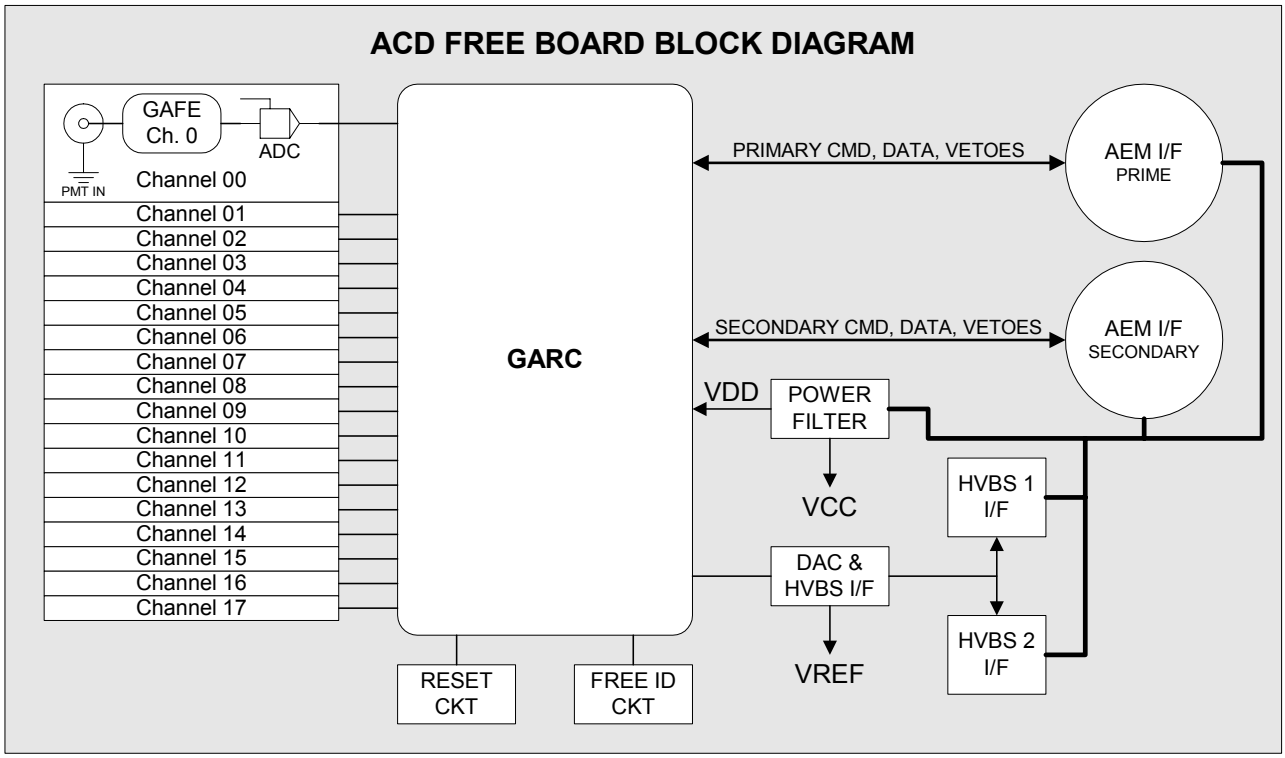
This document describes the instructions for electrical integration of the FREE electronics assembly to the LAT AEM (or AEM GSE) and details the test sequence of the comprehensive performance test (CPT). It does not duplicate all the functions of component-level testing, but instead concentrates on board-level verification issues. This procedure provides the instructions necessary to electrically integrate and functionally test the FREE assembly. This test is intended to be modified to provide other levels of functional testing as required during instrument and spacecraft level integrations.

The FREE assembly is designed to meet the electrical specifications of:

(1) The ACD Level IV Electronics Requirements, LAT-SS-00352
(2) The ACD-LAT Interface Control Document, LAT-SS-00-363

2.0 Description of the FREE Electronics Assembly

FREE is the acronym for the Gamma-Ray Large Area Space Telescope (GLAST) Anti-Coincidence Detector (ACD) Front End and Event Electronics processing board. The FREE provides the main electrical interface between the ACD and the LAT instrument electronics. The FREE circuitry provides command and data return functions for electronics boards associated with the ACD. There are 12 FREE boards in the flight system, with each FREE card supporting 18 phototube chains and 2 high voltage bias supplies (HVBS). The FREE circuitry utilizes a single +3.3V power supply and provides an interface to pass +28V power to the two HVBS. The FREE is a 10 layer flexible-rigid printed circuit board. It has 18 analog signal processing channels serviced by a single GARC readout controller. It also provides the interface to the HVBS and has two thermistors on board.



The design of the FREE circuitry was done utilizing Orcad for schematic capture, PSpice for simulation, and PADS for board layout. The current copy of the schematic package is posted in PDF format on the ACD website at: http://lhea-glast.gsfc.nasa.gov/acd/electronics/#free

3.0 Definitions, Terms, and Acronyms

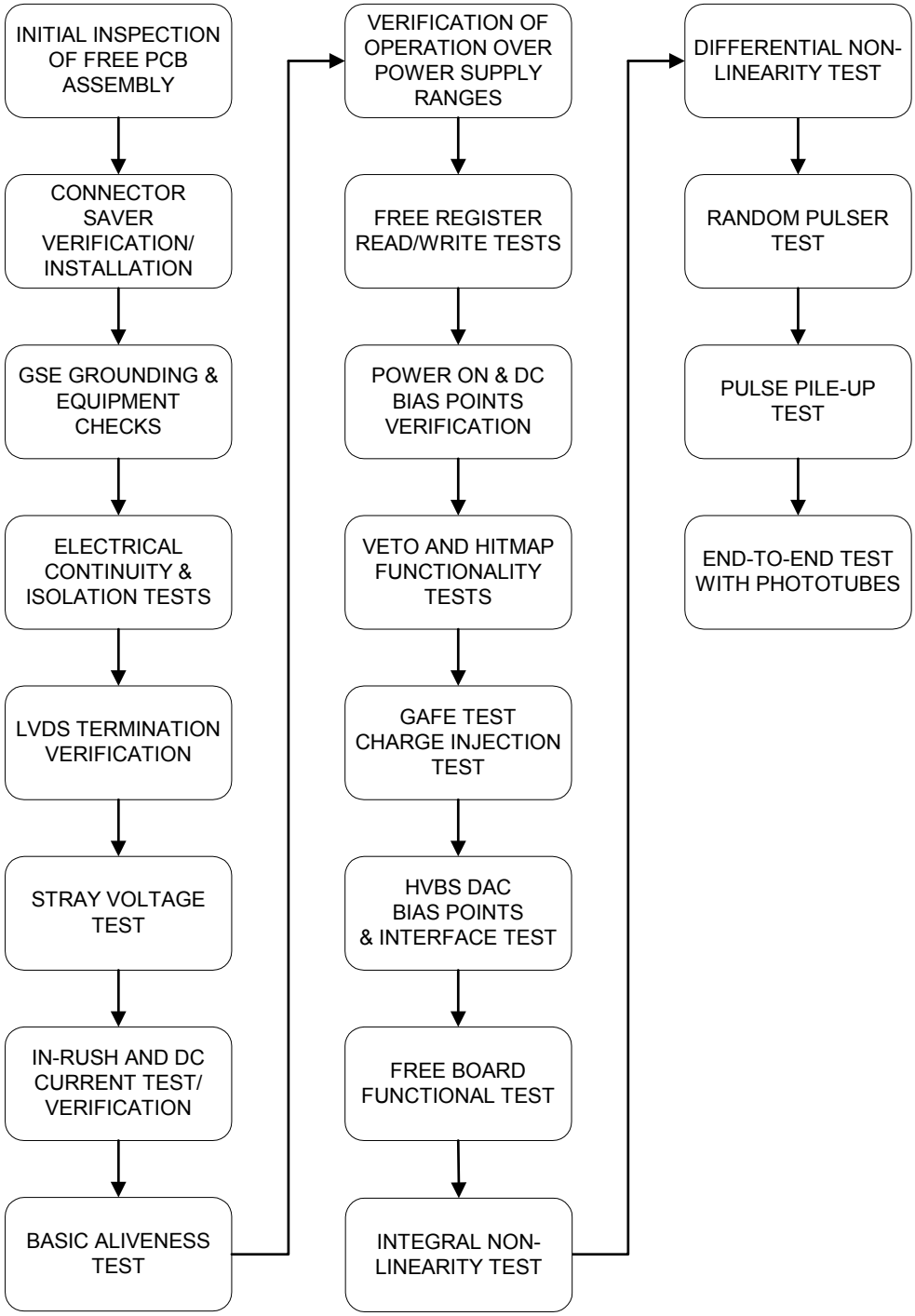
This can be found in any number of other GLAST documents and no one reads that stuff anyway, so it won't be repeated here. Hey – just to see if you were still reading!

4.0 Steps of the Electrical Integration and Performance Evaluation

The following block diagram details pictorially the sequence of events required to electrically integrate and test a FREE circuit card assembly with other LAT components.

<<PROBABLY NEED TO UPDATE THIS LIST AFTER FINAL ORDERING OF EVENTS>>

FREE ASSEMBLY INTEGRATION AND TEST FLOW



5.0 Preconditions to and Preparations for Starting this Test

Prior to starting this test, a responsible Test Conductor shall be named. The Test Conductor is responsible for the safety of the hardware and the documentation of results, including anomalies, for the duration of the test. The Test Conductor, or a designated representative, shall be present for each testing sequence. All testing on flight hardware will require a signed Work Authorization Order. Each person working directly with the ACD flight hardware shall be NASA certified for electrostatic discharge control as per NASA-STD-8739.7. All measurement equipment used for verification tests on flight hardware shall have a valid calibration sticker. The Test Conductor shall have the authority to determine deviations from this procedure and shall redline this procedure as necessary. Mates and demates to flight connectors shall be recorded. Connectors shall not be mated nor demated unless the instrument electronics is powered off. Connector savers shall be utilized wherever practical to minimize flight connector mate/demates.

5.1 FREE Assembly Identification

The test conductor for this test is: _____

The serial number of the FREE card assembly is: _____

The identification listed on the GARC ASIC is: _____

The identification listed on the GAFE ASICs is: _____

Note that for flight assemblies, the performance of this test must be listed on the appropriate Work Authorization Order (WOA).

5.2 Test Equipment Utilized

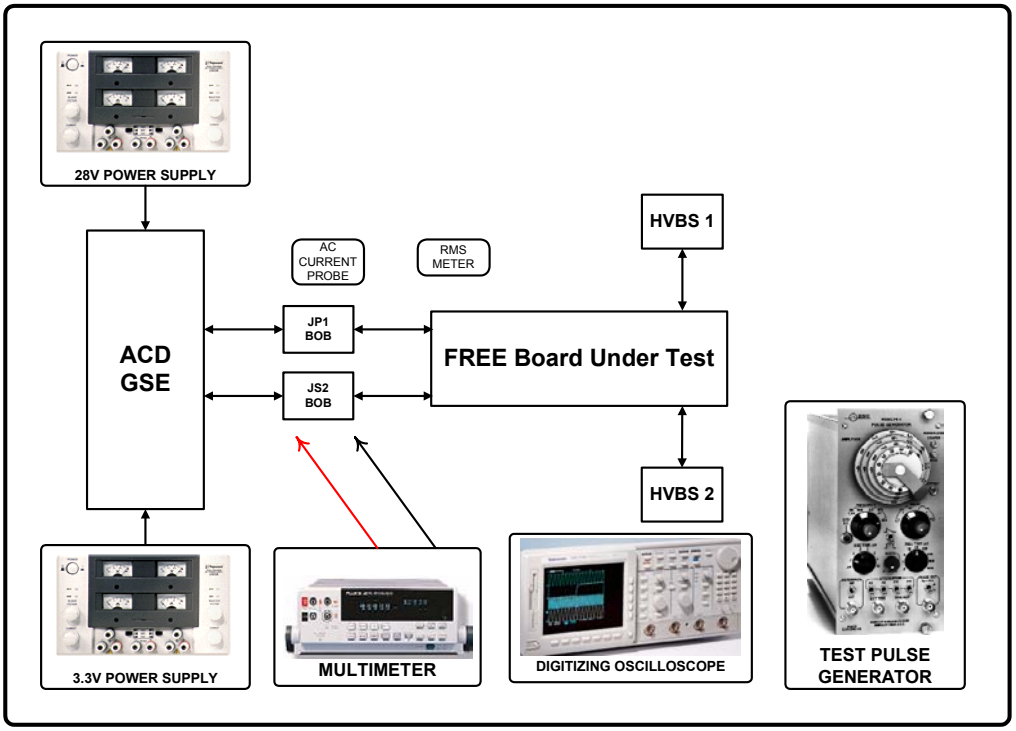
Prior to starting this test, the Test Conductor shall record the test and measurement hardware used in the performance of this test. Note that all multimeters used in testing of the FREE circuit card assembly shall be of low current output design, such as the Fluke 70 series, HP3400 series or similar. High current output multimeters are not compatible with the FREE circuitry.

Instrument Type	Manufacturer & Model Number	NASA ID Number	Last Calibrated Date	Calibration Due Date
Power Supply for HVBS +28V				
Power Supply for FREE +3.3V				
Multimeter 1				
Multimeter 2				
Test Pulse Generator				
Oscilloscope				
AC Current Probe				

RMS Voltmeter				

5.3 Suggested Equipment Setup

Prior to starting the test, it is recommended that the laboratory bench be set up as follows:



<< STILL WORKING ON THIS DIAGRAM >>

5.4 Verification of the GSE Ground

Prior to the connection of FREE assembly hardware to other electronics, it shall be verified that all power supplies, signal generators, VME racks, and any other test and measurement equipment shall be connected to the same AC ground. The simplest way to do this is to connect all AC-powered equipment to the same power strip. In cases where this is not practical (e.g., possibly a thermal-vacuum test), greater care must be taken to ensure there are no floating grounds since this would represent a hazard to the FREE assembly.

5.5 Description of the ACD Interface Signals

ACD_CLK: 20 MHz ± 1% continuous, 45-55% duty cycle clock from the ACD Electronics Module (AEM).

ACD_NSCMD: The command signal from AEM. The ACD_NSCMD signal transitions on the trailing edge of ACD_CLK and is shifted into the ACD on the leading edge of ACD_CLK. A single start bit, (logic 1), signals the beginning of a command.

ACD_NRST: Reset from AEM. ACD_NRST is synchronous to ACD_CLK. ACD_NRST at logic one resets state machines and initializes registers and modes in ACD. The ACD_NRST is at least five ACD_CLK cycles.

ACD_NSDATA: Data from ACD. The ACD_NSDATA signal transitions on the leading edge of ACD_CLK. The beginning of a data packet indicated by a single start bit.

ACD_NCNO: The ACD_NCNO interface signal is the OR of the selected (via command) HLD discriminators.

ACD_NVETOs: Veto discriminator output signals from ACD.

ACD_HV: Analog monitor of the high voltage power supply voltage output. 0 - 2.5 volts indicates 0 - full-scale volts at the supply output. Pseudo differential analog, with signal on ACD_HVP and ground on ACD_HVN, 10K ± 5% source impedance all lines.

ACD_TEMP: ACD board temperature monitor, 30K thermistor, GSFC S-311-P-18 series (YSI 44900 series).

ACD_VDD(0-2): +3.3V supplied by the AEM to the ACD FREE circuit cards.

ACD_28V(0-1): +28V supplied by the AEM to the ACD HVBSs.

6.0 Electrical Safe-to-Mate Verification of the FREE Assembly

The intent of this section is to ensure that the FREE assembly is both electrically and mechanically safe to mate to the AEM or AEM GSE to be used for testing.

Break Out Boxes (BOB) and a calibrated multimeter will be required for this test.

This section is to be performed prior to doing electrical testing on the FREE assembly. If this verification has been performed successfully, the Test Conductor may opt to omit this portion of the test if both the FREE assembly and the AEM interface have been checked.

1. Verify that connector savers have been installed on FREE connectors JP1, JS2, JHV1,and JHV2.
2. Verify that break out boxes (BOB) are available for these four connectors. These BOB should have been pin-for-pin checked for continuity and isolation prior to the start of this procedure.
3. The test conductor shall visually inspect each of the connector halves on the FREE assembly, BOB, harnessing, and AEM interface to ensure there are no bent pins, debris, or other physical damage that would impede safe connector mating.
4. All shorting pins should be removed from the BOBs.
5. Verify that the power supplies are OFF.
6. Mate the four BOBs to connectors JP1, JS2, JHV1 and JHV2 on the FREE assembly. Mate the other side of the BOBs to the AEM interface connectors.

6.1 Electrical Continuity Check

With the shorting plugs removed from the break out box, make the following continuity measurements on the FREE assembly side of the interface.

Measurement No.	Signal Pair	Pin 1	Pin 2	Expected Resistance	Verified
1	ACD_VDD	JP1 – 3	JP1 - 1	R < 1 Ω	
2	ACD_VDD	JP1 - 4	JP1 - 1	R < 1 Ω	
3	ACD_VDD	JS2 - 1	JP1 - 1	R < 1 Ω	
4	ACD_VDD	JS2 – 3	JP1 - 1	R < 1 Ω	
5	ACD_VDD	JS2 - 4	JP1 - 1	R < 1 Ω	
6	ACD_RTN	JP1-31	JP1-30	R < 1 Ω	
7	ACD_RTN	JP1-32	JP1-30	R < 1 Ω	
8	ACD_RTN	JS2-30	JP1-30	R < 1 Ω	
9	ACD_RTN	JS2-31	JP1-30	R < 1 Ω	
10	ACD_RTN	JS2-32	JP1-30	R < 1 Ω	
11	ACD_28V	JP1-7	JP1-5	R < 1 Ω	
12	ACD_28V	JS2-5	JP1-5	R < 1 Ω	
13	ACD_28V	JS2-7	JP1-5	R < 1 Ω	
14	ACD_28V_RTN	JP1-34	JP1-33	R < 1 Ω	
15	ACD_28V_RTN	JS2-33	JP1-33	R < 1 Ω	
16	ACD_28V_RTN	JS2-34	JP1-33	R < 1 Ω	

6.2 Electrical Isolation Check

With the shorting plugs removed from the break out box, make the following isolation measurements on the FREE assembly side of the interface.

Meas. No.	Signal 1	Signal 2	Pin 1	Pin 2	Expected Resistance	Verified
1	ACD_VDD	ACD_RTN	JP1 – 1	JP1 - 30	R > 1 kΩ	
2	ACD_VDD	ACD_28V	JP1 – 1	JP1 - 5	R > 1 MΩ	
3	ACD_VDD	ACD_28V_RTN	JP1 – 1	JP1 - 33	R > 1 MΩ	
4	ACD_28V	ACD_RTN	JP1 - 5	JP1 - 30	R > 1 MΩ	
5	ACD_28V	ACD_28V_RTN	JP1 – 5	JP1 - 33	R > 1 kΩ	
6	ACD_RTN	ACD_28V_RTN	JS2 - 30	JP1 - 33	R > 90 Ω	
7	ACD_NVETO_00AP	ACD_NVETO_00AM	JP1-70	JP1-71	R > 100 kΩ	
8	ACD_NVETO_01AP	ACD_NVETO_01AM	JP1-68	JP1-69	R > 100 kΩ	
9	ACD_NVETO_02AP	ACD_NVETO_02AM	JP1-66	JP1-67	R > 100 kΩ	
10	ACD_NVETO_03AP	ACD_NVETO_03AM	JP1-64	JP1-65	R > 100 kΩ	
11	ACD_NVETO_04AP	ACD_NVETO_04AM	JP1-62	JP1-63	R > 100 kΩ	
12	ACD_NVETO_05AP	ACD_NVETO_05AM	JP1-60	JP1-61	R > 100 kΩ	
13	ACD_NVETO_06AP	ACD_NVETO_06AM	JP1-58	JP1-59	R > 100 kΩ	
14	ACD_NVETO_07AP	ACD_NVETO_07AM	JP1-56	JP1-57	R > 100 kΩ	
15	ACD_NVETO_08AP	ACD_NVETO_08AM	JP1-54	JP1-55	R > 100 kΩ	
16	ACD_NVETO_09AP	ACD_NVETO_09AM	JP1-52	JP1-53	R > 100 kΩ	
17	ACD_NVETO_10AP	ACD_NVETO_10AM	JP1-50	JP1-51	R > 100 kΩ	
18	ACD_NVETO_11AP	ACD_NVETO_11AM	JP1-48	JP1-49	R > 100 kΩ	
19	ACD_NVETO_12AP	ACD_NVETO_12AM	JP1-46	JP1-47	R > 100 kΩ	
20	ACD_NVETO_13AP	ACD_NVETO_13AM	JP1-44	JP1-45	R > 100 kΩ	
21	ACD_NVETO_14AP	ACD_NVETO_14AM	JP1-42	JP1-43	R > 100 kΩ	

22	ACD_NVETO_15AP	ACD_NVETO_15AM	JP1-40	JP1-41	R > 100 kΩ	
23	ACD_NVETO_16AP	ACD_NVETO_16AM	JP1-17	JP1-18	R > 100 kΩ	
24	ACD_NVETO_17AP	ACD_NVETO_17AM	JP1-19	JP1-20	R > 100 kΩ	
25	ACD_NCNO_AP	ACD_NCNO_AM	JP1-21	JP1-22	R > 100 kΩ	
26	ACD_NSDATA_AP	ACD_NSDATA_AM	JP1-72	JP1-73	R > 100 kΩ	
27	ACD_NRST_AP	ACD_NTST_AM	JP1-74	JP1-75	R > 100 kΩ	
28	ACD_NSCMD_AP	ACD_NSCMD_AM	JP1-76	JP1-77	R > 100 kΩ	
29	ACD_NSCLK_AP	ACD_NSCLK_AM	JP1-78	JP1-79	R > 100 kΩ	
30	ACD_HV_AP	ACD_HV_AM	JP1-23	JP1-24	R > 1 kΩ	
31	ACD_TEMP_AP	ACD_TEMP_AM	JP1-25	JP1-26	R > 20 kΩ	
32	ACD_NVETO_00BP	ACD_NVETO_00BM	JS2-70	JS2-71	R > 100 kΩ	
33	ACD_NVETO_01BP	ACD_NVETO_01BM	JS2-68	JS2-69	R > 100 kΩ	
34	ACD_NVETO_02BP	ACD_NVETO_02BM	JS2-66	JS2-67	R > 100 kΩ	
35	ACD_NVETO_03BP	ACD_NVETO_03BM	JS2-64	JS2-65	R > 100 kΩ	
36	ACD_NVETO_04BP	ACD_NVETO_04BM	JS2-62	JS2-63	R > 100 kΩ	
37	ACD_NVETO_05BP	ACD_NVETO_05BM	JS2-60	JS2-61	R > 100 kΩ	
38	ACD_NVETO_06BP	ACD_NVETO_06BM	JS2-58	JS2-59	R > 100 kΩ	
39	ACD_NVETO_07BP	ACD_NVETO_07BM	JS2-56	JS2-57	R > 100 kΩ	
40	ACD_NVETO_08BP	ACD_NVETO_08BM	JS2-54	JS2-55	R > 100 kΩ	
41	ACD_NVETO_09BP	ACD_NVETO_09BM	JS2-52	JS2-53	R > 100 kΩ	
42	ACD_NVETO_10BP	ACD_NVETO_10BM	JS2-50	JS2-51	R > 100 kΩ	
43	ACD_NVETO_11BP	ACD_NVETO_11BM	JS2-48	JS2-49	R > 100 kΩ	
44	ACD_NVETO_12BP	ACD_NVETO_12BM	JS2-46	JS2-47	R > 100 kΩ	
45	ACD_NVETO_13BP	ACD_NVETO_13BM	JS2-44	JS2-45	R > 100 kΩ	
46	ACD_NVETO_14BP	ACD_NVETO_14BM	JS2-42	JS2-43	R > 100 kΩ	
47	ACD_NVETO_15BP	ACD_NVETO_15BM	JS2-40	JS2-41	R > 100 kΩ	
48	ACD_NVETO_16BP	ACD_NVETO_16BM	JS2-17	JS2-18	R > 100 kΩ	
49	ACD_NVETO_17BP	ACD_NVETO_17BM	JS2-19	JS2-20	R > 100 kΩ	
50	ACD_NCNO_BP	ACD_NCNO_BM	JS2-21	JS2-22	R > 100 kΩ	
51	ACD_NSDATA_BP	ACD_NSDATA_BM	JS2-72	JS2-73	R > 100 kΩ	
52	ACD_NRST_BP	ACD_NTST_BM	JS2-74	JS2-75	R > 100 kΩ	
53	ACD_NSCMD_BP	ACD_NSCMD_BM	JS2-76	JS2-77	R > 100 kΩ	
54	ACD_NSCLK_BP	ACD_NSCLK_BM	JS2-78	JS2-79	R > 100 kΩ	
55	ACD_HV_BP	ACD_HV_BM	JS2-23	JS2-24	R > 1 kΩ	
56	ACD_TEMP_BP	ACD_TEMP_BM	JS2-25	JS2-26	R > 20 kΩ	

6.3 Verification of Proper LVDS Terminations

The ACD interface requires proper termination of the LVDS drivers. The specification is 100 Ω +/- 5%. This test is performed with the ACD power off. A multimeter is used to measure the impedance across the following signal pairs and the result is recorded

Measurement No.	Signal Pair	Pin 1	Pin 2	Expected Resistance	Verified
1	ACD_NVETO_16_A	JP1 - 17	JP1 - 18	95 Ω < R < 105 Ω	
2	ACD_NVETO_17_A	JP1 - 19	JP1 - 20	95 Ω < R < 105 Ω	
3	ACD_CNO_A	JP1 - 21	JP1 - 22	95 Ω < R < 105 Ω	
4	ACD_NVETO_15_A	JP1 - 40	JP1 - 41	95 Ω < R < 105 Ω	
5	ACD_NVETO_14_A	JP1 - 42	JP1 - 43	95 Ω < R < 105 Ω	
6	ACD_NVETO_13_A	JP1 - 44	JP1 - 45	95 Ω < R < 105 Ω	
7	ACD_NVETO_12_A	JP1 - 46	JP1 - 47	95 Ω < R < 105 Ω	
8	ACD_NVETO_11_A	JP1 - 48	JP1 - 49	95 Ω < R < 105 Ω	

9	ACD_NVETO_10_A	JP1 - 50	JP1 - 51	95 Ω < R < 105 Ω	
10	ACD_NVETO_09_A	JP1 - 52	JP1 - 53	95 Ω < R < 105 Ω	
11	ACD_NVETO_08_A	JP1 - 54	JP1 - 55	95 Ω < R < 105 Ω	
12	ACD_NVETO_07_A	JP1 - 56	JP1 - 57	95 Ω < R < 105 Ω	
13	ACD_NVETO_06_A	JP1 - 58	JP1 - 59	95 Ω < R < 105 Ω	
14	ACD_NVETO_05_A	JP1 - 60	JP1 - 61	95 Ω < R < 105 Ω	
15	ACD_NVETO_04_A	JP1 - 62	JP1 - 63	95 Ω < R < 105 Ω	
16	ACD_NVETO_03_A	JP1 - 64	JP1 - 65	95 Ω < R < 105 Ω	
17	ACD_NVETO_02_A	JP1 - 66	JP1 - 67	95 Ω < R < 105 Ω	
18	ACD_NVETO_01_A	JP1 - 68	JP1 - 69	95 Ω < R < 105 Ω	
19	ACD_NVETO_00_A	JP1 - 70	JP1 - 71	95 Ω < R < 105 Ω	
20	ACD_NSDATA_A	JP1 - 72	JP1 - 73	95 Ω < R < 105 Ω	
21	ACD_NRST_A	JP1 - 74	JP1 - 75	95 Ω < R < 105 Ω	
22	ACD_NSCMD_A	JP1 - 76	JP1 - 77	95 Ω < R < 105 Ω	
23	ACD_CLK_A	JP1 - 78	JP1 - 79	95 Ω < R < 105 Ω	
24	ACD_NVETO_16_B	JS2 - 17	JS2 - 18	95 Ω < R < 105 Ω	
25	ACD_NVETO_17_B	JS2 - 19	JS2 - 20	95 Ω < R < 105 Ω	
26	ACD_CNO_B	JS2 - 21	JS2 - 22	95 Ω < R < 105 Ω	
27	ACD_NVETO_15_B	JS2 - 40	JS2 - 41	95 Ω < R < 105 Ω	
28	ACD_NVETO_14_B	JS2 - 42	JS2 - 43	95 Ω < R < 105 Ω	
29	ACD_NVETO_13_B	JS2 - 44	JS2 - 45	95 Ω < R < 105 Ω	
30	ACD_NVETO_12_B	JS2 - 46	JS2 - 47	95 Ω < R < 105 Ω	
31	ACD_NVETO_11_B	JS2 - 48	JS2 - 49	95 Ω < R < 105 Ω	
32	ACD_NVETO_10_B	JS2 - 50	JS2 - 51	95 Ω < R < 105 Ω	
33	ACD_NVETO_09_B	JS2 - 52	JS2 - 53	95 Ω < R < 105 Ω	
34	ACD_NVETO_08_B	JS2 - 54	JS2 - 55	95 Ω < R < 105 Ω	
35	ACD_NVETO_07_B	JS2 - 56	JS2 - 57	95 Ω < R < 105 Ω	
36	ACD_NVETO_06_B	JS2 - 58	JS2 - 59	95 Ω < R < 105 Ω	
37	ACD_NVETO_05_B	JS2 - 60	JS2 - 61	95 Ω < R < 105 Ω	
38	ACD_NVETO_04_B	JS2 - 62	JS2 - 63	95 Ω < R < 105 Ω	
39	ACD_NVETO_03_B	JS2 - 64	JS2 - 65	95 Ω < R < 105 Ω	
40	ACD_NVETO_02_B	JS2 - 66	JS2 - 67	95 Ω < R < 105 Ω	
41	ACD_NVETO_01_B	JS2 - 68	JS2 - 69	95 Ω < R < 105 Ω	
42	ACD_NVETO_00_B	JS2 - 70	JS2 - 71	95 Ω < R < 105 Ω	
43	ACD_NSDATA_B	JS2 - 72	JS2 - 73	95 Ω < R < 105 Ω	
44	ACD_NRST_B	JS2 - 74	JS2 - 75	95 Ω < R < 105 Ω	
45	ACD_NSCMD_B	JS2 - 76	JS2 - 77	95 Ω < R < 105 Ω	
46	ACD_CLK_B	JS2 - 78	JS2 - 79	95 Ω < R < 105 Ω	

7.0 Initial Power-On Measurements

With the shorting plugs removed from the break out boxes, turn the ACD power on and perform the following voltage

7.1 Stray Voltage Test at the AEM Interface

With the shorting plugs removed from the break out boxes, turn the ACD power on and perform the following voltage measurements on the AEM interface side of the break out box. The (-) input to the voltmeter may be referenced to pin 30, the +3.3V return.

Meas. No.	AEM Interface Pin	Signal Name	Expected Voltage	Measured Voltage
1	JP1 - 1	ACD_VDD_0A	+3.3V	
2	JP1 - 3	ACD_VDD_1A	+3.3V	
3	JP1 - 4	ACD_VDD_2A	+3.3V	
4	JP1 – 5	ACD_28V_0A	+28V	
5	JP1 - 7	ACD_28V_1A	+28V	
6	JP1 - 17	ACD_NVETO_16AP	0< V < 3.3	
7	JP1 - 18	ACD_NVETO_16AM	0< V < 3.3	
8	JP1 - 19	ACD_NVETO_17AP	0< V < 3.3	
9	JP1 - 20	ACD_NVETO_17AM	0< V < 3.3	
10	JP1 - 21	ACD_NCNO_AP	0< V < 3.3	
11	JP1 - 22	ACD_NCNO_AM	0< V < 3.3	
12	JP1 - 23	ACD_HV_AP	0< V < 3.3	
13	JP1 - 24	ACD_HV_AM	0< V < 3.3	
14	JP1 - 25	ACD_TEMP_AP	TBD	
15	JP1 - 26	ACD_TEMP_AM	TBD	
16	JP1 - 30	ACD_GND_0A	0	
17	JP1 - 31	ACD_GND_1A	0	
18	JP1 - 32	ACD_GND_2A	0	
19	JP1 - 33	ACD_28V_RTN_0A	0	
20	JP1 - 34	ACD_28V_RTN_1A	0	
21	JP1 - 40	ACD_NVETO_15AM	0< V < 3.3	
22	JP1 - 41	ACD_NVETO_15AP	0< V < 3.3	
23	JP1 - 42	ACD_NVETO_14AM	0< V < 3.3	
24	JP1 - 43	ACD_NVETO_14AP	0< V < 3.3	
25	JP1 - 44	ACD_NVETO_13AM	0< V < 3.3	
26	JP1 - 45	ACD_NVETO_13AP	0< V < 3.3	
27	JP1 - 46	ACD_NVETO_12AM	0< V < 3.3	
28	JP1 - 47	ACD_NVETO_12AP	0< V < 3.3	
29	JP1 - 48	ACD_NVETO_11AM	0< V < 3.3	
30	JP1 - 49	ACD_NVETO_11AP	0< V < 3.3	
31	JP1 - 50	ACD_NVETO_10AM	0< V < 3.3	
32	JP1 - 51	ACD_NVETO_10AP	0< V < 3.3	
33	JP1 - 52	ACD_NVETO_09AM	0< V < 3.3	
34	JP1 - 53	ACD_NVETO_09AP	0< V < 3.3	
35	JP1 - 54	ACD_NVETO_08AM	0< V < 3.3	
36	JP1 - 55	ACD_NVETO_08AP	0< V < 3.3	
37	JP1 - 56	ACD_NVETO_07AM	0< V < 3.3	
38	JP1 - 57	ACD_NVETO_07AP	0< V < 3.3	

39	JP1 - 58	ACD_NVETO_06AM	0< V < 3.3	
40	JP1 - 59	ACD_NVETO_06AP	0< V < 3.3	
41	JP1 - 60	ACD_NVETO_05AM	0< V < 3.3	
42	JP1 - 61	ACD_NVETO_05AP	0< V < 3.3	
43	JP1 - 62	ACD_NVETO_04AM	0< V < 3.3	
44	JP1 - 63	ACD_NVETO_04AP	0< V < 3.3	
45	JP1 - 64	ACD_NVETO_03AM	0< V < 3.3	
46	JP1 - 65	ACD_NVETO_03AP	0< V < 3.3	
47	JP1 - 66	ACD_NVETO_02AM	0< V < 3.3	
48	JP1 - 67	ACD_NVETO_02AP	0< V < 3.3	
49	JP1 - 68	ACD_NVETO_01AM	0< V < 3.3	
50	JP1 - 69	ACD_NVETO_01AP	0< V < 3.3	
51	JP1 - 70	ACD_NVETO_00AM	0< V < 3.3	
52	JP1 - 71	ACD_NVETO_00AP	0< V < 3.3	
53	JP1 - 72	ACD_NSDATA_AM	0< V < 3.3	
54	JP1 - 73	ACD_NSDATA_AP	0< V < 3.3	
55	JP1 - 74	ACD_NRST_AM	0< V < 3.3	
56	JP1 - 75	ACD_NRST_AP	0< V < 3.3	
57	JP1 - 76	ACD_NSCMD_AM	0< V < 3.3	
58	JP1 - 77	ACD_NSCMD_AP	0< V < 3.3	
59	JP1 - 78	ACD_CLK_AM	0< V < 3.3	
60	JP1 - 79	ACD_CLK_AP	0< V < 3.3	
61	JS2 - 1	ACD_VDD_0B	+3.3V	
62	JS2 - 3	ACD_VDD_1B	+3.3V	
63	JS2 - 4	ACD_VDD_2B	+3.3V	
64	JS2 – 5	ACD_28V_0B	+28V	
65	JS2 - 7	ACD_28V_1B	+28V	
66	JS2 - 17	ACD_NVETO_16BP	0< V < 3.3	
67	JS2 - 18	ACD_NVETO_16BM	0< V < 3.3	
68	JS2 - 19	ACD_NVETO_17BP	0< V < 3.3	
69	JS2 - 20	ACD_NVETO_17BM	0< V < 3.3	
70	JS2 - 21	ACD_NCNO_BP	0< V < 3.3	
71	JS2 - 22	ACD_NCNO_BM	0< V < 3.3	
72	JS2 - 23	ACD_HV_BP	0< V < 3.3	
73	JS2 - 24	ACD_HV_BM	0< V < 3.3	
74	JS2 - 25	ACD_TEMP_BP	TBD	
75	JS2 - 26	ACD_TEMP_BM	TBD	
76	JS2 - 30	ACD_GND_0B	0	
77	JS2 - 31	ACD_GND_1B	0	
78	JS2 - 32	ACD_GND_2B	0	
79	JS2 - 33	ACD_28V_RTN_0B	0	
80	JS2 - 34	ACD_28V_RTN_1B	0	
81	JS2 - 40	ACD_NVETO_15BM	0< V < 3.3	
82	JS2 - 41	ACD_NVETO_15BP	0< V < 3.3	
83	JS2 - 42	ACD_NVETO_14BM	0< V < 3.3	
84	JS2 - 43	ACD_NVETO_14BP	0< V < 3.3	
85	JS2 - 44	ACD_NVETO_13BM	0< V < 3.3	
86	JS2 - 45	ACD_NVETO_13BP	0< V < 3.3	
87	JS2 - 46	ACD_NVETO_12BM	0< V < 3.3	
88	JS2 - 47	ACD_NVETO_12BP	0< V < 3.3	
89	JS2 - 48	ACD_NVETO_11BM	0< V < 3.3	

90	JS2 - 49	ACD_NVETO_11BP	0< V < 3.3	
91	JS2 - 50	ACD_NVETO_10BM	0< V < 3.3	
92	JS2 - 51	ACD_NVETO_10BP	0< V < 3.3	
93	JS2 - 52	ACD_NVETO_09BM	0< V < 3.3	
94	JS2 - 53	ACD_NVETO_09BP	0< V < 3.3	
95	JS2 - 54	ACD_NVETO_08BM	0< V < 3.3	
96	JS2 - 55	ACD_NVETO_08BP	0< V < 3.3	
97	JS2 - 56	ACD_NVETO_07BM	0< V < 3.3	
98	JS2 - 57	ACD_NVETO_07BP	0< V < 3.3	
99	JS2 - 58	ACD_NVETO_06BM	0< V < 3.3	
100	JS2 - 59	ACD_NVETO_06BP	0< V < 3.3	
101	JS2 - 60	ACD_NVETO_05BM	0< V < 3.3	
102	JS2 - 61	ACD_NVETO_05BP	0< V < 3.3	
103	JS2 - 62	ACD_NVETO_04BM	0< V < 3.3	
104	JS2 - 63	ACD_NVETO_04BP	0< V < 3.3	
105	JS2 - 64	ACD_NVETO_03BM	0< V < 3.3	
106	JS2 - 65	ACD_NVETO_03BP	0< V < 3.3	
107	JS2 - 66	ACD_NVETO_02BM	0< V < 3.3	
108	JS2 - 67	ACD_NVETO_02BP	0< V < 3.3	
109	JS2 - 68	ACD_NVETO_01BM	0< V < 3.3	
110	JS2 - 69	ACD_NVETO_01BP	0< V < 3.3	
111	JS2 - 70	ACD_NVETO_00BM	0< V < 3.3	
112	JS2 - 71	ACD_NVETO_00BP	0< V < 3.3	
113	JS2 - 72	ACD_NSDATA_BM	0< V < 3.3	
114	JS2 - 73	ACD_NSDATA_BP	0< V < 3.3	
115	JS2 - 74	ACD_NRST_BM	0< V < 3.3	
116	JS2 - 75	ACD_NRST_BP	0< V < 3.3	
117	JS2 - 76	ACD_NSCMD_BM	0< V < 3.3	
118	JS2 - 77	ACD_NSCMD_BP	0< V < 3.3	
119	JS2 - 78	ACD_CLK_BM	0< V < 3.3	
120	JS2 - 79	ACD_CLK_BP	0< V < 3.3	
121	JHV1 - 1	HVBS +28V	+28V	
122	JHV1 - 2	28V_RTN	0	
123	JHV1 - 3	HV_MON_P1	0	
124	JHV1 - 4	DAC_P	0	
125	JHV1 - 5	HV_ENABLE_1	0	
126	JHV1 - 6	HVBS +28V	+28V	
127	JHV1 - 7	28V_RTN	0	
128	JHV1 - 8	HV_MON_N1	0	
129	JHV1 - 9	DAC_N	0	
130	JHV2 - 1	HVBS +28V	+28V	
131	JHV2 - 2	28V_RTN	0	
132	JHV2 - 3	HV_MON_P2	0	
133	JHV2 - 4	DAC_P	0	
134	JHV2 - 5	HV_ENABLE_2	0	
135	JHV2 - 6	HVBS +28V	+28V	
136	JHV2 - 7	28V_RTN	0	
137	JHV2 - 8	HV_MON_N2	0	
138	JHV2 - 9	DAC_N	0	

7.2 Electrical Integration of the FREE Card and Initial Current Tests

The pin connections for the two 79 pin circular connectors JP1 and JS2 are listed in Appendix 1. The pin connections for the two 9 pin MDM connectors to the High Voltage Bias Supplies are listed in Appendix 2.

At this time, insert all jumper plugs into the break out boxes on connectors JP1, JS2, JHV1, and JHV2.

7.3 Characterizations of the FREE Assembly In-Rush Currents

This test will be used to characterize the transient currents associated with power-on for each FREE assembly. This test will require a current probe connected to an oscilloscope as shown in the diagram below. This test is useful for characterizing the ACD electronics assembly. A fully populated FREE card with two high voltage bias supplies integrated and both ACD interface connectors connected is required to obtain an accurate measurement. Since the phototube signals are AC coupled to the system and provide essentially no load to the FREE card, these inputs need not be connected for this test. The PMT resistor divider networks do provide a load to the HVBS, so the bias supplies should either be connected to 18 PMTs or have a representative dummy load for this test.

Breakout boxes may be in place for this test and all pins shall be inserted into the break-out boxes except the following:

JP1 – 1, 2, 4 (ACD_VDD) and JP1 – 5, 7 (ACD_28V).
JP2 – 1, 2, 4 (ACD_VDD) and JP2 – 5, 7 (ACD_28V).

Place jumper wire loops in the BOB at pins JP1-1 and JP1-5 to measure the +3.3V and +28V in-rush currents, respectively.

Set the oscilloscope to trigger on the current probe. Place the current probe on the JP1-1 wire. Turn on the ACD electronics and capture the in-rush waveform (repeat this measurement until the full time and amplitude are correctly captured). Record the peak in-rush current for the +3.3V rail below and attach a printout of the in-rush plot to the as-run copy of this procedure.

+3.3V peak in-rush current: _____

Move the current probe to the JP1-5 wire and reset the oscilloscope to trigger on the current probe. Turn on the ACD electronics and capture the in-rush waveform (repeat this measurement until the full time and amplitude are correctly captured). Record the peak in-rush current for the +28V rail below and attach a printout of the in-rush plot to the as-run copy of this procedure.

+28V peak in-rush current: _____

Label the copies of the in-rush current plots with the test number, date, and test conductor name before continuing.

7.4 Measurement of the Power Supply, DC Bias and Reference Voltages

This section measures DC voltages at several points on the FREE board and compares them to the expected values. Using the digital multimeter (autoscaling) and measuring DC volts, connect the (-) lead to the +3.3V return and measure the following voltages with the (+) lead at the FREE board location listed in the table below.

Meas. No.	FREE Circuit Card Probe Point	Alternate Probe Point	Expected Voltage	Measured Voltage
1	UADC00-5	CA4-1	2.50 V	
2	UADC01-5	C139-1	2.50 V	
3	UADC02-5	C172-1	2.50 V	
4	UADC03-5	C183-1	2.50 V	
5	UADC04-5	C29-1	2.50 V	
6	UADC05-5	C40-1	2.50 V	
7	UADC06-5	C73-1	2.50 V	
8	UADC07-5	C84-1	2.50 V	
9	UADC08-5	C117-1	2.50 V	
10	UADC09-5	C128-1	2.50 V	
11	UADC10-5	C150-1	2.50 V	
12	UADC11-5	C161-1	2.50 V	
13	UADC12-5	C194-1	2.50 V	
14	UADC13-5	C208-1	2.50 V	
15	UADC14-5	C51-1	2.50 V	
16	UADC15-5	C62-1	2.50 V	
17	UADC16-5	C95-1	2.50 V	
18	UADC17-5	C106-1	2.50 V	
19	UGA00-8	C266-1	2.50 V	
20	UGA01-8	C248-1	2.50 V	
21	UGA02-8	C257-1	2.50 V	
22	UGA03-8	C260-1	2.50 V	
23	UGA04-8	C218-1	2.50 V	
24	UGA05-8	C221-1	2.50 V	
25	UGA06-8	C230-1	2.50 V	
26	UGA07-8	C233-1	2.50 V	
27	UGA08-8	C242-1	2.50 V	
28	UGA09-8	C245-1	2.50 V	
29	UGA10-8	C251-1	2.50 V	
30	UGA11-8	C254-1	2.50 V	
31	UGA12-8	C263-1	2.50 V	
32	UGA13-8	C269-1	2.50 V	
33	UGA14-8	C224-1	2.50 V	
34	UGA15-8	C227-1	2.50 V	
35	UGA16-8	C236-1	2.50 V	
36	UGA17-8	C239-1	2.50 V	
37	UGA00-22 (VDD)	CA5-1	> 3.1 V	
38	UGA01-22 (VDD)	CA5-1	> 3.1 V	
39	UGA02-22 (VDD)	C143-1	> 3.1 V	
40	UGA03-22 (VDD)	C176-1	> 3.1 V	
41	UGA04-22 (VDD)	C33-1	> 3.1 V	
42	UGA05-22 (VDD)	C44-1	> 3.1 V	
43	UGA06-22 (VDD)	C77-1	> 3.1 V	
44	UGA07-22 (VDD)	C88-1	> 3.1 V	
45	UGA08-22 (VDD)	C121-1	> 3.1 V	
46	UGA09-22 (VDD)	C132-1	> 3.1 V	
47	UGA10-22 (VDD)	C154-1	> 3.1 V	
48	UGA11-22 (VDD)	C165-1	> 3.1 V	

49	UGA12-22 (VDD)	C198-1	> 3.1 V	
50	UGA13-22 (VDD)	C212-1	> 3.1 V	
51	UGA14-22 (VDD)	C55-1	> 3.1 V	
52	UGA15-22 (VDD)	C66-1	> 3.1 V	
53	UGA16-22 (VDD)	C99-1	> 3.1 V	
54	UGA17-22 (VDD)	C110-1	> 3.1 V	
55	UGA00-1 (VCC)	CA7-1	> 3.1 V	
56	UGA01-1 (VCC)	C145-1	> 3.1 V	
57	UGA02-1 (VCC)	C178-1	> 3.1 V	
58	UGA03-1 (VCC)	C189-1	> 3.1 V	
59	UGA04-1 (VCC)	C35-1	> 3.1 V	
60	UGA05-1 (VCC)	C46-1	> 3.1 V	
61	UGA06-1 (VCC)	C79-1	> 3.1 V	
62	UGA07-1 (VCC)	C90-1	> 3.1 V	
63	UGA08-1 (VCC)	C123-1	> 3.1 V	
64	UGA09-1 (VCC)	C134-1	> 3.1 V	
65	UGA10-1 (VCC)	C156-1	> 3.1 V	
66	UGA11-1 (VCC)	C167-1	> 3.1 V	
67	UGA12-1 (VCC)	C200-1	> 3.1 V	
68	UGA13-1 (VCC)	C214-1	> 3.1 V	
69	UGA14-1 (VCC)	C57-1	> 3.1 V	
70	UGA15-1 (VCC)	C68-1	> 3.1 V	
71	UGA16-1 (VCC)	C101-1	> 3.1 V	
72	UGA17-1 (VCC)	C112-1	> 3.1 V	
73	TPR1	UR1-2	< 0.2 V	
74	DR1-A	UR1-1	> 3.0 V	
75	UID1-1	N/A	< 0.2 V	
76	UID1-2	N/A	< 0.2 V	
77	UID1-3	N/A	N/A (ID_4)	
78	UID1-4	N/A	N/A (ID_5)	
79	UID1-5	N/A	N/A (ID_6)	
80	UID1-6	N/A	N/A (ID_7)	
81	UID1-11	N/A	N/A (ID_0)	
82	UID1-12	N/A	N/A (ID_1)	
83	UID1-13	N/A	N/A (ID_2)	
84	UID1-14	N/A	N/A (ID_3)	
85	JH0 Coax	N/A	N/A	
86	JSA0 Coax	N/A	N/A	
87	JHLD0 Coax	N/A	N/A	
88	CP1-1	CP2-1	> 3.1 V (VDD)	
89	CP1-2	CP2-2	< 0.1 V (RTN)	
90	CP7-1	CP8-1	> 3.1 V (VCC)	
91	CP7-2	CP8-2	< 0.1 V (RTN)	
92	CG9-1	CG8-1	> 3.1 V (VDD)	
93	JHV1-1	JHV1-6	28 V	
94	JHV1-2	JHV1-7	0 V	
95	JHV1-3	N/A	0 V	
96	JHV1-4	N/A	0 V	
97	JHV1-5	N/A	0 V	
98	JHV1-8	N/A	0 V	
99	JHV1-9	N/A	0 V	

100	JHV2-1	JHV2-6	28 V	
101	JHV2-2	JHV2-7	0 V	
102	JHV2-3	N/A	0 V	
103	JHV2-4	N/A	0 V	
104	JHV2-5	N/A	0 V	
105	JHV2-8	N/A	0 V	
106	JHV2-9	N/A	0 V	
107	UDAC3-16	CDA1-1	> 3.1 V	
108	UDAC3-2	TPDAC2	0 V	
109	UDAC3-14	TPREF1	1.25 V	
110	TPDAC3	UDAC1-1	2.50 V	
111	JDAC2 Coax	UDAC2-7	1.40 V	
112	JDAC1 Coax	RDA3-2	< 0.3 V	

7.5 Measurement of the GARC Bias Resistors and Voltages

Record the biasing resistors associated with the GARC in the table below. If assembly information (or visual inspection) is not available, power off the ACD electronics and make the following measurements on the circuit card (do not perform the measurement step on a conformally coated board). Indicate in the table if the biasing resistor values are either measured or taken from other documentation.

FREE Resistor	Value	Measured	Recorded from FREE Assembly Documentation
RG1			
RG2			
RG3			
RG4			
RG5			
RG6			
RG7			
RG8			
RG9			
RG10			

Using the DC multimeter, measure the following GARC bias voltages.

GARC Bias Voltage	Location	Alternate Location	Measured Voltage
HLD_WOR_BIAS	GARC-104	RG8-1	
BIAS_RCVR	GARC-156	RG6-1	
BIAS_DRV_H	GARC-160	RG3-1	
BIAS_DRV_L	GARC-169	RG1-1	
LVDS_PRESET_ADJ	GARC-184	RG10-1	

7.6 Measurement of the Power Supply Quality at the ACD Interface

This section of the test will verify the general functionality and performance of the power provided to the ACD at the interface while loaded with the nominal load of the ACD electronics assembly.

The ICD requires that the +3.3V supply be in the range of +3.20 to +3.60V referenced to the +3.3V return at the interface. Additionally, the RMS noise on this supply shall be less than 5 mV over a bandwidth of 1 MHz and below.

The +28V supply to the high voltage bias supplies must be in the range of +22.0V to +38.7V referenenced to the 28V return. The RMS noise on this supply shall be less than 10 mV over a bandwidth of 1 MHz and below.

Using a calibrated multimeter, measure the DC and RMS voltages at the following points and verify that they are within specification prior to proceeding. LAT commands to turn on power to the Primary and Secondary sides, respectively, will be required for these measurements. In situations where only one power supply is present, the Test Conductor shall make a note to this effect and skip the measurements which are not applicable.

Meas. No.	Signal Name	FREE Circuit Card Break Out Box V+ Probe	FREE Circuit Card Break Out Box RTN Probe	Measured DC Voltage	Expected DC Voltage	Measured RMS Voltage	Expected RMS Voltage
1	+3.3 V Primary	JP1-1	JP1-30		3.20 – 3.60		< 5 mV RMS
2	+28 V Primary	JP1-5	JP1-33		22.0 – 38.7		< 10 mV RMS
3	+3.3V Secondary	JS2-1	JS2-30		3.20 – 3.60		< 5 mV RMS
4	+28 V Secondary	JS2-5	JS2-33		22.0 – 38.7		< 10 mV RMS

8.0 Basic Aliveness Test

This section of the test will verify the general aliveness of the powered-on FREE circuit assembly. The initial power levels, the command and data interface, the DAC commanding, the GAFE test charge injection, and the pulse height analysis (PHA) circuitry will be checked for basic functionality. The pin connections for the two 79 pin circular connectors JP1 and JS2 are listed in Appendix 1. The pin connections for the two 9 pin MDM connectors to the High Voltage Bias Supplies are listed in Appendix 2.

8.1 GARC and GAFE Registers Initial Reset Test

This section will verify that GARC registers have been properly initialized during a reset command. The following test sequence of commands will perform this verification. This test will also verify that the GARC to AEM command and data return link is functional.

1. Send the GARC_Reset command.
2. Send the Trigger_ZS command (this captures the FREE board ID).
3. Send the Veto_Delay_Rd command. The data field in the return data stream should be 5.
4. Send the HVBS_Level_Rd command. The data field in the return data stream should be 0.
5. Send the SAA_Level_Rd command. The data field in the return data stream should be 0.
6. Send the Hold_Delay_Rd command. The data field in the return data stream should be 28.

7. Send the Veto_Width_Rd command. The data field in the return data stream should be 2.
8. Send the HitMap_Width_Rd command. The data field in the return data stream should be 7.
9. Send the HitMap_Deadtime_Rd command. The data field in the return data stream should be 0.
10. Send the HitMap_Delay_Rd command. The data field in the return data stream should be 16.
11. Send the PHA_En0_Rd command. The data field in the return data stream should be 65535.
12. Send the PHA_En1_Rd command. The data field in the return data stream should be 3.
13. Send the Veto_En0_Rd command. The data field in the return data stream should be 65535.
14. Send the Veto_En1_Rd command. The data field in the return data stream should be 3.
15. Send the Max_PHA_Rd command. The data field in the return data stream should be 4.
16. Send the GARC_Mode_Rd command. The data field in the return data stream should be 768.
17. Send the GARC_Status command. The data field in the return data stream should be 24.
18. Send the GARC_Cmd_Reg command. The data field in the return data stream should be 0.
19. Send the GARC_Cmd_Rejects command. The data field in the return data stream should be 0.
20. Send the FREE_Board_ID command. The data field in the return data stream should be the same as the FREE board serial number.
21. Send the GARC_Version command. The data field in the return data stream should be 1 for GARC V1 (e.g., version #1, May 2002).
22. Send the PHA_Thresh00_Rd command. The data field in the return data stream should be 1114.
23. Send the PHA_Thresh01_Rd command. The data field in the return data stream should be 1114.
24. Send the PHA_Thresh02_Rd command. The data field in the return data stream should be 1114.
25. Send the PHA_Thresh03_Rd command. The data field in the return data stream should be 1114.
26. Send the PHA_Thresh04_Rd command. The data field in the return data stream should be 1114.
27. Send the PHA_Thresh05_Rd command. The data field in the return data stream should be 1114.
28. Send the PHA_Thresh06_Rd command. The data field in the return data stream should be 1114.
29. Send the PHA_Thresh07_Rd command. The data field in the return data stream should be 1114.
30. Send the PHA_Thresh08_Rd command. The data field in the return data stream should be 1114.
31. Send the PHA_Thresh09_Rd command. The data field in the return data stream should be 1114.
32. Send the PHA_Thresh10_Rd command. The data field in the return data stream should be 1114.
33. Send the PHA_Thresh11_Rd command. The data field in the return data stream should be 1114.
34. Send the PHA_Thresh12_Rd command. The data field in the return data stream should be 1114.
35. Send the PHA_Thresh13_Rd command. The data field in the return data stream should be 1114.
36. Send the PHA_Thresh14_Rd command. The data field in the return data stream should be 1114.
37. Send the PHA_Thresh15_Rd command. The data field in the return data stream should be 1114.
38. Send the PHA_Thresh16_Rd command. The data field in the return data stream should be 1114.
39. Send the PHA_Thresh17_Rd command. The data field in the return data stream should be 1114.
40. Send the ADC_TACQ_Rd command. The data field in the return data stream should be 0.

41. Initiate an external GARC Power On Reset by placing a 0V to 3.3V to 0V pulse approximately 100 μsec wide (or longer) on the GARC external reset pin, pin 182.

42. Verify the status of the GARC registers as shown in steps 2 – 40 above.

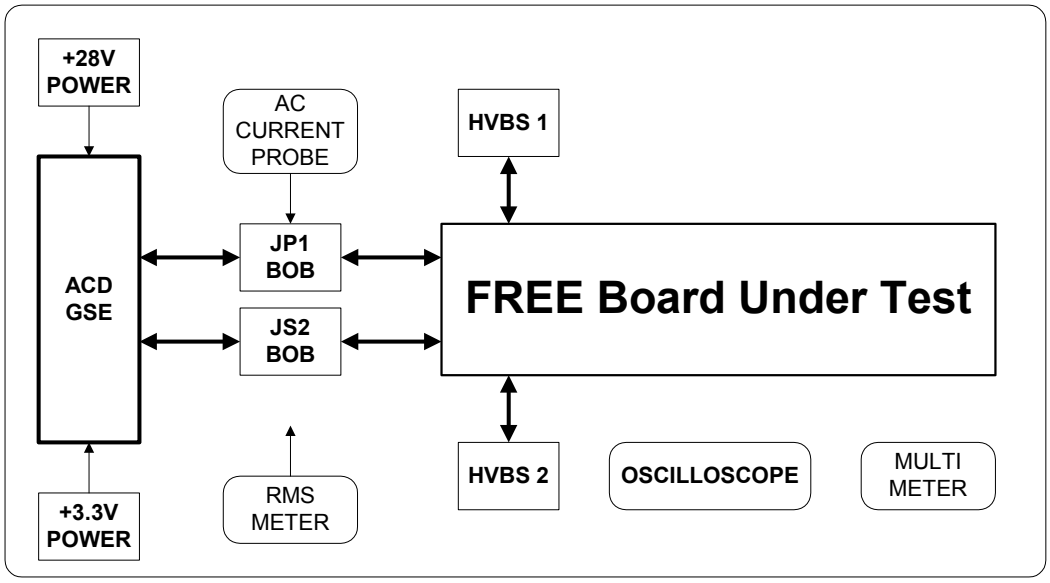
If all steps above have verified correctly, the GARC reset function has been verified. A summary of the initial reset parameters for GARC and GAFE is detailed below.

Register	Initial Value (decimal)	Initial Value (hex)
Veto_Delay	5	5
HVBS_Level	0	0
SAA_Level	0	0
Hold_Delay	28	1C
Veto_Width	2	2
HitMap_Width	7	7
HitMap_Deadtime	3	3
HitMap_Delay	16	10
PHA_EN0	65535	FFFF
PHA_EN1	3	3
VETO_EN0	65535	FFFF
VETO_EN1	3	3
Max_PHA	4	4
GARC_Mode	768	300
GARC_Status	24	18
Command_Register	0	0
GARC_Diagnostic	0	0
Cmd_Reject_Ctr	0	0
FREE_Board_ID	*	*
GARC_Version	1	1
PHA_Threshold_00	1114	45A
PHA_Threshold_01	1114	45A
PHA_Threshold_02	1114	45A
PHA_Threshold_03	1114	45A
PHA_Threshold_04	1114	45A
PHA_Threshold_05	1114	45A
PHA_Threshold_06	1114	45A
PHA_Threshold_07	1114	45A
PHA_Threshold_08	1114	45A
PHA_Threshold_09	1114	45A
PHA_Threshold_10	1114	45A
PHA_Threshold_11	1114	45A
PHA_Threshold_12	1114	45A
PHA_Threshold_13	1114	45A
PHA_Threshold_14	1114	45A
PHA_Threshold_15	1114	45A
PHA_Threshold_16	1114	45A
PHA_Threshold_17	1114	45A
ADC_TACQ	0	0
GAFE_Mode	48	30
GAFE_DAC1	57	39
GAFE_DAC2	38	26
GAFE_DAC3	55	37
GAFE_DAC4	32	20
GAFE_DAC5	0	0
GAFE_Wr_Ctr	0	0

GAFE_Reject_Ctr	0	0

9.0 Testing the FREE Electronics via Command-Response Protocol

The GARC logic is based on a command-response protocol and requires an AEM or AEM simulator to access the logic functions. For each of the following commands, the proper GARC and/or GAFE response may be tested. Note that all GAFEs will respond to a broadcast write command (e.g., GAFE address of decimal 31), but a read command requires a unique GAFE address. The following test details the proper sequence for a single GARC ASIC. The steps are numbered for easier reference. The test setup required is detailed in the diagram below.



9.1 FREE Power Measurement at the Nominal Power Supply Voltage

Verify that the GARC power supply is set to +3.30V. After initial power up, measure the +3.30V power supply current to the GARC in the following modes. Record these values in the table below.

1. Send the GARC_Mode_Wr command with a data argument of 768. Send the GARC_Mode_Rd command to verify. Both the primary and secondary LVDS VETO drivers should be enabled. Record the GARC current in the table below.
2. Send the GARC_Mode_Wr command with a data argument of 256. Send the GARC_Mode_Rd command to verify. Only the primary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
3. Send the GARC_Mode_Wr command with a data argument of 512. Send the GARC_Mode_Rd command to verify. Only the secondary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
4. Send the GARC_Mode_Wr command with a data argument of 0. Send the GARC_Mode_Rd command to verify. None of the LVDS VETO drivers should now be enabled. Record the GARC current in the table below.

GARC Mode	GARC_Mode_Wr	+3.3V Current	+3.3V Current

	Data Argument	Measured (mA)	Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		

9.2 FREE Power Measurement at the Minimum Power Supply Voltage

Verify that the GARC power supply is set to +3.0V. After initial power up, measure the +3.0V power supply current to the GARC in the following modes. Record these values in the table below.

1. Send the GARC_Mode_Wr command with a data argument of 768. Send the GARC_Mode_Rd command to verify. Both the primary and secondary LVDS VETO drivers should be enabled. Record the GARC current in the table below.
2. Send the GARC_Mode_Wr command with a data argument of 256. Send the GARC_Mode_Rd command to verify. Only the primary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
3. Send the GARC_Mode_Wr command with a data argument of 512. Send the GARC_Mode_Rd command to verify. Only the secondary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
4. Send the GARC_Mode_Wr command with a data argument of 0. Send the GARC_Mode_Rd command to verify. None of the LVDS VETO drivers should now be enabled. Record the GARC current in the table below.

GARC Mode	GARC_Mode_Wr Data Argument	+3.0V Current Measured (mA)	+3.0V Current Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		

9.3 FREE Power Measurement at the Maximum Power Supply Voltage

Verify that the GARC power supply is set to +3.6V. After initial power up, measure the +3.6V power supply current to the GARC in the following modes. Record these values in the table below.

1. Send the GARC_Mode_Wr command with a data argument of 768. Send the GARC_Mode_Rd command to verify. Both the primary and secondary LVDS VETO drivers should be enabled. Record the GARC current in the table below.
2. Send the GARC_Mode_Wr command with a data argument of 256. Send the GARC_Mode_Rd command to verify. Only the primary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.

3. Send the GARC_Mode_Wr command with a data argument of 512. Send the GARC_Mode_Rd command to verify. Only the secondary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
4. Send the GARC_Mode_Wr command with a data argument of 0. Send the GARC_Mode_Rd command to verify. None of the LVDS VETO drivers should now be enabled. Record the GARC current in the table below.

GARC Mode	GARC_Mode_Wr Data Argument	+3.6V Current Measured (mA)	+3.6V Current Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		

5. Reset the GARC power supply to +3.3V.
6. Send the GARC_Reset command to return the GARC to the initial power-on configuration.

9.4 High Voltage Bias Supply Power Measurements at the Minimum Power Supply Voltage

1. ***The Test Conductor shall verify that it is safe to apply high voltage to the output of both High Voltage Bias Supplies prior to starting this section. If in any way applying high voltage is not appropriate for this section of the test, the Test Conductor shall note this and skip this section.***

2. Remove the jumper plugs in the ACD interface break out at the following pins: JP1-5, JP1-7, JS2-5, JS2-7. Place a current meter in series with the JP1-5 connection on the break out box.
3. Set the ground support equipment to power the ACD on the Primary side.
4. Verify that the FREE board power supply is set to +3.3V. Set the High Voltage Bias Supply power supply to +22.0 V. Power on the FREE electronics assembly (both +3.3V and +28V).
5. Send the GARC command HVBS_Level_Rd and verify that the return data is 0.
6. Send the GARC mode command GARC_Mode_Wr with a data argument of decimal 894 (0x37E) to enable both high voltage bias supplies.
7. The analog command level to the high voltage bias supplies are << more >>

Only one at a time

Measure 28V current, dac command, hv mon out

Do levels of DAC 0 (0v), DAC 1092 (~400V), DAC 2730 (~1000V), DAC 3412 (~1250V)

9.5 High Voltage Bias Supply Power Measurements at the Nominal Power Supply Voltage

The Test Conductor shall verify that it is safe to apply high voltage to the output of both High Voltage Bias Supplies prior to starting this section. If in any way applying high voltage is not appropriate for this section of the test, the Test Conductor shall note this and skip this section.

<< REPEAT above section >>

9.6 High Voltage Bias Supply Power Measurements at the Maximum Power Supply Voltage

The Test Conductor shall verify that it is safe to apply high voltage to the output of both High Voltage Bias Supplies prior to starting this section. If in any way applying high voltage is not appropriate for this section of the test, the Test Conductor shall note this and skip this section.

<< REPEAT above section >>

9.7 GARC Register Read/Write Tests

This section tests the proper functioning of each bit of the commandable registers in the GARC. The intent is to toggle each bit in a variety of patterns to ensure that all bits are addressable and that there is no stuck-at-fault condition. The following GARC registers will be tested in this section:

GARC Register Name	Register Width (bits)	Register Type
Veto_Delay	5	Read/Write
HVBS_Level	12	Read/Write
SAA_Level	12	Read/Write
Hold_Delay	7	Read/Write
Veto_Width	3	Read/Write
HitMap_Width	4	Read/Write
HitMap_Deadtime	3	Read/Write
HitMap_Delay	5	Read/Write
PHA_En0	16	Read/Write
PHA_En1	2	Read/Write
VETO_En0	16	Read/Write
VETO_En1	2	Read/Write
Max_PHA	5	Read/Write
GARC_Mode	11	Read/Write
GARC_Status	6	Read Only
Command_Register	16	Read Only
GARC_Diagnostic	16	Read Only
Cmd_Reject_Counter	8	Read Only
FREE_Board_ID	8	Read Only
GARC_Version	3	Read Only
PHA_Threshold_00	16	Read/Write
PHA_Threshold_01	16	Read/Write
PHA_Threshold_02	16	Read/Write
PHA_Threshold_03	16	Read/Write
PHA_Threshold_04	16	Read/Write
PHA_Threshold_05	16	Read/Write
PHA_Threshold_06	16	Read/Write
PHA_Threshold_07	16	Read/Write
PHA_Threshold_08	16	Read/Write
PHA_Threshold_09	16	Read/Write
PHA_Threshold_10	16	Read/Write
PHA_Threshold_11	16	Read/Write
PHA_Threshold_12	16	Read/Write
PHA_Threshold_13	16	Read/Write

PHA_Threshold_14	16	Read/Write
PHA_Threshold_15	16	Read/Write
PHA_Threshold_16	16	Read/Write
PHA_Threshold_17	16	Read/Write
ADC_TACQ	6	Read/Write

9.8 Veto_Delay Register Test

1. Send the Veto_Delay_Wr command with a data argument of 5'h0 (0). Send the Veto_Delay_Rd command and read back this commanded data argument.
2. Send the Veto_Delay_Wr command with a data argument of 5'h15 (21). Send the Veto_Delay_Rd command and read back this commanded data argument.
3. Send the Veto_Delay_Wr command with a data argument of 5'h0A (10). Send the Veto_Delay_Rd command and read back this commanded data argument.
4. Send the Veto_Delay_Wr command with a data argument of 5'h1F (31). Send the Veto_Delay_Rd command and read back this commanded data argument.
5. Send the Veto_Delay_Wr command with a data argument of 5'h5 (5). Send the Veto_Delay_Rd command and read back this commanded data argument.

9.9 HVBS Level Register Test

1. Send the HVBS_Level_Wr command with a data argument of 12'h0 (0). Send the HVBS_Level_Rd command and read back this commanded data argument.
2. Send the HVBS_Level_Wr command with a data argument of 12'h555 (1365). Send the HVBS_Level_Rd command and read back this commanded data argument.
3. Send the HVBS_Level_Wr command with a data argument of 12'hAAA (2730). Send the HVBS_Level_Rd command and read back this commanded data argument.
4. Send the HVBS_Level_Wr command with a data argument of 12'hFFF (4095). Send the HVBS_Level_Rd command and read back this commanded data argument.
5. Send the HVBS_Level_Wr command with a data argument of 12'h0 (0). Send the HVBS_Level_Rd command and read back this commanded data argument.

9.10 SAA Level Register Test

1. Send the SAA_Level_Wr command with a data argument of 12'h0 (0). Send the SAA_Level_Rd command and read back this commanded data argument.
2. Send the SAA_Level_Wr command with a data argument of 12'h555 (1365). Send the SAA_Level_Rd command and read back this commanded data argument.
3. Send the SAA_Level_Wr command with a data argument of 12'hAAA (2730). Send the SAA_Level_Rd command and read back this commanded data argument.
4. Send the SAA_Level_Wr command with a data argument of 12'hFFF (4095). Send the SAA_Level_Rd command and read back this commanded data argument.
5. Send the SAA_Level_Wr command with a data argument of 12'h0 (0). Send the SAA_Level_Rd command and read back this commanded data argument.

9.11 Hold Delay Register Test

1. Send the Hold_Delay_Wr command with a data argument of 7'h0 (0). Send the Hold_Delay_Rd command and read back this commanded data argument.
2. Send the Hold_Delay_Wr command with a data argument of 7'h55 (85). Send the Hold_Delay_Rd command and read back this commanded data argument.
3. Send the Hold_Delay_Wr command with a data argument of 7'h2A (42). Send the Hold_Delay_Rd command and read back this commanded data argument.
4. Send the Hold_Delay_Wr command with a data argument of 7'h7F (127). Send the Hold_Delay_Rd command and read back this commanded data argument.
5. Send the Hold_Delay_Wr command with a data argument of 7'h1C (28). Send the Hold_Delay_Rd command and read back this commanded data argument.

9.12 Veto Width Register Test

1. Send the Veto_Width_Wr command with a data argument of 3'h0 (0). Send the Veto_Width_Rd command and read back this commanded data argument.
2. Send the Veto_Width_Wr command with a data argument of 3'h5 (5). Send the Veto_Width_Rd command and read back this commanded data argument.
3. Send the Veto_Width_Wr command with a data argument of 3'h7 (7). Send the Veto_Width_Rd command and read back this commanded data argument.
4. Send the Veto_Width_Wr command with a data argument of 3'h2 (2). Send the Veto_Width_Rd command and read back this commanded data argument.

9.13 HitMap Width Register Test

1. Send the HitMap_Width_Wr command with a data argument of 4'h0 (0). Send the HitMap_Width_Rd command and read back this commanded data argument.
2. Send the HitMap_Width_Wr command with a data argument of 4'h5 (5). Send the HitMap_Width_Rd command and read back this commanded data argument.
3. Send the HitMap_Width_Wr command with a data argument of 4'hA (10). Send the HitMap_Width_Rd command and read back this commanded data argument.
4. Send the HitMap_Width_Wr command with a data argument of 4'hF (15). Send the HitMap_Width_Rd command and read back this commanded data argument.
5. Send the HitMap_Width_Wr command with a data argument of 4'h7 (7). Send the HitMap_Width_Rd command and read back this commanded data argument.

9.14 HitMap Deadtime Register Test

1. Send the HitMap_Deadtime_Wr command with a data argument of 3'h0 (0). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.
2. Send the HitMap_Deadtime_Wr command with a data argument of 3'h5 (5). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.

3. Send the HitMap_Deadtime_Wr command with a data argument of 3'h2 (2). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.
4. Send the HitMap_Deadtime_Wr command with a data argument of 3'h7 (7). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.
5. Send the HitMap_Deadtime_Wr command with a data argument of 3'h3 (3). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.

9.15 HitMap Delay Register Test

1. Send the HitMap_Delay_Wr command with a data argument of 5'h0 (0). Send the HitMap_Delay_Rd command and read back this commanded data argument.
2. Send the HitMap_Delay_Wr command with a data argument of 5'h15 (21). Send the HitMap_Delay_Rd command and read back this commanded data argument.
3. Send the HitMap_Delay_Wr command with a data argument of 5'h0A (10). Send the HitMap_Delay_Rd command and read back this commanded data argument.
4. Send the HitMap_Delay_Wr command with a data argument of 5'h1F (31). Send the HitMap_Delay_Rd command and read back this commanded data argument.
5. Send the HitMap_Delay_Wr command with a data argument of 5'h10 (16). Send the HitMap_Delay_Rd command and read back this commanded data argument.

9.16 PHA En0 Register Test

1. Send the PHA_En0_Wr command with a data argument of 16'h0 (0). Send the PHA_En0_Rd command and read back this commanded data argument.
2. Send the PHA_En0_Wr command with a data argument of 16'h5555 (21845). Send the PHA_En0_Rd command and read back this commanded data argument.
3. Send the PHA_En0_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_En0_Rd command and read back this commanded data argument.
4. Send the PHA_En0_Wr command with a data argument of 16'hFFFF (65535). Send the PHA_En0_Rd command and read back this commanded data argument.

9.17 PHA En1 Register Test

1. Send the PHA_En1_Wr command with a data argument of 2'h0 (0). Send the PHA_En1_Rd command and read back this commanded data argument.
2. Send the PHA_En1_Wr command with a data argument of 2'h1 (1). Send the PHA_En1_Rd command and read back this commanded data argument.
3. Send the PHA_En1_Wr command with a data argument of 2'h2 (2). Send the PHA_En1_Rd command and read back this commanded data argument.
4. Send the PHA_En1_Wr command with a data argument of 2'h3 (3). Send the PHA_En1_Rd command and read back this commanded data argument.

9.18 Veto En0 Register Test

1. Send the VETO_En0_Wr command with a data argument of 16'h0 (0). Send the VETO_En0_Rd command and read back this commanded data argument.
2. Send the VETO_En0_Wr command with a data argument of 16'h5555 (21845). Send the VETO_En0_Rd command and read back this commanded data argument.
3. Send the VETO_En0_Wr command with a data argument of 16'hAAAA (43690). Send the VETO_En0_Rd command and read back this commanded data argument.
4. Send the VETO_En0_Wr command with a data argument of 16'hFFFF (65535). Send the VETO_En0_Rd command and read back this commanded data argument.

9.19 Veto En1 Register Test

1. Send the VETO_En1_Wr command with a data argument of 2'h0 (0). Send the VETO_En1_Rd command and read back this commanded data argument.
2. Send the VETO_En1_Wr command with a data argument of 2'h1 (1). Send the VETO_En1_Rd command and read back this commanded data argument.
3. Send the VETO_En1_Wr command with a data argument of 2'h2 (2). Send the VETO_En1_Rd command and read back this commanded data argument.
4. Send the VETO_En1_Wr command with a data argument of 2'h3 (3). Send the VETO_En1_Rd command and read back this commanded data argument.

9.20 MaxPHA Register Test

1. Send the MaxPHA_Wr command with a data argument of 5'h0 (0). Send the MaxPHA_Rd command and read back this commanded data argument.
2. Send the MaxPHA_Wr command with a data argument of 5'h15 (21). Send the MaxPHA_Rd command and read back this commanded data argument.
3. Send the MaxPHA_Wr command with a data argument of 5'h0A (10). Send the MaxPHA_Rd command and read back this commanded data argument.
4. Send the MaxPHA_Wr command with a data argument of 5'h1F (31). Send the MaxPHA_Rd command and read back this commanded data argument.
5. Send the MaxPHA_Wr command with a data argument of 5'h4 (4). Send the MaxPHA_Rd command and read back this commanded data argument.

9.21 GARC Mode Register Test

**** Note this sequence must be followed exactly to preclude placing the GARC into the undesired mode of sending return data with incorrect parity.**

1. Send the GARC_Mode_Wr command with a data argument of 11'h2 (2). Send the GARC_Mode_Rd command and read back this commanded data argument.
2. Send the GARC_Mode_Wr command with a data argument of 11'h4 (4). Send the GARC_Mode_Rd command and read back this commanded data argument.
3. Send the GARC_Mode_Wr command with a data argument of 11'h8 (8). Send the GARC_Mode_Rd command and read back this commanded data argument.

4. Send the GARC_Mode_Wr command with a data argument of 11'h10 (16). Send the GARC_Mode_Rd command and read back this commanded data argument.
5. Send the GARC_Mode_Wr command with a data argument of 11'h20 (32). Send the GARC_Mode_Rd command and read back this commanded data argument.
6. Send the GARC_Mode_Wr command with a data argument of 11'h40 (64). Send the GARC_Mode_Rd command and read back this commanded data argument.
7. Send the GARC_Mode_Wr command with a data argument of 11'h80 (128). Send the GARC_Mode_Rd command and read back this commanded data argument.
8. Send the GARC_Mode_Wr command with a data argument of 11'h100 (256). Send the GARC_Mode_Rd command and read back this commanded data argument.
9. Send the GARC_Mode_Wr command with a data argument of 11'h200 (512). Send the GARC_Mode_Rd command and read back this commanded data argument.
10. Send the GARC_Mode_Wr command with a data argument of 11'h400 (1024). Send the GARC_Mode_Rd command and read back this commanded data argument.
11. Send the GARC_Mode_Wr command with a data argument of 11'h300 (768). Send the GARC_Mode_Rd command and read back this commanded data argument.

9.22 PHA Threshold Channel 00 Register Test

1. Send the PHA_Thresh00_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh00_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh00_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh00_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh00_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh00_Rd command and read back this commanded data argument.

9.23 PHA Threshold Channel 01 Register Test

1. Send the PHA_Thresh01_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh01_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh01_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh01_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh01_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh01_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh01_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh01_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh01_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh01_Rd command and read back this commanded data argument.

9.24 PHA Threshold Channel 02 Register Test

1. Send the PHA_Thresh02_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh02_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh02_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh02_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh02_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh02_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh02_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh02_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh02_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh02_Rd command and read back this commanded data argument.

9.25 PHA Threshold Channel 03 Register Test

1. Send the PHA_Thresh03_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh03_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh03_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh03_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh03_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh03_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh03_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh03_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh03_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh03_Rd command and read back this commanded data argument.

9.26 PHA Threshold Channel 04 Register Test

1. Send the PHA_Thresh04_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh04_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh04_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh04_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh04_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh04_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh04_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh04_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh04_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh04_Rd command and read back this commanded data argument.

9.27 PHA Threshold Channel 05 Register Test

1. Send the PHA_Thresh05_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh05_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh05_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh05_Rd command and read back this commanded data argument.

3. Send the PHA_Thresh05_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh05_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh05_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh05_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh05_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh05_Rd command and read back this commanded data argument.

9.28 PHA Threshold Channel 06 Register Test

1. Send the PHA_Thresh06_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh06_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh06_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh06_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh06_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh06_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh06_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh06_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh06_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh06_Rd command and read back this commanded data argument.

9.29 PHA Threshold Channel 07 Register Test

1. Send the PHA_Thresh07_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh07_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh07_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh07_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh07_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh07_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh07_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh07_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh07_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh07_Rd command and read back this commanded data argument.

9.30 PHA Threshold Channel 08 Register Test

1. Send the PHA_Thresh08_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh08_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh08_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh08_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh08_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh08_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh08_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh08_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh08_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh08_Rd command and read back this commanded data argument.

9.31 PHA Threshold Channel 09 Register Test

1. Send the PHA_Thresh09_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh09_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh09_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh09_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh09_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh09_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh09_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh09_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh09_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh09_Rd command and read back this commanded data argument.

9.32 PHA Threshold Channel 10 Register Test

1. Send the PHA_Thresh10_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh10_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh10_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh10_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh10_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh10_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh10_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh10_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh10_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh10_Rd command and read back this commanded data argument.

9.33 PHA Threshold Channel 11 Register Test

1. Send the PHA_Thresh11_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh11_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh11_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh11_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh11_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh11_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh11_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh11_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh11_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh11_Rd command and read back this commanded data argument.

9.34 PHA Threshold Channel 12 Register Test

1. Send the PHA_Thresh12_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh12_Rd command and read back this commanded data argument.

2. Send the PHA_Thresh12_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh12_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh12_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh12_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh12_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh12_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh12_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh12_Rd command and read back this commanded data argument.

9.35 PHA Threshold Channel 13 Register Test

1. Send the PHA_Thresh13_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh13_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh13_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh13_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh13_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh13_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh13_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh13_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh13_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh13_Rd command and read back this commanded data argument.

9.36 PHA Threshold Channel 14 Register Test

1. Send the PHA_Thresh14_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh14_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh14_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh14_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh14_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh14_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh14_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh14_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh14_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh14_Rd command and read back this commanded data argument.

9.37 PHA Threshold Channel 15 Register Test

1. Send the PHA_Thresh15_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh15_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh15_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh15_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh15_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh15_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh15_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh15_Rd command and read back this commanded data argument.

5. Send the PHA_Thresh15_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh15_Rd command and read back this commanded data argument.

9.38 PHA Threshold Channel 16 Register Test

1. Send the PHA_Thresh16_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh16_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh16_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh16_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh16_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh16_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh16_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh16_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh16_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh16_Rd command and read back this commanded data argument.

9.39 PHA Threshold Channel 17 Register Test

1. Send the PHA_Thresh17_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh17_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh17_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh17_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh17_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh17_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh17_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh17_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh17_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh17_Rd command and read back this commanded data argument.

9.40 ADC TACQ Register Test

1. Send the ADC_TACQ_Wr command with a data argument of 6'h0 (0). Send the ADC_TACQ_Rd command and read back this commanded data argument.
2. Send the ADC_TACQ_Wr command with a data argument of 6'h15 (21). Send the ADC_TACQ_Rd command and read back this commanded data argument.
3. Send the ADC_TACQ_Wr command with a data argument of 6'h2A (42). Send the ADC_TACQ_Rd command and read back this commanded data argument.
4. Send the ADC_TACQ_Wr command with a data argument of 6'h3F (63). Send the ADC_TACQ_Rd command and read back this commanded data argument.
5. Send the ADC_TACQ_Wr command with a data argument of 6'h0 (0). Send the ADC_TACQ_Rd command and read back this commanded data argument.
6. Send the GARC_Reset command to ensure all registers are at the proper initial value.

At the successful conclusion of this section, the GARC commandable register bits have been demonstrated to be functional with all bits toggling as commanded.

9.41 Test of the GARC Return Data Parity

This section tests the proper functioning of the GARC return data parity select bit.

1. Send the GARC_Status command. The data field in the return data stream should be decimal 24.
2. Send the GARC_Mode command with a data argument of decimal 769, a command to return event data with even parity.
3. Send the GARC_Status command. The data field in the return data stream should show a parity error.
4. Send the GARC_Mode command with a data argument of decimal 768, a command back to odd parity, the nominal mode.
5. Send the GARC_Status command. The data field in the return data stream should be decimal 24.

9.42 Test of the HVBS Triple Modular Redundancy Circuitry

This section tests the proper functioning of the GARC HVBS enable circuitry. Each pattern in the TMR logic will be tested to verify proper recovery from a single event upset. The GARC HV_ENABLE_1 pin is 185 and the HV_ENABLE_2 pins is 186. A truth table for proper TMR circuitry function is detailed below.

Enable Bit A	Enable Bit B	Enable Bit C	HV Enable Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

1. Send the GARC_Mode_Wr command with a data argument of 768 and verify the data with the GARC_Mode_Rd command.
2. Send the GARC_Status command. The data field in the return data stream should show that bits 1 and 2 are 0, indicating that HVBS 1 and 2, respectively, are disabled. Verify that GARC pins 185 and 186 are at 0V.
3. Send the GARC_Mode_Wr command with decimal argument 770 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is disabled. Verify that GARC pins 185 and 186 are at 0V.
4. Send the GARC_Mode_Wr command with decimal argument 772 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is disabled. Verify that GARC pins 185 and 186 are at 0V.
5. Send the GARC_Mode_Wr command with decimal argument 774 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is enabled and HVBS 2 is disabled. Verify that GARC pins 185 is at 3.3V and pin 186 is at 0V.

6. Send the GARC_Mode_Wr command with decimal argument 776 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is disabled. Verify that GARC pins 185 is at 0V and pin 186 is at 0V.
7. Send the GARC_Mode_Wr command with decimal argument 778 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is enabled and HVBS 2 is disabled. Verify that GARC pins 185 is at 3.3V and pin 186 is at 0V.
8. Send the GARC_Mode_Wr command with decimal argument 780 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is enabled and HVBS 2 is disabled. Verify that GARC pins 185 is at 3.3V and pin 186 is at 0V.
9. Send the GARC_Mode_Wr command with decimal argument 782 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is enabled and HVBS 2 is disabled. Verify that GARC pins 185 is at 3.3V and pin 186 is at 0V.
10. Send the GARC_Mode_Wr command with decimal argument 784 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is disabled. Verify that GARC pins 185 is at 0V and pin 186 is at 0V.
11. Send the GARC_Mode_Wr command with decimal argument 800 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is disabled. Verify that GARC pins 185 is at 0V and pin 186 is at 0V.
12. Send the GARC_Mode_Wr command with decimal argument 816 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is enabled. Verify that GARC pins 185 is at 0V and pin 186 is at 3.3V.
13. Send the GARC_Mode_Wr command with decimal argument 832 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is disabled. Verify that GARC pins 185 is at 0V and pin 186 is at 0V.
14. Send the GARC_Mode_Wr command with decimal argument 848 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is enabled. Verify that GARC pins 185 is at 0V and pin 186 is at 3.3V.
15. Send the GARC_Mode_Wr command with decimal argument 864 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is enabled. Verify that GARC pins 185 is at 0V and pin 186 is at 3.3V.
16. Send the GARC_Mode_Wr command with decimal argument 880 and verify the data with the GARC_Mode_Rd command. Send the GARC_Status command and verify that HVBS 1 is disabled and HVBS 2 is enabled. Verify that GARC pins 185 is at 0V and pin 186 is at 3.3V.
17. Send the GARC_Reset command. Verify that GARC pins 185 is at 0V and pin 186 is at 0V.

This completes the test of the HVBS enable/disable TMR circuitry.

9.43 Test of the Look-At-Me Circuitry

This section tests the proper functioning of the GARC Look-At-Me circuitry. The GARC has the capability to receive commands from either the primary side or secondary side. The selection of which set of receivers to listen to is controlled by the Look-At-Me circuitry. This circuitry toggles the status of the receiving side based upon receipt of a special command pattern (e.g., **34'h24153D721**).

1. Verify that the AEM (or AEM simulator) is connected to the GARC primary side interface (e.g., the "A" side clock and data). Send the GARC_Status command. The data field in the return data stream

should be decimal 24. Bit 0 (LSB) of the Status register represents the Look-At-Me status (0 = A, 1 = B).

2. Disconnect the interface from the primary side and move the interface to the secondary side. Send the GARC_Status command. Observe that there is no response from the GARC.
3. Send the Look_At_Me command to the GARC from the secondary side interface. Send the GARC_Status command. The data field in the return data should be decimal 25 (with the LSB = 1, indicating the Look-At-Me status is B side).
4. Disconnect the interface from the secondary side and move the interface to the primary side. Send the GARC_Status command. Observe that there is no response from the GARC.
5. Send the Look_At_Me command to the GARC from the primary side interface. Send the GARC_Status command. The data field in the return data should be decimal 24 (with the LSB = 0, indicating the Look-At-Me status is A side).

This concludes the test of the GARC Look-At-Me circuitry.

9.44 Test of the GAFE Parity Command Circuitry

This section tests the proper functioning of the GAFE parity command circuitry. At least one GAFE ASIC needs to be connected to the GARC under test to perform this section. GAFE commands must be sent to the same address as the hardwired GAFE address. The GAFE logic nominally expects odd parity. When the GARC command to change the GAFE command message to even parity is executed, the GAFE should no longer respond to messages from the GARC. When the GAFE parity is again commanded to be odd, nominal response should resume.

1. Send the GARC_Mode_Wr command with a decimal data argument of 768 (0x300). Verify this command with the GARC_Mode_Rd command.
2. Send the GAFE command GAFE_Mode_Rd and verify that the GAFE logic responds with the contents of the mode register.
3. Send the GARC_Mode_Wr command with a decimal data argument of 896 (0x380). Verify this command with the GARC_Mode_Rd command.
4. Send the GAFE command GAFE_Mode_Rd and verify that the GAFE logic does not respond to this command (e.g., the command is rejected due to incorrect parity).
5. Send the GARC_Mode_Wr command with a decimal data argument of 768 (0x300). Verify this command with the GARC_Mode_Rd command.
6. Send the GAFE command GAFE_Mode_Rd and verify that the GAFE logic responds with the contents of the mode register.

9.45 FREE Board ID Circuit Test

This section tests the proper capture of the FREE circuit card serial number identification by the GARC logic. The FREE circuit ID is operated during a PHA conversion, which may be initiated via a trigger command.

1. Set the hardwired FREE board address to decimal 0. Send the Trigger_NOZS command to the GARC to initiate a PHA conversion. Send the FREE_Board_ID command and look for the address value to be returned in the GARC readback data.

2. Set the hardwired FREE board address to decimal 255. Send the Trigger_NOZS command to the GARC to initiate a PHA conversion. Send the FREE_Board_ID command and look for the address value to be returned in the GARC readback data.
3. Set the hardwired FREE board address to decimal 85. Send the Trigger_NOZS command to the GARC to initiate a PHA conversion. Send the FREE_Board_ID command and look for the address value to be returned in the GARC readback data.
4. Set the hardwired FREE board address to decimal 170. Send the Trigger_NOZS command to the GARC to initiate a PHA conversion. Send the FREE_Board_ID command and look for the address value to be returned in the GARC readback data.

This completes the test of the FREE board ID circuitry.

9.46 Maximum PHA Return Test

This section tests the proper functioning of the event data processor as it handles the Maximum Number of PHA words command.

1. Send the Max_PHA_Wr command with a data argument of 18. Send the Max_PHA_Rd command and verify that the data value has been set. Send the Trigger_NOZS command and verify that the event data processor sends back 18 PHA words.
2. Repeat the previous step with a data argument of 17.
3. Repeat the previous step with a data argument of 16.
4. Repeat the previous step with a data argument of 15.
5. Repeat the previous step with a data argument of 14.
6. Repeat the previous step with a data argument of 13.
7. Repeat the previous step with a data argument of 12.
8. Repeat the previous step with a data argument of 11.
9. Repeat the previous step with a data argument of 10.
10. Repeat the previous step with a data argument of 9.
11. Repeat the previous step with a data argument of 8.
12. Repeat the previous step with a data argument of 7.
13. Repeat the previous step with a data argument of 6.
14. Repeat the previous step with a data argument of 5.
15. Repeat the previous step with a data argument of 4.
16. Repeat the previous step with a data argument of 3.
17. Repeat the previous step with a data argument of 2.
18. Repeat the previous step with a data argument of 1.
19. Repeat the previous step with a data argument of 0.
20. Send the GARC_Reset command to restore the Max_PHA number to the default. This completes the test of the GARC MAX_PHA function.

9.47 PHA Enable/Disable Test

This section tests the proper functioning of the event data processor as it handles the PHA enables and disabled in selection of PHA words for transmission. For this section, the GAFE simulator shall be programmed such that each of the 18 simulated PHA channels has a uniquely identifying PHA value (e.g., for example, related to the GAFE channel number)

1. Send the PHA_En0_Wr command with a data argument of decimal 'hFFFF (65535). Verify this value with the PHA_En0_Rd command. This enables channels 0 – 15.
2. Send the PHA_En1_Wr command with a data argument of decimal 3. Verify this value with the PHA_En1_Rd command. This enables channels 16 and 17.
3. Send the Trigger_NOZS command. The event data processor should respond with 18 PHA words during event readout.
4. Send the PHA_En0_Wr command with a data argument of 'hFFFD (65534). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 0.
5. Send the PHA_En0_Wr command with a data argument of 'hFFFB (65533). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 1.
6. Send the PHA_En0_Wr command with a data argument of 'hFFF7 (65531). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 2.
7. Send the PHA_En0_Wr command with a data argument of 'hFFF7 (65527). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 3.
8. Send the PHA_En0_Wr command with a data argument of 'hFFEF (65519). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 4.
9. Send the PHA_En0_Wr command with a data argument of 'hFFDF (65503). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 5.
10. Send the PHA_En0_Wr command with a data argument of 'hFFBF (65471). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 6.
11. Send the PHA_En0_Wr command with a data argument of 'hFF7F (65407). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 7.
12. Send the PHA_En0_Wr command with a data argument of 'hFFEF (65279). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 8.
13. Send the PHA_En0_Wr command with a data argument of 'hFCFF (64767). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 9.
14. Send the PHA_En0_Wr command with a data argument of 'hFBFF (64511). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 10.

15. Send the PHA_En0_Wr command with a data argument of 'hF7FF (63487). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 11.
16. Send the PHA_En0_Wr command with a data argument of 'hEFFF (61439). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 12.
17. Send the PHA_En0_Wr command with a data argument of 'hCFFF (53247). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 13.
18. Send the PHA_En0_Wr command with a data argument of 'hBFFF (49151). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 14.
19. Send the PHA_En0_Wr command with a data argument of 'h7FFF (32767). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 15.
20. Send the PHA_En0_Wr command with a data argument of 'hFFFF (65535). Verify this value with the PHA_En0_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words.
21. Send the PHA_En1_Wr command with a data argument of 2. Verify this value with the PHA_En1_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 16.
22. Send the PHA_En1_Wr command with a data argument of 1. Verify this value with the PHA_En1_Rd command. Send the Trigger_NOZS command. The event processor should respond with all PHA words except channel 17.
23. Send the GARC_Reset command to reinitialize all GARC registers. This completes the testing of the PHA enable and disable register bits.

9.48 PHA Threshold Verification Test

This section tests the proper functioning of each of the 18 PHA thresholding circuits. The intent of the circuits is such that the ZS Map bit is set active high for a given PHA word if either the range bit is high or the 12 bit PHA word is greater than the PHA threshold. This test requires a GAFE simulator test board that is programmed to emulate each of the possible PHA patterns, decimal 0 through 8191 (range bit + 12 bits ADC conversion). This test needs to be run in an automated format due to the very large number of possibilities. A suggestion for the proper functionality is detailed below. The Max_PHA should be set to 18 so that none of the PHA words are suppressed by that function.

```
for (pha_word = 0; pha_word < 8192 ; pha_word++)
     for (pha_threshold = 0; pha_threshold < 65536; pha_threshold++)
        for (GAFE_channel = 0; GAFE_channel < 18; GAFE_channel++)
           {
              Send Trigger_ZS command;
              Capture the 18 bit HitMap word;
              If HitMap[GAFE_channel] != range_bit | (pha_word > pha_threshold)
                 GARC_Error;
           }
```

During this test, the proper PHA values are checked for return in the event data. For the duration of this test, the PHA value input to each channel up to 17 should be an increment of 1 more than the previous channel so that each channel has a unique PHA identifier (for the first 17 pha_words, some channels will have a zero

value multiple times). It is important to have a unique identifier for PHA words to verify proper position in the event data stream. The ZS Map needs to be compared to the PHA in the event data stream to ensure a match.

At the conclusion of this section of the test, send the GARC_Reset command to initialize the GARC registers. This completes the PHA Threshold Verification section of the procedure.

9.49 Test of the GARC Diagnostic Status Register

1. Send the GARC_Reset command to initialize the GARC registers.
2. Send the GARC_Mode_Rd command. The data field in the return data stream should be decimal 768.
3. Send the GARC_Status command. The data field in the return data stream should be decimal 24.
4. Send the GARC_Cmd_Reg command. The data field in the return data stream should be decimal 0.
5. Send the GARC_Diagnostic command. The most significant four bits in the return data stream should be decimal 0. The remaining 12 bits are state machine loop and command counters and the total should not be zero.

9.50 AEM Command Parity Error Simulation Test

This test generates simulated AEM command word errors and verifies that the GARC responds in the appropriate manner. In this section of the test, the AEM (or AEM simulator) must send a GARC command with an error in the command addressing portion of the command word. For example,

A correct (odd parity) Calib command has the format: 34'h2404E0001.

A Calib command with even command parity has the format: 34'h2404C0001.

1. Send the GARC_Reset command to reinitialize all GARC registers.
2. Send the GARC_Diagnostic command and verify that bits 15:12 in the return data word are 4'b0000.
3. Send the GARC_Cal_Strobe command, 34'h2404E0001.
4. Send the GARC_Diagnostic command and verify that bits 15:12 in the return data word are 4'b0000.
5. Send the Command_Register command and verify that the return data is 16'h0000.
6. Send the even command parity (i.e., error) command: 34'h2404C0001.
7. Send the GARC_Diagnostic command and verify that bits 15:12 in the return data word are 4'b1101.
8. Send the Command_Register command and verify that the return data is 16'h404E.

9.51 AEM Data Parity Error Simulation Test

This test generates simulated AEM data word errors and verifies that the GARC responds in the appropriate manner. In this section of the test, the AEM (or AEM simulator) must send a GARC command with an error in the command addressing portion of the command word. For example,

A correct (odd parity) Calib command has the format: 34'h2404E0001.

A Calib command with even data parity could have the format: 34'h2404E00081.

1. Send the GARC_Reset command to reinitialize all GARC registers.
2. Send the GARC_Diagnostic command and verify that bits 15:12 in the return data word are 4'b0000.
3. Send the GARC_Cal_Strobe command, 34'h2404E0001.
4. Send the GARC_Diagnostic command and verify that bits 15:12 in the return data word are 4'b0000.
5. Send the Command_Register command and verify that the return data is 16'h0000.
6. Send the even command parity (i.e., error) command: 34'h2404E00081.
7. Send the GARC_Diagnostic command and verify that bits 15:12 in the return data word are 4'b1011.
8. Send the Command_Register command and verify that the return data is 16'h0008.

9.52 Command Counter Test

1. Send the GARC_Reset command to initialize the GARC registers.
2. Send the GARC_Diagnostic command. The value in bits 7:0 of the returned data word should have the value 1.
3. Send the GARC_Version command ten times.
4. Send the GARC_Diagnostic command. The value in bits 7:0 of the returned data word should have the value 11.

9.53 GARC Diagnostic State Loop Counter Test

1. Send the GARC_Reset command to initialize the GARC registers.
2. Send the GARC_Diagnostic command. The value in bits 11:8 of the returned data word should have the value 1.
3. Send the GARC_Version command ten times.
4. Send the GARC_Diagnostic command. The value in bits 11:8 of the returned data word should have the value 11.

9.54 GAFE Interface Test

This section tests the proper interface of the GARC to the GAFE ASIC. At least one GAFE ASIC must be present on the board being tested. This section will test the GAFE Hold, Strobe, Command Data, and Return Data.

1. Send the Trig_NOZS command. Monitor the GAFE_HOLD signal differentially on the GARC (GAFE_HOLDP on pin 161 and GAFE_HOLDM on pin 162) and verify TBD voltage across the termination resistor.
2. Send the GARC_Cal_Strobe command. Monitor the STROBE signal to the GAFE differentially on the GARC (GAFE_STROBEP on pin 163 and GAFE_STROBEM on pin 164). Verify TBD voltage across the termination resistor.
3. Send the GAFE_Version command to a valid GAFE address. The data in the readback should indicate GAFE version 2. This verifies the GAFE command and return data paths.

4. Send the GARC_Reset command to reinitialize the GARC ASIC. Additional testing of the GAFE logic functions may be performed during the GAFE functional test.

10.0 Testing the GAFE Logic for each ASIC Connected to the GARC

At this point in the GARC functional test, functional testing of the logic for any GAFEs connected to the GARC may be performed. A minimum of one GAFE ASIC (or the GAFE logic simulator) must be connected for this test. If multiple GAFE logic cores are to be tested concurrently, such as the nominal condition for a populated FREE circuit card, then each address may be tested in sequence, replicating the steps detailed in the GAFE logic test procedure. Each GAFE is independent and the commands for each test may be performed either in series or in parallel up to a total of 18 GAFE ASICs.

10.1 Initial GAFE Logic Reset Test

After initial power up or a GARC Reset command is sent, the GAFE registers should be initialized. This is tested by the following command sequence (sent to each GAFE being tested at the unique five bit address for that GAFE).

1. Send the GARC_Reset command.
2. Send the GAFE_Mode_Read command. The data field in the return data stream should be 16'h30 (48).
3. Send the GAFE_DAC1_Read command. The data field in the return data stream should be 16'h39 (57).
4. Send the GAFE_DAC2_Read command. The data field in the return data stream should be 16'h26 (38).
5. Send the GAFE_DAC3_Read command. The data field in the return data stream should be 16'h37 (55).
6. Send the GAFE_DAC4_Read command. The data field in the return data stream should be 16'h20 (32).
7. Send the GAFE_DAC5_Read command. The data field in the return data stream should be 16'h00 (0).
8. Send the GAFE_Version command. The data field in the return data stream should be 16'h02 (2).
9. Send the GAFE_Write_Ctr command. The data field in the return data stream should be 16'h00 (0).
10. Send the GAFE_Reject_Ctr command. The data field in the return data stream should be 16'h00 (0).
11. Send the GAFE_Cmd_Ctr command. The data field in the return data stream should be 16'h0A (10).
12. Send the GAFE_Chip_Address command. The data field in the return data stream should be the address of the chip that was commanded.

10.2 GAFE ASICs Mode Register Test

This section tests the proper functioning of the GAFE mode register. Repeat for all GAFE ASICs present using valid GAFE addressing.

1. Send the GAFE_Mode_Write command to the GAFE being tested with a data field of 16'h00 (0).

2. Send the GAFE_Mode_Read command. The data field in the return data stream should be 16'h00 (0).
3. Send the GAFE_Mode_Write command to the GAFE being tested with a data field of 16'hFF (255).
4. Send the GAFE_Mode_Read command. The data field in the return data stream should be 16'hFF (255).
5. Send the GAFE_Mode_Write command to the GAFE being tested with a data field of 16'h30 (48).
6. Send the GAFE_Mode_Read command. The data field in the return data stream should be 16'h30 (48).

10.3 GAFE ASIC DAC1 Register Test

This section tests the proper functioning of the GAFE DAC #1 register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_DAC1_Write command to the GAFE being tested with a data field of 16'h00 (0).
2. Send the GAFE_DAC1_Read command. The data field in the return data stream should be 16'h00 (0).
3. Send the GAFE_DAC1_Write command to the GAFE being tested with a data field of 16'h3F (63).
4. Send the GAFE_DAC1_Read command. The data field in the return data stream should be 16'h3F (63).
5. Send the GAFE_DAC1_Write command to the GAFE being tested with a data field of 16'h39 (57).
6. Send the GAFE_DAC1_Read command. The data field in the return data stream should be 16'h39 (57).

10.4 GAFE ASIC DAC2 Register Test

This section tests the proper functioning of the GAFE DAC #2 register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_DAC2_Write command to the GAFE being tested with a data field of 16'h00 (0).
2. Send the GAFE_DAC2_Read command. The data field in the return data stream should be 16'h00 (0).
3. Send the GAFE_DAC2_Write command to the GAFE being tested with a data field of 16'h3F (63).
4. Send the GAFE_DAC2_Read command. The data field in the return data stream should be 16'h3F (63).
5. Send the GAFE_DAC2_Write command to the GAFE being tested with a data field of 16'h26 (38).
6. Send the GAFE_DAC2_Read command. The data field in the return data stream should be 16'h26 (38).

10.5 GAFE ASIC DAC3 Register Test

This section tests the proper functioning of the GAFE DAC #3 register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_DAC3_Write command to the GAFE being tested with a data field of 16'h00 (0).
2. Send the GAFE_DAC3_Read command. The data field in the return data stream should be 16'h00 (0).
3. Send the GAFE_DAC3_Write command to the GAFE being tested with a data field of 16'h3F (63).
4. Send the GAFE_DAC3_Read command. The data field in the return data stream should be 16'h3F (63).
5. Send the GAFE_DAC3_Write command to the GAFE being tested with a data field of 16'h37 (55).
6. Send the GAFE_DAC3_Read command. The data field in the return data stream should be 16'h37 (55).

10.6 GAFE ASIC DAC4 Register Test

This section tests the proper functioning of the GAFE DAC #4 register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_DAC4_Write command to the GAFE being tested with a data field of 16'h00 (0).
2. Send the GAFE_DAC4_Read command. The data field in the return data stream should be 16'h00 (0).
3. Send the GAFE_DAC4_Write command to the GAFE being tested with a data field of 16'h3F (63).
4. Send the GAFE_DAC4_Read command. The data field in the return data stream should be 16'h3F (63).
5. Send the GAFE_DAC4_Write command to the GAFE being tested with a data field of 16'h20 (32).
6. Send the GAFE_DAC4_Read command. The data field in the return data stream should be 16'h20 (32).

10.7 GAFE ASIC DAC5 Register Test

This section tests the proper functioning of the GAFE DAC #5 register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_DAC5_Write command to the GAFE being tested with a data field of 16'h00 (0).
2. Send the GAFE_DAC5_Read command. The data field in the return data stream should be 16'h00 (0).
3. Send the GAFE_DAC5_Write command to the GAFE being tested with a data field of 16'h3F (63).
4. Send the GAFE_DAC5_Read command. The data field in the return data stream should be 16'h3F (63).
5. Send the GAFE_DAC5_Write command to the GAFE being tested with a data field of 16'h00 (0).
6. Send the GAFE_DAC5_Read command. The data field in the return data stream should be 16'h00 (0).

10.8 GAFE ASIC Diagnostics Verification

This section tests the proper functioning of the diagnostics command counters in the GAFE. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_Reject_Ctr command to the GAFE being tested. The data field in the return data stream should be 16'h00 (0).
2. Send the GAFE_Write_Ctr command. The data field in the return data stream should be 'h12 (18).
3. Send the GAFE_Cmd_Ctr command. The data field in the return data stream should be 'h32 (50).

10.9 GAFE ASIC Broadcast Command Functional Verification

This section tests the proper response to broadcast commands (e.g., address 31) by the GAFEs.

1. Send the GAFE_Mode_Read command with a broadcast address, i.e., address 'h1F (31) , to all the GAFEs being tested. There should be no data returned from any GAFE.
2. Send the GAFE_Mode_Write command with a broadcast address and a data field of 'hAA (170) to the GAFEs.
3. Send a GAFE_Mode_Read address command to each GAFE under test. The return data field from each addressed GAFE should be 'hAA (170).
4. Send a GARC_Reset command to reinitialize each of the GAFE logic cores.

11.0 FREE Verification Measurements Requiring an Oscilloscope and Multimeter

This section continues the verification of the FREE circuit card assembly. The measurements in this section require manual intervention due to the necessity for a measurement with the oscilloscope and/or multimeter.

11.1 Test of the HVBS DAC Commands, DAC Buffer, and Differential Drivers

The pin connections for the two 79 pin circular connectors JP1 and JS2 are listed in Appendix 1. The pin connections for the two 9 pin MDM connectors to the High Voltage Bias Supplies are listed in Appendix 2.

Prior to starting this test, the enable pins should be removed from the HVBS breakout boxes (these are pins JHV1-5 and JHV2-5) to prevent the bias supplies from delivering high voltage during the DAC test.

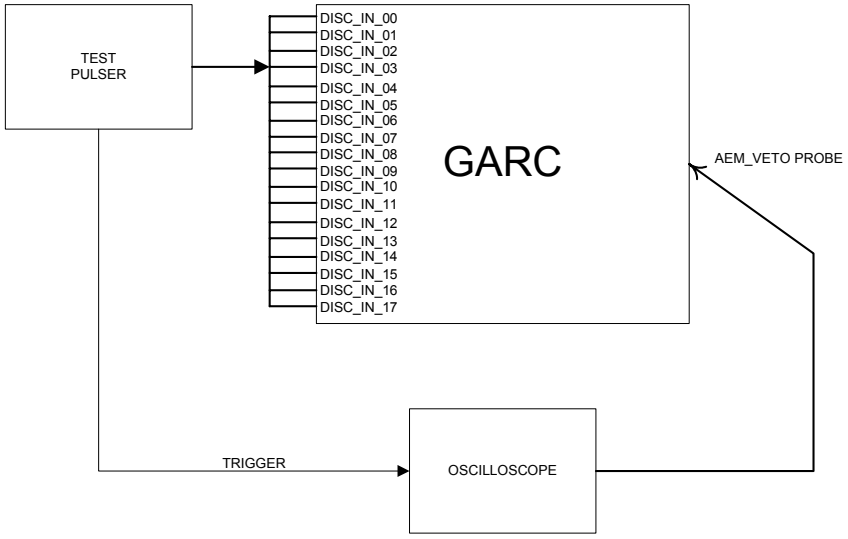
11.2 GARC Test Pin Mux Verification

This section tests the proper functionality, switch the GARC test pin between the "Live" and "HitMap Test" outputs.

1. Send the GARC_Mode_Wr command with a data argument of 1792 to switch the GARC test pin mux to view the "live" signal. Send the GARC_Mode_Rd command to verify the register data. When configuration commands are sent, the "live" signal should go inactive low during the state machine busy time. This may be verified via the oscilloscope. The GARC test pin is pin 179.
2. Send the GARC_Mode_Wr command with a data argument of 768 to switch the GARC test pin mux to view the "HitMap test" signal. Send the GARC_Mode_Rd command to verify the register data. When trigger commands are sent, the "HitMap test" signal should go active high during times when there is a delayed VETO signal present at the DISC_IN input. This may be verified via the oscilloscope. The GARC test pin is pin 179.
3. Send the GARC_Reset command to ensure all registers are at the proper initial value.

11.3 Test of the AEM VETO Signal Functionality

This section tests the proper functioning of the commandable functions of the AEM_VETO signals. This test requires the use of an oscilloscope, a test pulser for input stimulus, and test points to monitor the AEM_VETOs. The test setup is as shown below.

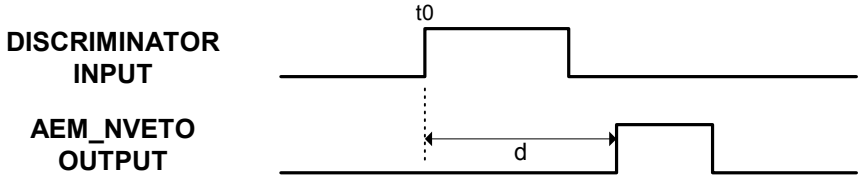


GARC VETO Test Setup

All 18 of the discriminator inputs are tied together and also to the output of the test pulser for this portion of the test. The input test pulse should be approximately 200 ns wide and the rate should be set at approximately 20 kHz. The oscilloscope probe will be monitoring the 18 AEM_VETO outputs. Between commands, this probe will be moved manually from test point to test point to perform this test. Setting the scope to acquire on the pulse output, this test will be measuring the delay (time from trigger to VETO leading edge) and width (time between leading and trailing edges) of each of the VETO pulses. Since each input has the same signal, the outputs should also be common in delay and width. In timing measurements, the activation signal from the test pulser will be the relative zero.

11.4 Veto Delay Test

1. Send the Veto_Delay_Wr command with a data argument of 0. Confirm this command via the Veto_Delay_Rd command.
2. Send the Veto_Width_Wr command with a data argument of 0. Confirm this command via the Veto_Width_Rd command. Using the oscilloscope, measure the delay from scope trigger to leading edge of VETO for each of the 18 pulses and record the data in the table below. Note that all 18 pulses are designed to have identical timing for this test. The VETO Delay is the time **d** in the diagram below.



VETO DELAY MEASUREMENT

3. Repeat the Veto_Delay_Wr, Veto_Delay_Rd sequence for data argument values of decimal 1 to 31. Record the results in the table below. Note that there is a 50 ns jitter on each of the "expected" numbers.

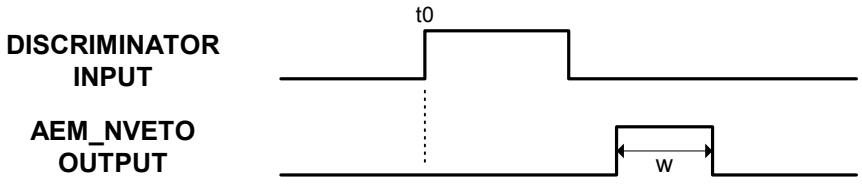
Veto_Delay Setting	Delay from Scope trigger to Leading Edge (ns)	Expected (ns)
0		150

1		200
2		250
3		300
4		350
5		400
6		450
7		500
8		550
9		600
10		650
11		700
12		750
13		800
14		850
15		900
16		950
17		1000
18		1050
19		1100
20		1150
21		1200
22		1250
23		1300
24		1350
25		1400
26		1450
27		1500
28		1550
29		1600
30		1650
31		1700

4. Send the Veto_Delay_Wr command with a data argument of 0 and a Veto_Delay_Rd command to verify this value.

11.5 Veto Width Test

1. Send the Veto_Width_Wr command with a data argument of 0. Confirm this command via the Veto_Width_Rd command. Using the oscilloscope, measure the width of each of the 18 VETO pulses and record the data in the table below. Note that the width of each of these pulses is expected to be identical for any given commanded setting. The VETO Width is the "**w**" parameter in the diagram below.



VETO WIDTH MEASUREMENT

2. Repeat the Veto_Width_Wr, Veto_Width_Rd sequence for data argument values of decimal 1 to 7. Record the results in the table below.

Veto_Width	Measured Width of VETO Pulse (ns)	Expected Width (ns)
0		50
1		100
2		150
3		200
4		250
5		300
6		350
7		400

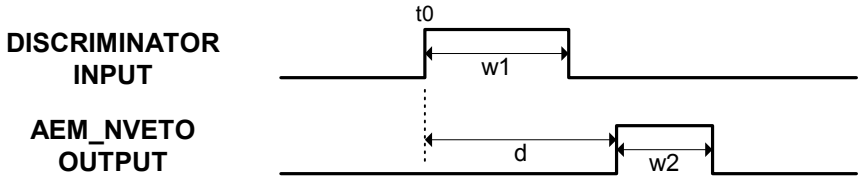
11.6 Veto Enable and Disable Test

1. Send the VETO_En0_Rd command and verify that the return data has value ‘hFFFF (65535). Send the VETO_En1_Rd command and verify that the return data has the value 3.
2. Send the VETO_En0_Wr command with a data argument ‘hFFFE (65534). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 0.
3. Send the VETO_En0_Wr command with a data argument ‘hFFFD (65533). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 1.
4. Send the VETO_En0_Wr command with a data argument ‘hFFFB (65531). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 2.
5. Send the VETO_En0_Wr command with a data argument ‘hFFF7 (65527). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 3.
6. Send the VETO_En0_Wr command with a data argument ‘hFFEF (65519). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 4.
7. Send the VETO_En0_Wr command with a data argument ‘hFFDF (65503). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 5.
8. Send the VETO_En0_Wr command with a data argument ‘hFFBF (65471). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 6.
9. Send the VETO_En0_Wr command with a data argument ‘hFF7F (65407). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 7.
10. Send the VETO_En0_Wr command with a data argument ‘hFFEF (65279). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 8.
11. Send the VETO_En0_Wr command with a data argument ‘hFCFF (64767). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 9.
12. Send the VETO_En0_Wr command with a data argument ‘hFBFF (64511). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 10.
13. Send the VETO_En0_Wr command with a data argument ‘hF7FF (63487). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 11.
14. Send the VETO_En0_Wr command with a data argument ‘hEFFF (61439). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 12.
15. Send the VETO_En0_Wr command with a data argument ‘hCFFF (53247). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 13.

16. Send the VETO_En0_Wr command with a data argument ‘hBFFF (49151). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 14.
17. Send the VETO_En0_Wr command with a data argument ‘h7FFF (32767). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on with the exception of channel 15.
18. Send the VETO_En0_Wr command with a data argument ‘hFFFF (65535). Verify the command with the VETO_En0_Rd command. Verify that all VETOs are on.
19. Send the VETO_En1_Wr command with a data argument 2. Verify the command with the VETO_En1_Rd command. Verify that all VETOs are on with the exception of channel 16.
20. Send the VETO_En1_Wr command with a data argument 1. Verify the command with the VETO_En1_Rd command. Verify that all VETOs are on with the exception of channel 17.
21. Send the GARC_Reset command to restore all register values to their initial state.

11.7 Discriminator Input Minimum Width Test

This test verifies that the VETO processing circuitry meets the specifications detailed in the ICD for the input width. For this test, all discriminator inputs greater than 100 ns in width should be processed, those in the 50 ns to 100 ns range should be sometimes processed (due to jitter), and those inputs less than 50 ns should never be processed.



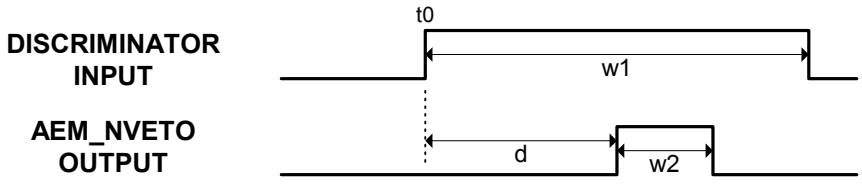
DISCRIMINATOR INPUT/OUTPUT FOR MINIMUM WIDTH TEST

1. Send the VETO_Delay_Wr command with a data argument of 0. Verify the command with the VETO_Delay_Rd command. This fixes the delay (**d** in the figure) at 150 ns for this test.
2. Send the VETO_Width_Wr command with a data argument of 0. Verify the command with the VETO_Width_Rd command. This fixes the output width (**w2** in the figure) at 50 ns for this test.
3. Set the discriminator input on channel 00 to be a pulse of greater than 300 ns in width (w1 in the figure). Verify the ACD_NVETO_00 signal out has a pulse output of 50 ns width.
4. Steadily decrease the width (w1) of the input pulse down to just greater than 100 ns in width. Verify that no pulses are missed. Adjust the width (w1) of the input pulse to between 50 ns and 100 ns in width. Verify that some pulses are there and some are missed due to the jitter in synchronization of the input signal. Adjust the input pulse width (w1) to just less than 50 ns and verify that all input pulses are rejected (i.e., no pulses output at NVETO).
5. Repeat this test for channel 01 and verify.
6. Repeat this test for channel 02 and verify.
7. Repeat this test for channel 03 and verify.
8. Repeat this test for channel 04 and verify.
9. Repeat this test for channel 05 and verify.
10. Repeat this test for channel 06 and verify.

11. Repeat this test for channel 07 and verify.
12. Repeat this test for channel 08 and verify.
13. Repeat this test for channel 09 and verify.
14. Repeat this test for channel 10 and verify.
15. Repeat this test for channel 11 and verify.
16. Repeat this test for channel 12 and verify.
17. Repeat this test for channel 13 and verify.
18. Repeat this test for channel 14 and verify.
19. Repeat this test for channel 15 and verify.
20. Repeat this test for channel 16 and verify.
21. Repeat this test for channel 17 and verify.
22. Send the GARC_Reset command to restore all register values to their initial state.

11.8 Discriminator Input Extended Width Test

This test verifies that the VETO processing circuitry meets the specifications detailed in the ICD for extended input widths. For this test, discriminator inputs greater than



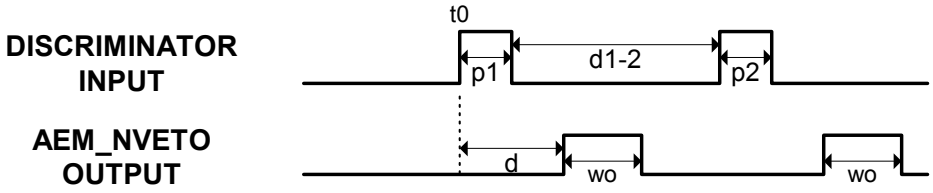
DISCRIMINATOR INPUT/OUTPUT FOR EXTENDED WIDTH TEST

1. Send the Veto_Delay_Wr command with a data argument of 0. Verify with the Veto_Delay_Rd command.
2. Send the Veto_Width_Wr command with a data argument of 1. Verify with the Veto_Width_Rd command.
3. Connect the oscilloscope to monitor the discriminator input and the AEM_NVETO_00 output as shown in the diagram above. Start with the width of the input discriminator at approximately 150 ns. Verify that the Veto_Delay is 150 ns and the Veto_Width is 100 ns.
4. Increase the discriminator input from 150 ns to 10 μs. Note that the output from the AEM_NVETO_00 signal does not change.
5. Repeat this test for channel 01 and verify.
6. Repeat this test for channel 02 and verify.
7. Repeat this test for channel 03 and verify.
8. Repeat this test for channel 04 and verify.
9. Repeat this test for channel 05 and verify.
10. Repeat this test for channel 06 and verify.

11. Repeat this test for channel 07 and verify.
12. Repeat this test for channel 08 and verify.
13. Repeat this test for channel 09 and verify.
14. Repeat this test for channel 10 and verify.
15. Repeat this test for channel 11 and verify.
16. Repeat this test for channel 12 and verify.
17. Repeat this test for channel 13 and verify.
18. Repeat this test for channel 14 and verify.
19. Repeat this test for channel 15 and verify.
20. Repeat this test for channel 16 and verify.
21. Repeat this test for channel 17 and verify.
22. Send the GARC_Reset command to restore all register values to their initial state.

11.9 Discriminator Input Double Pulse Test

This test verifies that the VETO processing circuitry meets the specifications detailed in the ICD for extended input widths. For this test, multiple discriminator inputs that cause the VETO outputs to pile-up are verified.



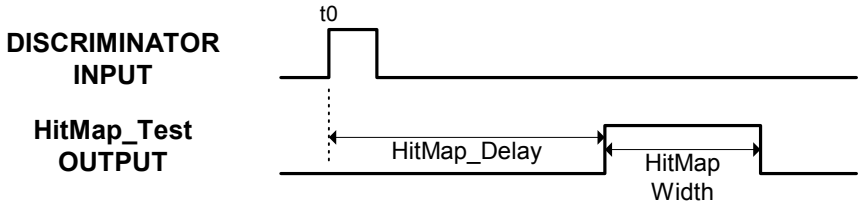
DISCRIMINATOR INPUT/OUTPUT FOR DOUBLE PULSE TEST

1. Send the Veto_Delay_Wr command with a data argument of 0. Verify with the Veto_Delay_Rd command.
2. Send the Veto_Width_Wr command with a data argument of 3. Verify with the Veto_Width_Rd command.
3. Using the test pulser in the double pulse mode, start with the width of the input pulses at 150 ns and the distance, d1-2, at 1 μsec. With two pulses input, there should be two pulses output, each 200 ns in width (**wo**) at the AEM_NVETO_00 output.
4. Move the input pulses closer until the two AEM_NVETO_00 pulses touch. At this point, they should become one extended pulse. As the input pulses move closer, the tail of the extended pulse should become shorter. As the input pulses are separated again by greater than 200 ns, the AEM_NVETO_00 should again show two pulses.
5. Repeat this test for channel 01 and verify.
6. Repeat this test for channel 02 and verify.
7. Repeat this test for channel 03 and verify.
8. Repeat this test for channel 04 and verify.

9. Repeat this test for channel 05 and verify.
10. Repeat this test for channel 06 and verify.
11. Repeat this test for channel 07 and verify.
12. Repeat this test for channel 08 and verify.
13. Repeat this test for channel 09 and verify.
14. Repeat this test for channel 10 and verify.
15. Repeat this test for channel 11 and verify.
16. Repeat this test for channel 12 and verify.
17. Repeat this test for channel 13 and verify.
18. Repeat this test for channel 14 and verify.
19. Repeat this test for channel 15 and verify.
20. Repeat this test for channel 16 and verify.
21. Repeat this test for channel 17 and verify.
22. Send the GARC_Reset command to restore all register values to their initial state.

11.10 Test of the HitMap Functionality

This section tests the proper functioning of the commandable functions of the HitMap test point. This test requires the use of an oscilloscope, a test pulser for input stimulus, and test points to monitor the AEM_VETOs. The test setup is identical to the VETO test setup shown above. Only channel 00 is available on the test pin for HitMap verification. The diagram below details the measurement points to use with the digitizing oscilloscope.



HITMAP DELAY & WIDTH MEASUREMENT

11.11 HitMap Width Test

1. Send the GARC_Mode_Wr command with a data argument of decimal 768 to ensure the test pin multiplexer is set to the HitMap_Test output. Place an oscilloscope probe on this test point.
2. Send the HitMap_Width_Wr command with a data argument of 0. Verify the command using the HitMap_Width_Rd readback.
3. Send the HitMap_Delay_Wr command with a data argument of 0. Verify the command using the HitMap_Delay_Rd readback.
4. Send the HitMap_Deadtime_Wr command with a data argument of 0. Verify the command using the HitMap_Deadtime_Rd readback.

5. Set up the test pulser to stimulate the discriminator input for channel 0. Monitor the HitMap Test output with the oscilloscope triggered from the test pulser. Using the HitMap_Width_Wr command with data arguments of 0 – 15, verify the HitMap Width varies as expected and record the data in the table below.

HitMap Width Command	Measured Width (ns)	Expected Width (ns)
0		150
1		200
2		250
3		300
4		350
5		400
6		450
7		500
8		550
9		600
10		650
11		700
12		750
13		800
14		850
15		900

6. Send the HitMap_Width_Wr command with a data argument of 0 to reset the HitMap Width. Verify the command with the HitMap_Width_Rd command.

11.12 HitMap Delay Test

1. Using the HitMap_Delay_Wr command with data arguments of 0 – 31, verify the HitMap Delay varies as expected and record the data in the table below.

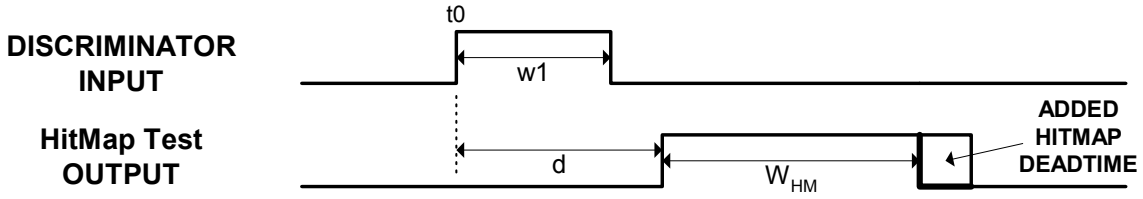
HitMap Delay Command	Measured Delay (ns)	Expected Delay (ns)
0		850
1		900
2		950
3		1000
4		1050
5		1100
6		1150
7		1200
8		1250
9		1300
10		1350
11		1400
12		1450
13		1500
14		1550
15		1600
16		1650

17		1700
18		1750
19		1800
20		1850
21		1900
22		1950
23		2000
24		2050
25		2100
26		2150
27		2200
28		2250
29		2300
30		2350
31		2400

2. Send the HitMap_Delay_Wr command with a data argument of 0 to reset the HitMap Delay. Verify the command with the HitMap_Delay_Rd command.

11.13 HitMap Deadtime Stretch Test

This test will monitor the function of the HitMap_Deadtime adjustment. This Deadtime register adds a time (0 to 350 ns) to the end of the HitMap pulse, starting at the end of the HitMap Width calculation. The diagram below illustrates the position of the added time window.



HITMAP DEADTIME TEST

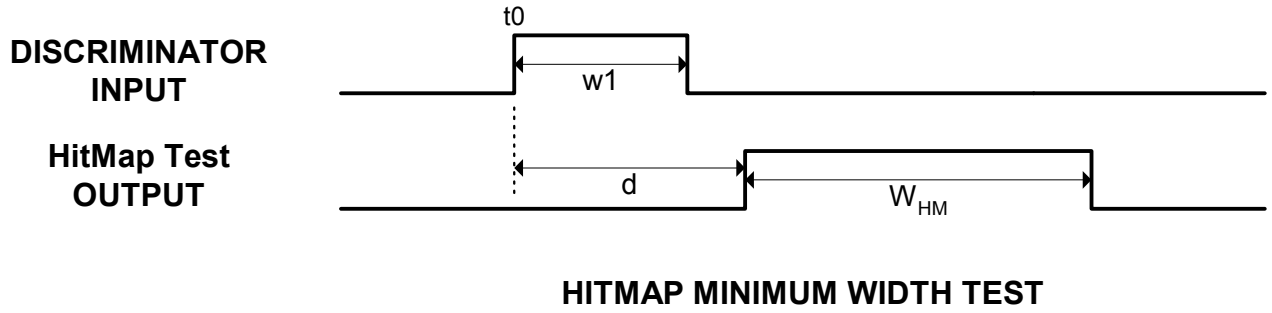
1. Using the HitMap_Deadtime_Wr command with data arguments of 0 – 7, verify the HitMap Deadtime extends the trailing edge of the HitMap test pulse as expected and record the data in the table below.

HitMap Deadtime Command	Measured Deadtime Pulse Extension - (ns)	Expected Deadtime Pulse Extension -(ns)
0		0
1		50
2		100
3		150
4		200
5		250
6		300
7		350

2. Send the GARC_Reset command to restore all register values to their initial state.

11.14 HitMap Minimum Width Test

This test characterizes the performance of the HitMap functionality for a minimum width discriminator input.

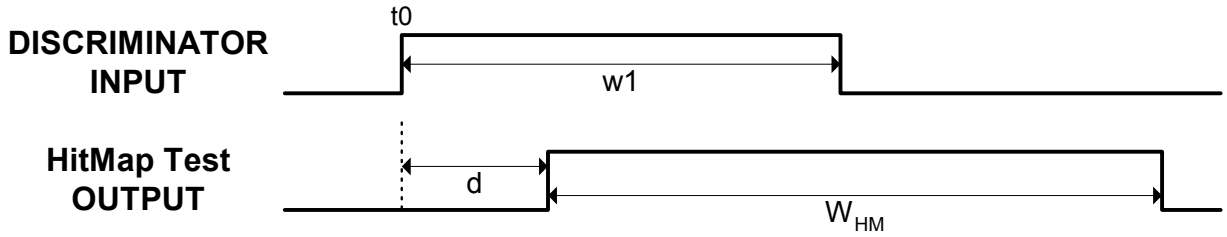


HITMAP MINIMUM WIDTH TEST

1. Send the HitMap_Delay_Wr command with a data argument of 0. Verify the command with the HitMap_Delay_Rd command. This fixes the delay (**d** in the figure) at 850 ns for this test.
2. Send the HitMap_Width_Wr command with a data argument of 0. Verify the command with the HitMap_Width_Rd command. This fixes the output width ($\mathbf{W_{HM}}$ in the figure) at 150 ns for this test.
3. Starting with the discriminator input width at 200 ns, note that the HitMap output is shown as above. Decrease the width of the discriminator input until the HitMap output starts to disappear. Verify that the HitMap_Test is stable for all discriminator inputs greater than 50 ns.

11.15 HitMap Extended Pulse Test

This test characterizes the performance of the HitMap circuitry with an extended discriminator input pulse.

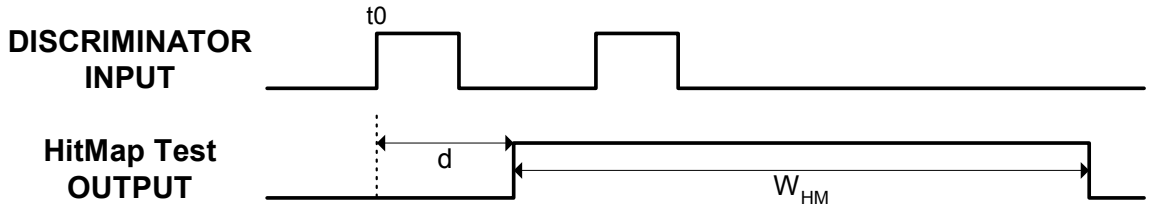


HITMAP EXTENDED PULSE TEST

1. Send the HitMap_Delay_Wr command with a data argument of 0. Verify the command with the HitMap_Delay_Rd command. This fixes the delay (**d** in the figure) at 850 ns for this test.
2. Send the HitMap_Width_Wr command with a data argument of 0. Verify the command with the HitMap_Width_Rd command. This fixes the output width ($\mathbf{W_{HM}}$ in the figure) at 150 ns for this test.
3. Starting with the discriminator input width at about 500 ns, increase the discriminator width up to approximately 10 μsec. Verify that the HitMap_Width increases with the discriminator input width.

11.16 HitMap Double Pulse Test

This test characterizes the performance of the HitMap circuitry with a double pulse at the discriminator input.



HITMAP DOUBLE PULSE TEST

1. Send the HitMap_Delay_Wr command with a data argument of 0. Verify the command with the HitMap_Delay_Rd command. This fixes the delay (**d** in the figure) at 850 ns for this test.
2. Send the HitMap_Width_Wr command with a data argument of 0. Verify the command with the HitMap_Width_Rd command. This fixes the output width ($\mathbf{W_{HM}}$ in the figure) at 150 ns for this test.
3. Send the HitMap_Deadtime_Wr command with a data argument of 0. Verify the command with the HitMap_Deadtime_Rd command.
4. Starting with the discriminator input pulses at about 150 ns in width and 10 μs apart, note that there are two HitMap output pulses each 850 ns wide.
5. Decrease the distance between the discriminator input pulses until the two HitMap output pulses merge to one extended pulse. Verify that this merge time is approximately 150+850 = 1000 ns. Verify that the length of the pulse is extended 850 ns from the leading edge of the second pulse.

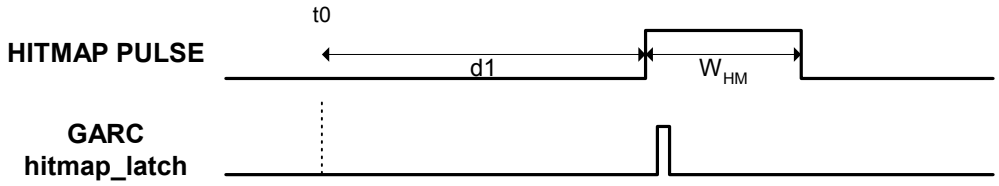
11.17 Synchronization of the HitMap Latch to the Discriminator Input Pulse

This section verifies the proper functionality of the HitMap latch function. The HitMap is latched within 50 ns of the moment that a trigger command is detected. The HitMap Delay must be adjusted such that the delayed Veto pulse coincides with the time from the incident particle (i.e., t0) to the receipt of the trigger.

The HitMap pulse width is adjustable to allow for jitter in the LAT global triggering algorithm. The HitMap "deadtime" is also adjustable; this parameter allows for the GAFE analog baseline sufficient time to return to zero following large pulses.

This test will check for the minimum delay and maximum delay for which the HitMap bits are set active high. Using the discriminator input setup detailed above (all 18 discriminator inputs driven by the pulser), perform the following test.

1. Adjust the HitMap Width to be 150ns by sending the HitMap_Width_Wr command with a data argument of 0. Verify this command with the HitMap_Width_Rd command.
2. Set the HitMap deadtime to be 0 with the HitMap_Deadtime_Wr command with a data argument of 0. Verify this command with the HitMap_Deadtime_Rd command.
3. Start with the HitMap_Delay at a value of 2400 ns by sending the HitMap_Delay_Wr command with a data argument of 31. Verify this command with the HitMap_Delay_Rd command.
4. Send the Hold_Delay_Wr command with a data argument value of 25. Verify with the Hold_Delay_Rd command. This sets the Trigger-to-Hold time to be 1500 ns.
5. Send the Trigger_NOZS command and note the contents of the HitMap. Decrement the value of the HitMap delay and verify, again using the HitMap_Delay_Wr and HitMap_Delay_Rd commands. Repeat until the HitMap bits in the event data word change from 0 (inactive) to 1 (active).

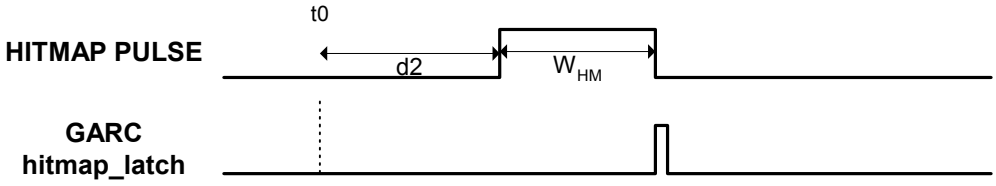


HITMAP LATCH ON LEADING EDGE

In the diagram above, this is equivalent to decreasing delay **d1** until the leading edge of the HitMap pulse is captured by the internal GARC latch signal, which is fixed by the Hold_Delay. Record this delay setting below (the expected setting is 13 = 1500 ns)

HitMap Delay (d1, leading edge of HitMap):

6. Send the Trigger_NOZS command and note the contents of the HitMap. Decrement the value of the HitMap delay and verify, again using the HitMap_Delay_Wr and HitMap_Delay_Rd commands. Repeat until the HitMap bits in the event data word change from 1 (active) to 0 (inactive).



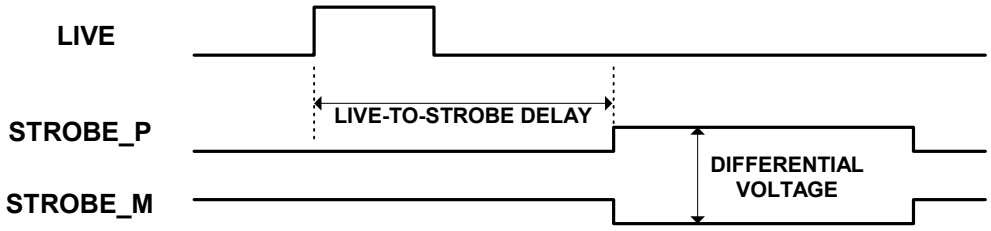
HITMAP LATCH ON TRAILING EDGE

In the diagram above, this is equivalent to decreasing delay **d2** until the trailing edge of the HitMap pulse is just past the internal GARC latch signal, which is fixed by the Hold_Delay. Record this delay setting below (the expected setting is 17 = 1700 ns). Record this delay setting below.

HitMap Delay (d2, trailing edge of HitMap):

11.18 Strobe Test

This section tests the proper functioning of the commandable strobe pulse to the GAFEs from the GARC. This signal is used as a level command to the GAFE(s) to indicate the timing of the test charge injection to the amplifier front end. Using the oscilloscope monitoring Live on GARC pin 179, GAFE_STROBEP on GARC pin 163 and GAFE_STROBEM on GARC pin 164, perform the following (as shown in the diagram below):



STROBE COMMAND TEST

1. Send the GARC_Mode_Wr command with data argument 1792 to switch the test pin multiplexer to the Live signal.

2. Set the scope to acquire on the posedge of Live. Send the GARC_Cal_Strobe command (this is a dataless command so the data argument is 0).
3. Verify the STROBE command has executed by noting the differential signal on the two oscilloscope channels. It is expected that the voltage amplitude of each side of the differential is greater than 50 mV and that the pulse duration is approximately 10 μsec.

STROBEP to STROBEM differential voltage:

Live-to-STROBE delay:

STROBE pulse duration (μsec):

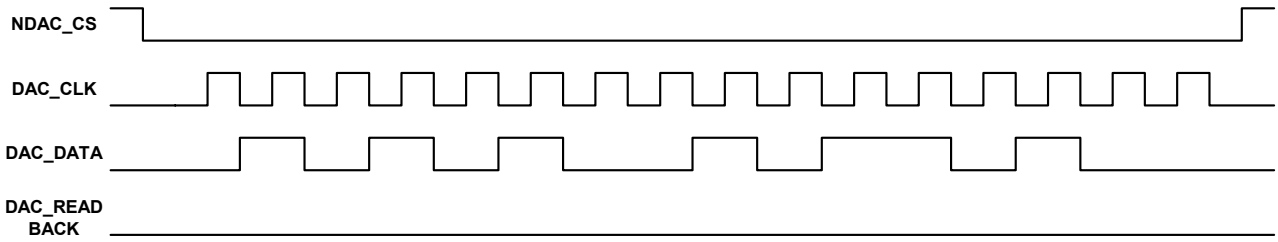
4. Send the GARC_Reset command to restore all register values to their initial state.

11.19 Test of the MAX5121 DAC and Associated Control Functions

This section tests the proper commanding of the MAX5121 DAC. A voltmeter will be used to monitor the DAC output on the board under test.

11.20 Capture of the DAC Control Signals

1. Connect the probe on channel 1 to NDAC_CS. Set the scope to trigger on the negedge of this signal.
2. Connect the probe on channel 2 to DAC_CLK.
3. Connect the probe on channel 3 to DAC_DATA.
4. Connect the probe on channel 4 to DAC_READBACK.
5. Send the HVBS_Level_Wr command with a data argument of hex A5A (decimal 2650). Send the HVBS_Level_Rd command to verify.
6. Check that the scope is armed and then send the Use_HV_Nominal_Wr command. The scope should trigger and the following waveform should be displayed. Verify that the DAC clock is nominally 5 MHz.
7.



DAC DATA WRITE WITH DATA PATTERN HEX A5A

8. Check that the scope is armed and then send the Use_HV_Nominal_Rd command. The software readback should capture the following value in the 16 bit return word: hex F4B4 (decimal 31348). This value may also be seen on the DAC DATA READBACK scope trace.

9. Move the probe on channel 1 to NDAC_CLR. Verify the scope is armed and send the GARC_Reset command. The scope should trigger and a valid active low clear signal should be observed on the scope.

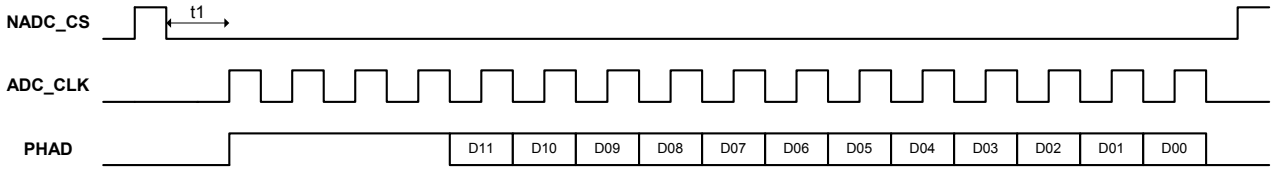
11.21 Test of the GARC DAC Interface Circuitry

This section tests the proper functioning of the circuitry that commands the MAX5121 DAC.

1. Send the HVBS_Level_Wr command with a data argument of decimal 2048. Send the HVBS_Level_Rd command to verify the write command.
2. Send the SAA_Level_Wr command with a data argument of decimal 1024. Send the SAA_Level_Rd command to verify.
3. Using a multimeter on the DAC output, verify that the MAX5121 output is at 0V.
4. Send the Use_HV_Nominal command. Verify that the MAX5121 output goes to half scale, e.g., 0.625 V.
5. Send the DAC_HVReg_Rd command and verify the return data pattern from the MAX5121 DAC is decimal 59392.
6. Send the Use_SAA_Level command. Verify that the MAX5121 output goes to 1/4 scale, e.g., 0.312 V.
7. Send the DAC_SAAReg_Rd command and verify the return data pattern from the MAX5121 DAC is decimal 58368.
8. Send the GARC_Reset command. Verify that the MAX5121 output goes to 0 V.

11.22 Capture of the ADC Control Signals

1. Connect the probe on channel 1 to NADC_CS. Set the scope to trigger on the negedge of this signal.
2. Connect the probe on channel 2 to ADC_CLK.
3. Connect the probe on channel 3 to PHAD00.
4. Send the TRIG_ZS command to initiate an analog-to-digital conversion. The scope should trigger and the following waveform should be displayed. Verify that the ADC clock is nominally 2 MHz. The time t1 represents the total ADC TACQ (ADC acquisition time).

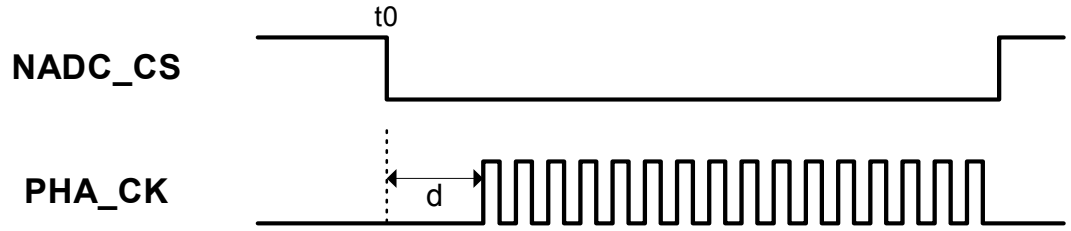


MAX145 ADC CONVERSION CYCLE

11.23 ADC TACQ Test

1. The next parameter to be measured will be the ADC time to acquisition (essentially the settling time allowed for the analog signal). This is a command with a 6 bit data argument. This alters the time between the ADC chip select transitioning active low and the start of the ADC clock. The register is set with the ADC_TACQ_Wr command and verified with the ADC_TACQ_Rd command. For each value of ADC_TACQ in the table below, record the time from the falling edge of the ADC_CS_N

and the rising edge of the first ADC clock. Nominally, with an ADC_TACQ register setting of 0, there will be a CS → PHA clock delay time of 2500 ns. There is the potential for a small amount of synchronization jitter (~few clocks) within the state machine on this parameter.



ADC_TACQ DELAY MEASUREMENT

For each ACD_TACQ step, measure the delay, **d**, as shown in the diagram above. For the table below, the important parameter is the even 50 ns spacing on each step from the original zero point.

ADC TACQ Register	Measured Time CS → PHA Clock (ns)	Expected Time CS → PHA Clock (ns)
0		2500
1		2550
2		2600
3		2650
4		2700
5		2750
6		2800
7		2850
8		2900
9		2950
10		3000
11		3050
12		3100
13		3150
14		3200
15		3250
16		3300
17		3350
18		3400
19		3450
20		3500
21		3550
22		3600
23		3650
24		3700
25		3750
26		3800
27		3850
28		3900
29		3950
30		4000
31		4050
32		4100
33		4150
34		4200

35		4250
36		4300
37		4350
38		4400
39		4450
40		4500
41		4550
42		4600
43		4650
44		4700
45		4750
46		4800
47		4850
48		4900
49		4950
50		5000
51		5050
52		5100
53		5150
54		5200
55		5250
56		5300
57		5350
58		5400
59		5450
60		5500
61		5550
62		5600
63		5650

2. Send the GARC_Reset command to restore all register values to their initial state.

11.24 Test of the GARC LVDS Circuitry Driver Currents

This section tests the proper functioning of the GARC LVDS Driver circuitry. It is required that all LVDS drivers be terminated at the receiver with a 100 ohm resistor. The ACD-to-AEM nominal drive current is 3.5 mA across the 100 ohms for a voltage differential of 350 mV.

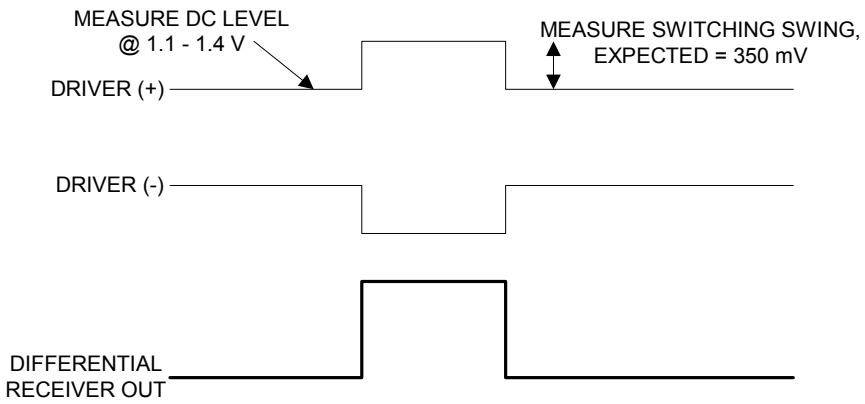
There are 6 LVDS receiver input pairs to the GARC:

a) ACD_CLK_A, ACD_CLK_B
b) ACD_CMD_A, ACD_CMD_B
c) ACD_RESET_A, ACD_RESET_B

There are 40 LVDS driver output pairs from the GARC:

d) ACD_NSDATA_A, ACD_NSDATA_B
e) ACD_NVETO_nnA, ACD_NVETO_nnB (where nn is 00 – 17)
f) ACD_CNO_A, ACD_CNO_B

The "A" side interface will be examined first. Verify that an AEM (or AEM simulator) is connected to the GARC "A" side interface. Using a digitizing oscilloscope, measure the DC baseline and switching differential voltage on each of the driver/receiver pairs. Two scope probes may be used, one on each side of the differential receiver inputs. The DC level and switching voltages may be measured as diagrammed below.



Record the information for each pin on the Primary side interface in the table below.

GARC Signal	GARC Pin (+)	GARC Pin (-)	DC Level (V)	Switch (+)	Switch (-)
ACD_CLK_A	2	3			
ACD_CMD_A	6	7			
ACD_RESET_A	10	11			
ACD_NSDATA_A	14	15			
ACD_NVETO_00A	22	23			
ACD_NVETO_01A	28	29			
ACD_NVETO_02A	32	33			
ACD_NVETO_03A	36	37			
ACD_NVETO_04A	40	41			
ACD_NVETO_05A	44	45			
ACD_NVETO_06A	48	49			
ACD_NVETO_07A	54	55			
ACD_NVETO_08A	58	59			
ACD_NVETO_09A	62	63			
ACD_NVETO_10A	66	67			
ACD_NVETO_11A	70	71			
ACD_NVETO_12A	74	75			
ACD_NVETO_13A	204	205			
ACD_NVETO_14A	200	201			
ACD_NVETO_15A	196	197			
ACD_NVETO_16A	192	193			
ACD_NVETO_17A	188	189			
ACD_CNO_A	18	19			

Switching control of the AEM interface to the Secondary side interface, record the information below.

GARC Signal	GARC Pin (+)	GARC Pin (-)	DC Level (V)	Switch (+)	Switch (-)
ACD_CLK_B	4	5			
ACD_CMD_B	8	9			
ACD_RESET_B	12	13			
ACD_NSDATA_B	16	17			
ACD_NVETO_00B	25	26			
ACD_NVETO_01B	30	31			
ACD_NVETO_02B	34	35			
ACD_NVETO_03B	38	39			
ACD_NVETO_04B	42	43			
ACD_NVETO_05B	46	47			

ACD_NVETO_06B	50	51			
ACD_NVETO_07B	56	57			
ACD_NVETO_08B	60	61			
ACD_NVETO_09B	64	65			
ACD_NVETO_10B	68	69			
ACD_NVETO_11B	72	73			
ACD_NVETO_12B	76	77			
ACD_NVETO_13B	206	207			
ACD_NVETO_14B	202	203			
ACD_NVETO_15B	198	199			
ACD_NVETO_16B	194	195			
ACD_NVETO_17B	190	191			
ACD_CNO_B	20	21			

This completes the testing of the GARC LVDS driver circuitry.

12.0 FREE Circuit Assembly Tests with Laboratory Pulse Generators

The following tests utilized NIM-mounted laboratory pulse generators to simulate a phototube input at the GAFE inputs. These tests will characterize the GAFE ASICs and associated readout circuitry. The VETO enable/disable function will be tested.

12.1 Test of the GAFE Low-Energy Channels

This test will verify the operation of the low-energy channel for each GAFE ASIC on the FREE printed circuit card. Included in this test will be the VETO output and the ADC conversion values as measured at the GARC. The low-energy threshold will be characterized during this test. The HLD enable/disable function will be tested. Autoranging switching of the shaped channel will be tested.

<< MORE HERE >>

12.2 Test of the GAFE High-Energy Channels

This test will verify the operation of the high-energy channel for each GAFE ASIC on the FREE printed circuit card. Included in this test will be the CNO output and the ADC conversion values as measured at the GARC. The high-energy threshold will be characterized during this test.

<< MORE HERE >>

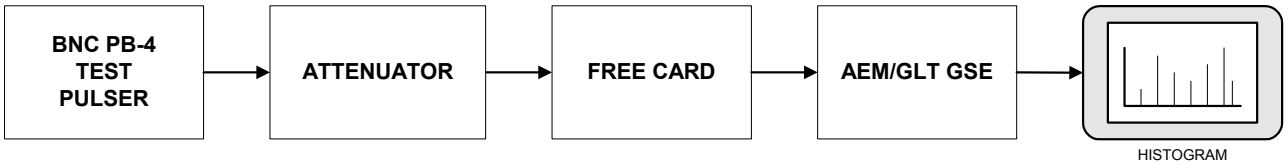
12.3 Characterization of the GAFE Test Charge Injection (TCI) Circuitry

This test will characterize the operation of the GAFE test charge injection circuitry.

<< MORE HERE >>

12.4 Integral Non-Linearity Test

Integral Non-Linearity Test: One possible setup is shown below.



INL TEST - REPEAT FOR EACH OF 18 CHANNELS

The following list is standardized to provide a meaningful set of pulse inputs to be used to test the FREE board. Using a calibrated charge terminator (set to 64 pC == 100 MIP), the test pulser amplitude should be adjusted for the following outputs. Peak centroid and FWHM width would be recorded from the histogram accumulated on the GSE. This test would be run at a nominal rate of 1 kHz (i.e., 1 msec pulse spacing).

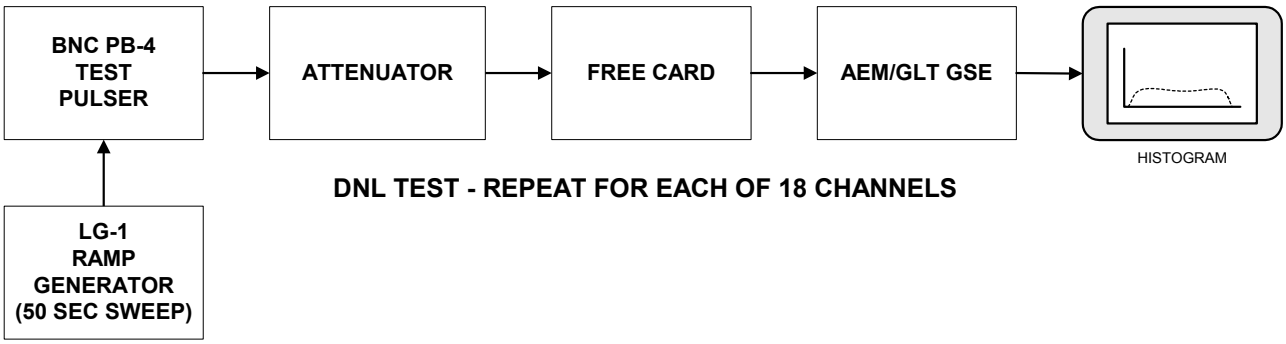
Charge Input (pC)	MIP Equivalent	Attenuator
0.064	0.1	60 dB
0.128	0.2	60 dB
0.192	0.3	60 dB
0.256	0.4	60 dB
0.32	0.5	60 dB
0.384	0.6	60 dB
0.448	0.7	60 dB
0.512	0.8	60 dB
0.576	0.9	60 dB
0.64	1	40 dB
1.28	2	40 dB
1.92	3	40 dB
2.56	4	40 dB
3.20	5	40 dB
3.84	6	40 dB
4.48	7	40 dB
5.12	8	40 dB
5.76	9	40 dB
6.4	10	20 dB
12.8	20	20 dB
19.2	30	20 dB
25.6	40	20 dB
32.0	50	20 dB
38.4	60	20 dB
44.8	70	20 dB
51.2	80	20 dB
57.6	90	20 dB
64.0	100	0 dB

128	200	0 dB
192	300	0 dB
256	400	0 dB
320	500	0 dB
384	600	0 dB
448	700	0 dB
512	800	0 dB
576	900	0 dB
640	1000	0 dB

12.5 Differential Non-Linearity Test – Low Energy Range

This test will monitor the DNL of each GAFE and ADC pair over the low-energy range.

Differential Non-Linearity Test: One possible setup is shown below.



The differential non-linearity test is performed similarly to the above, but instead of discrete energy inputs, the input pulse is swept through the entire range and a statistical analysis on ADC bin width is performed. For this test, we could use two ranges: 0 – 10 MIP equivalents and 0 – 1000 MIP equivalents.

<< MORE HERE >>

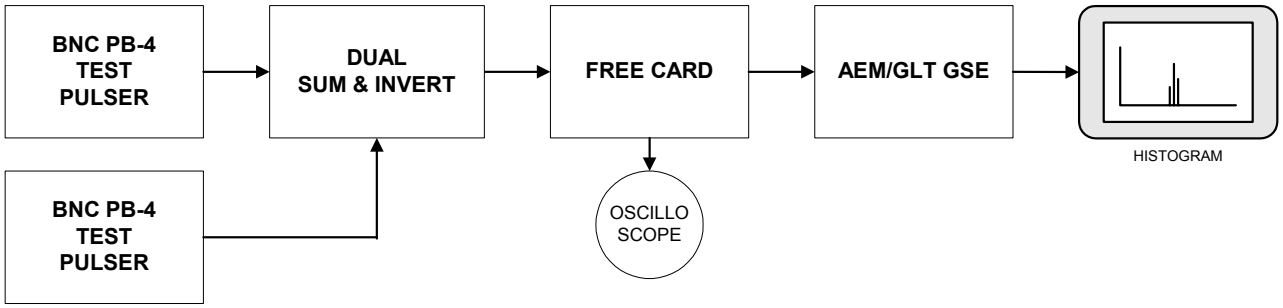
12.6 Differential Non-Linearity Test – High Energy Range

The setup is similar to that of the previous section. This test will monitor the DNL of each GAFE and ADC pair over the high-energy range.

<< MORE HERE >>

12.7 Pulse Pile-Up Test

Pulse Pile-Up Test: One possible setup is shown below.



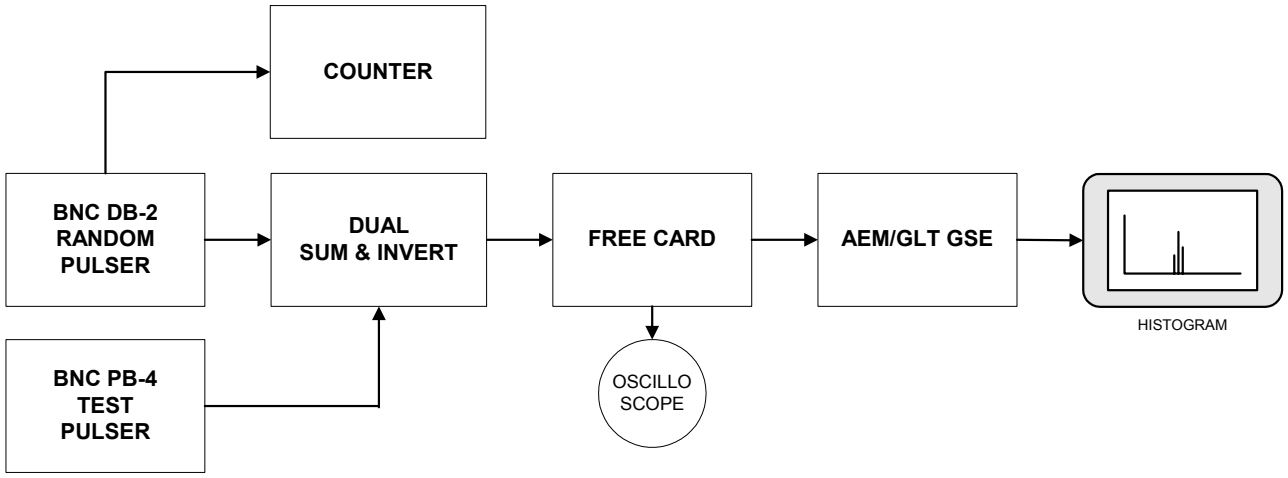
PULSE PILE UP TEST - ONE CHANNEL

This test should verify that effects on PHA and VETO threshold based on the proximity of one pulse to another. Specifically, the effect on small pulses following very large events should be explored. The two test pulsers have independent amplitudes and delays, providing the opportunity to characterize amplitude vs. timing variations.

<< MORE HERE >>

12.8 Random Pulser Test

Random Pulse testing at various rates: One possible setup is shown below.



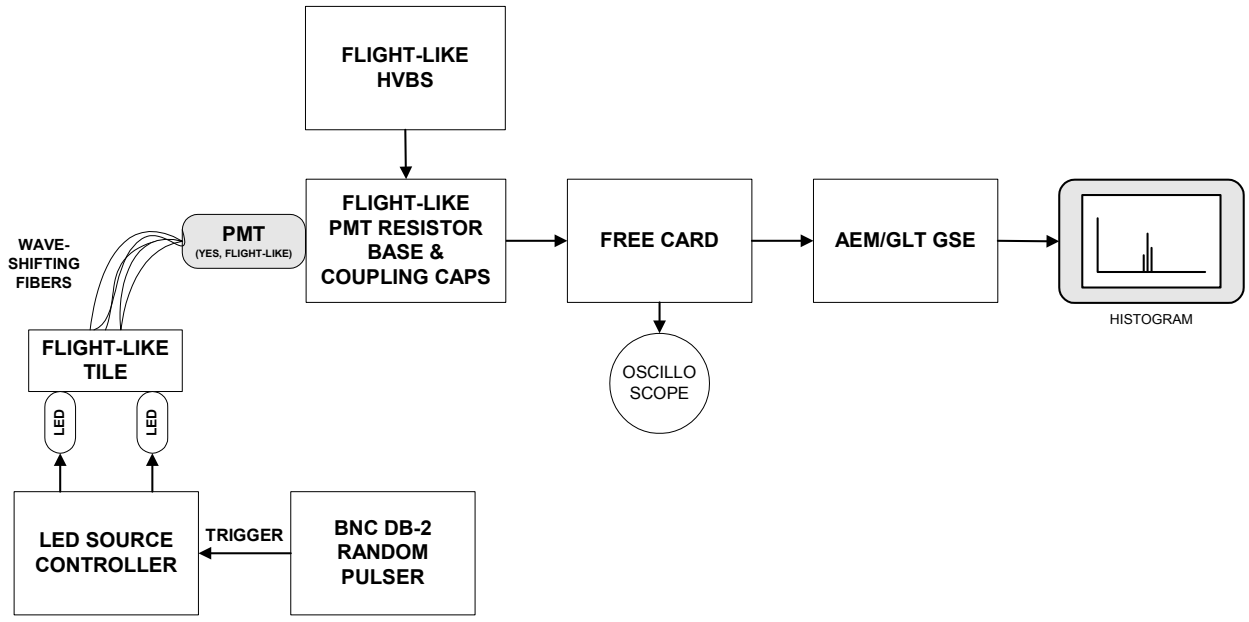
RANDOM RATE TEST

In this test we might set the PB-4 pulse to a 10 Hz, 1000 MIP-equivalent pulse to simulate CNO events. The random DB-2 pulser (this is like a selective white noise trigger) could be set to test at rates 100 Hz, 1000 Hz, 3000 Hz, and 10000 Hz with a 0.2 – 1 MIP equivalent pulse amplitude. During this test, both peak widths and VETO firings will be monitors. The counter from the random pulser can assist in comparing pulser triggers versus VETO outputs.

<< MORE HERE >>

13.0 FREE Circuit Card Assembly End-to-End Test with Phototubes

End-to-End Flight-Like Test: Once the electronics has been characterized with the above tests, we will be able to determine the effects of adding the tube, biasing base, and HVBS supply. One possible setup is shown below.



END-TO-END ACD FREE TEST

This setup will require the construction of the LED source controller with the collimated LEDs. One LED (or LED group) being used to simulate CNOs (we set this to be some fixed amplitude representing some random number like 26*26 MIP-equivalents and some fixed rate like 10 Hz). We set the other amplitude to be approximately 1 MIP-equivalent and then use the DB-2 pulser for random and/or fixed period triggers. This would enable us to generate light in a tile that approximates both particles and heavy nuclei at the same time.

The goal of the end-to-end test would be to characterize the performance of the ACD detector and electronics system in an environment as close to the flight environment as we could simulate on the lab bench.

<< MORE HERE >>

14.0 FREE Circuit Card Assembly System Noise Measurement

Once the system has been completely integrated, an end-to-end system noise test shall be performed. In this test, we desire to understand and measure noise contributions from each component in the electronics chain. Noise sources that may be identified are shown below.

<< MORE HERE >>

Appendix 1: GARC Documentation

A more complete set of documentation (such as the Verilog, EDIF, layout, and wirebonding diagram) is available on the LHEA ACD electronics web page at:

http://lhea-glast.gsfc.nasa.gov/acd/electronics/

Appendix 2: FREE Circuit Card Test Points

The FREE circuit card assembly has the following test point locations available for use during circuit-card level testing.

Schematic Page	FREE Card Test Point	Location	Near Part, Pin	Expected Value	Notes
3	Op Amp Ground	TPDAC4	UDAC1-8	~ 200 mV	Limited due to (-) rail
3	REFTST	TPDAC3	UDAC1-1	2.5	1.25V ref x 2
3	DAC Output	TPDAC2	UDAC3-2	0 – 1.249 V	Commandable DAC Output
3	DAC reference	TPREF1	UDAC3-14	1.25 V	Internal DAC Reference
3	DAC Receiver Test	JDAC1	UDAC1-7	HVP - HVN	Differential Receiver as in HVBS
3	Differential Driver Offset	JDAC2	UDAC2-7	~ 1.40V	HVP, HVN offset for DAC differential drivers
6	+3.3 V before CM choke	LCM1-1	LCM1-1	+3.3 V	FREE board voltage prior to filtering
6	+3.3 V digital	CP2-1	CP3-1	+3.3 V	FREE board voltage VDD after CM choke
6	+3.3 V analog	CP7-1	CP8-1	+3.3 V	FREE board voltage VCC after filtering
7	LVDS_PRESET_ADJ	RG10-1	GARC-184		
7	BIAS_DRV_H	RG3-1	GARC-160		
7	BIAS_DRV_L	RG1-1	GARC-169		
7	BIAS_RCVR	RG6-1	GARC-156		
7	HLD_WOR_BIAS	RG8-1	GARC-104		
7	HITMAP_TEST	TG179	GARC-179		GARC Trigger Test Point
7	TRIG_TEST	TG180	GARC-180		GARC HitMap Test Point
7	VETO_ENA_TEST	TG183	GARC-183		GARC Veto Enable A Status Test Point
7	GARC Core Voltage	CG9-1	CG8-1	+3.2 V	GARC VDD
9	FREE ID, Bit 0	RIL0-1	UID1-11	0 or +3.3 V	FREE Board ID
9	FREE ID, Bit 1	RIL1-1	UID1-12	0 or +3.3 V	FREE Board ID
9	FREE ID, Bit 2	RIL2-1	UID1-13	0 or +3.3 V	FREE Board ID
9	FREE ID, Bit 3	RIL3-1	UID1-14	0 or +3.3 V	FREE Board ID
9	FREE ID, Bit 4	RIL4-1	UID1-3	0 or +3.3 V	FREE Board ID
9	FREE ID, Bit 5	RIL5-1	UID1-4	0 or +3.3 V	FREE Board ID
9	FREE ID, Bit 6	RIL6-1	UID1-5	0 or +3.3 V	FREE Board ID
9	FREE ID, Bit 7	RIL7-1	UID1-6	0 or +3.3 V	FREE Board ID
8	Mux Hold, Ch. 00	JH0	GAFE00-31		Analog held signal, GAFE Channel 0
8	Mux Shaper, Ch. 00	JSA0	GAFE00-		Analog shaping amplifier

			33		signal, GAFE Channel 0
8	Analog VCC	TPVCC1			Analog +3.3V
8	GAFE HOLD	JHLD0	RT3-1		Hold command fromGARC
10	PWR_ON_RST	TPR1	UR1-2	0V after power up	Power on Reset
12	PMT Input, Ch 00	TP151	J00		PMT Input
12	PMT Input, Ch 01	TP	J01		PMT Input
12	PMT Input, Ch 02	TP	J02		PMT Input
12	PMT Input, Ch 03	TP	J03		PMT Input
12	PMT Input, Ch 04	TP	J04		PMT Input
12	PMT Input, Ch 05	TP	J05		PMT Input
12	PMT Input, Ch 06	TP	J06		PMT Input
12	PMT Input, Ch 07	TP	J07		PMT Input
12	PMT Input, Ch 08	TP	J08		PMT Input
12	PMT Input, Ch 09	TP	J09		PMT Input
12	PMT Input, Ch 10	TP	J10		PMT Input
12	PMT Input, Ch 11	TP	J11		PMT Input
12	PMT Input, Ch 12	TP	J12		PMT Input
12	PMT Input, Ch 13	TP	J13		PMT Input
12	PMT Input, Ch 14	TP	J14		PMT Input
12	PMT Input, Ch 15	TP	J15		PMT Input
12	PMT Input, Ch 16	TP	J16		PMT Input
12	PMT Input, Ch 17	TP	J17		PMT Input
12	ADC VREF, Ch. 00	CA4-1	UADC00-5		ADC Voltage Reference
12	GAFE VREF, Ch. 00	C266-1	GAFE00-8		GAFE Internal Voltage Reference
12	GAFE VDD, Ch. 00	CA6-1	CA5-1		GAFE Digital Power
12	GAFE VCC, Ch. 00	CA7-1	CA8-1		GAFE Analog Power

Appendix 3: GARC Configuration Command Format

The GARC logic core responds only to properly structured commands. There are two possible command types – Trigger Commands and Configuration Commands. Trigger commands initiate an Event Data cycle, causing a GAFE Hold, an analog-to-digital conversion, and the return of an event data packet. Configuration commands are used to either read or write GARC or GAFE registers.

The format for Trigger Commands is detailed in the table below.

Bit(s)	Bit Description	Value
3	Start Bit	1
2:1	Trigger Type Bits	10 for ZS Enabled Trigger 01 for Send All PHA Trigger
0	Parity Bit	Odd parity bit over previous two bits

Therefore, a 1100 is a ZS Enabled Trigger command and a 1010 is a Send All PHA Trigger command.

This format for GARC Configuration Commands is detailed in the table below.

Field	# bits	Function
Start	1	1 for start
CMD Type	2	00 for command
CMD Type Parity	1	Odd Parity over previous 2 bits (without Start bit)
GAFE/GARC Select	1	0 for GARC 1 for GAFE
GAFE/GARC Address	5	GAFE: Select which GAFE, 0x1F for all GAFE GARC: Select which function block
Read/Write	1	0 for write, 1 for read
Data/Dataless	1	0 for dataless, 1 for data, always 1 for ACD
register/function number	4	Which register/function in the function block
CMD Parity	1	Odd parity bit over previous 15 bits
Data	16	Data Field
Data Parity	1	Odd parity bit over previous 16 bits

Appendix 4: GARC Event Data Format

The GARC core returns an event data packet when a valid trigger command is received. The event data format is detailed in the table below.

Field	# bits	Function
Start Bit	1	1 for start
Hit Map Bits	18	Bits 17-0 for channels 0-17, bit set if hit in channel
Zero Suppression Bits	18	Bits 17-0 for channels 0-17, bit set if PHA above threshold
CMD/Data ERROR	1	Error in command parity detected
Header Parity	1	Odd parity bit over previous 37 bits
PHA Words (quantity 0-18) Order: Channel 0 to channel 17	15	Bit 14: 1 if another PHA word follows this one, 0 if this is last one Bit 13: 1 for high range, 0 for low range Bits 12-1: the PHA value, 0 to 4095 Bit 0: Odd parity over last 14 bits

Appendix 5: GARC Configuration Data Readback Format

The GARC core returns an register configuration data packet when a valid configuration readback command is received. The configuration readback data format is detailed in the table below.

Field	# bits	Function
Start	1	1 for start
GAFE/GARC Select	1	Copy of write command field (0 for GARC, 1 for GAFE)
GAFE/GARC Address	5	Copy of write command field (Select which GAFE)
Read/Write	1	1 for read
Data/Dataless	1	always 1
Register/function number	4	Copy of write command field (which register/function in the function block)
CMD Parity	1	Odd parity bit over previous 12 bits
Data	16	Data, MSB first
CMD/DATA ERROR	1	Error in parity detected
Parity	1	Odd parity bit over previous 17 bits

Appendix 6: MDM Connector Pin List

Connector JHV1 to High Voltage Bias Supply #1

Connector Pin	Signal Name
JHV1 - 1	HVBS +28V
JHV1 - 2	28V_RTN
JHV1 - 3	HV_MON_P1
JHV1 - 4	DAC_P
JHV1 - 5	HV_ENABLE_1
JHV1 - 6	HVBS +28V
JHV1 - 7	28V_RTN
JHV1 - 8	HV_MON_N1
JHV1 - 9	DAC_N

Connector JHV2 to High Voltage Bias Supply #2

Connector Pin	Signal Name
JHV2 - 1	HVBS +28V
JHV2 - 2	28V_RTN
JHV2 - 3	HV_MON_P2
JHV2 - 4	DAC_P
JHV2 - 5	HV_ENABLE_2
JHV2 - 6	HVBS +28V
JHV2 - 7	28V_RTN
JHV2 - 8	HV_MON_N2
JHV2 - 9	DAC_N

Appendix 7: Series II Circular Connector Pin List

Pin Locations for Connectors JP1 and (JS2)

Signal Name	Function	Connector Pin
ACD_VDD_0A(B)	+3.3V Power to ACD	1
ACD_VDD_1A(B)	+3.3V Power to ACD	3
ACD_VDD_2A(B)	+3.3V Power to ACD	4
ACD_28V_0A(B)	+28V Power to ACD HVBS	5
ACD_28V_1A(B)	+28V Power to ACD HVBS	7
ACD_NVETO_16A(B)P	Veto Ch 16 + to AEM	17
ACD_NVETO_16A(B)M	Veto Ch 16 - to AEM	18
ACD_NVETO_17A(B)P	Veto Ch 17 + to AEM	19
ACD_NVETO_17A(B)M	Veto Ch 17 - to AEM	20
ACD_NCNO_A(B)P	CNO + to AEM	21
ACD_NCNO_A(B)M	CNO - to AEM	22
ACD_HV_AP	High Voltage Supply 1 Monitor + to AEM	23
ACD_HV_AM	High Voltage Supply 1 Monitor - to AEM	24
ACD_TEMP_A(B)P	Temp Monitor + to AEM	25
ACD_TEMP_A(B)M	Temp Monitor - to AEM	26
ACD_HV_BP	High Voltage Supply 2 Monitor + to AEM	27
ACD_HV_BM	High Voltage Supply 2 Monitor - to AEM	28
ACD_GND_0A(B)	+3.3V Power Return	30
ACD_GND_1A(B)	+3.3V Power Return	31
ACD_GND_2A(B)	+3.3V Power Return	32
ACD_28V_RTN_0A(B)	+28V Return	33
ACD_28V_RTN_1A(B)	+28V Return	34
ACD_NVETO_15A(B)M	Veto Ch 15 - to AEM	40
ACD_NVETO_15A(B)P	Veto Ch 15 + to AEM	41
ACD_NVETO_14A(B)M	Veto Ch 14 - to AEM	42
ACD_NVETO_14A(B)P	Veto Ch 14 + to AEM	43
ACD_NVETO_13A(B)M	Veto Ch 13 - to AEM	44
ACD_NVETO_13A(B)P	Veto Ch 13 + to AEM	45
ACD_NVETO_12A(B)M	Veto Ch 12 - to AEM	46
ACD_NVETO_12A(B)P	Veto Ch 12 + to AEM	47
ACD_NVETO_11A(B)M	Veto Ch 11 - to AEM	48
ACD_NVETO_11A(B)P	Veto Ch 11 + to AEM	49
ACD_NVETO_10A(B)M	Veto Ch 10 - to AEM	50
ACD_NVETO_10A(B)P	Veto Ch 10 + to AEM	51
ACD_NVETO_09A(B)M	Veto Ch 9 - to AEM	52
ACD_NVETO_09A(B)P	Veto Ch 9 + to AEM	53
ACD_NVETO_08A(B)M	Veto Ch 8 - to AEM	54
ACD_NVETO_08A(B)P	Veto Ch 8 + to AEM	55
ACD_NVETO_07A(B)M	Veto Ch 7 - to AEM	56

ACD_NVETO_07A(B)P	Veto Ch 7 + to AEM	57
ACD_NVETO_06A(B)M	Veto Ch 6 - to AEM	58
ACD_NVETO_06A(B)P	Veto Ch 6 + to AEM	59
ACD_NVETO_05A(B)M	Veto Ch 5 - to AEM	60
ACD_NVETO_05A(B)P	Veto Ch 5 + to AEM	61
ACD_NVETO_04A(B)M	Veto Ch 4 - to AEM	62
ACD_NVETO_04A(B)P	Veto Ch 4 + to AEM	63
ACD_NVETO_03A(B)M	Veto Ch 3 - to AEM	64
ACD_NVETO_03A(B)P	Veto Ch 3 + to AEM	65
ACD_NVETO_02A(B)M	Veto Ch 2 - to AEM	66
ACD_NVETO_02A(B)P	Veto Ch 2 + to AEM	67
ACD_NVETO_01A(B)M	Veto Ch 1 - to AEM	68
ACD_NVETO_01A(B)P	Veto Ch 1 + to AEM	69
ACD_NVETO_00A(B)M	Veto Ch 0 - to AEM	70
ACD_NVETO_00A(B)P	Veto Ch 0 + to AEM	71
ACD_NSDATA_A(B)M	Data- to AEM	72
ACD_NSDATA_A(B)P	Data+ to AEM	73
ACD_NRST_A(B)M	Reset- to ACD	74
ACD_NRST_A(B)P	Reset+ to ACD	75
ACD_NSCMD_A(B)M	Command- to ACD	76
ACD_NSCMD_A(B)P	Command+ to ACD	77
ACD_CLK_A(B)M	Clock- to ACD	78
ACD_CLK_A(B)P	Clock+ to ACD	79
Spare Pins	Pins 8-16, 29, 35-39 not used	rest

Appendix 8: GARC Pin List

Tanner IO Cell	MOSIS Bond Pad Name	GARC Package Pin Number	GARC Signal Name
PADGnd	L1	1	DGND
PADAREF	L2	2	ACD_CLK_AP
PADAREF	L3	3	ACD_CLK_AM
PADAREF	L4	4	ACD_CLK_BP
PADAREF	L5	5	ACD_CLK_BM
PADAREF	L6	6	ACD_NSCMD_AP
PADAREF	L7	7	ACD_NSCMD_AM
PADAREF	K8	8	ACD_NSCMD_BP
PADAREF	L9	9	ACD_NSCMD_BM
PADAREF	L10	10	ACD_NRST_AP
PADAREF	L11	11	ACD_NRST_AM
PADAREF	L12	12	ACD_NRST_BP
PADAREF	L13	13	ACD_NRST_BM
PADAREF	L14	14	ACD_NSDATA_AP
PADAREF	L15	15	ACD_NSDATA_AM
PADAREF	L16	16	ACD_NSDATA_BP
PADAREF	L17	17	ACD_NSDATA_BM
PADAREF	L18	18	ACD_NCNO_AP
PADAREF	L19	19	ACD_NCNO_AM
PADAREF	L20	20	ACD_NCNO_BP
PADAREF	L21	21	ACD_NCNO_BM
PADAREF	L22	22	ACD_NVETO_00AP
PADAREF	L23	23	ACD_NVETO_00AM
PADVdd	L24	24	DVDD
PADAREF	L25	25	ACD_NVETO_00BP
PADAREF	L26	26	ACD_NVETO_00BM
PADGnd	L27	27	DGND
PADAREF	L28	28	ACD_NVETO_01AP
PADAREF	L29	29	ACD_NVETO_01AM
PADAREF	L30	30	ACD_NVETO_01BP
PADAREF	L31	31	ACD_NVETO_01BM
PADAREF	L32	32	ACD_NVETO_02AP
PADAREF	L33	33	ACD_NVETO_02AM
PADAREF	L34	34	ACD_NVETO_02BP
PADAREF	L35	35	ACD_NVETO_02BM
PADAREF	L36	36	ACD_NVETO_03AP
PADAREF	L37	37	ACD_NVETO_03AM
PADAREF	L38	38	ACD_NVETO_03BP
PADAREF	L39	39	ACD_NVETO_03BM
PADAREF	L40	40	ACD_NVETO_04AP
PADAREF	L41	41	ACD_NVETO_04AM
PADAREF	L42	42	ACD_NVETO_04BP
PADAREF	L43	43	ACD_NVETO_04BM
PADAREF	L44	44	ACD_NVETO_05AP
PADAREF	L45	45	ACD_NVETO_05AM
PADAREF	L46	46	ACD_NVETO_05BP
PADAREF	L47	47	ACD_NVETO_05BM
PADAREF	L48	48	ACD_NVETO_06AP
PADAREF	L49	49	ACD_NVETO_06AM
PADAREF	L50	50	ACD_NVETO_06BP
PADAREF	L51	51	ACD_NVETO_06BM
PADVdd	L52	52	DVDD
PADGnd	B1	53	DGND
PADAREF	B2	54	ACD_NVETO_07AP
PADAREF	B3	55	ACD_NVETO_07AM

PADAREF	B4	56	ACD_NVETO_07BP
PADAREF	B5	57	ACD_NVETO_07BM
PADAREF	B6	58	ACD_NVETO_08AP
PADAREF	B7	59	ACD_NVETO_08AM
PADAREF	B8	60	ACD_NVETO_08BP
PADAREF	B9	61	ACD_NVETO_08BM
PADAREF	B10	62	ACD_NVETO_09AP
PADAREF	B11	63	ACD_NVETO_09AM
PADAREF	B12	64	ACD_NVETO_09BP
PADAREF	B13	65	ACD_NVETO_09BM
PADAREF	B14	66	ACD_NVETO_10AP
PADAREF	B15	67	ACD_NVETO_10AM
PADAREF	B16	68	ACD_NVETO_10BP
PADAREF	B17	69	ACD_NVETO_10BM
PADAREF	B18	70	ACD_NVETO_11AP
PADAREF	B19	71	ACD_NVETO_11AM
PADAREF	B20	72	ACD_NVETO_11BP
PADAREF	B21	73	ACD_NVETO_11BM
PADAREF	B22	74	ACD_NVETO_12AP
PADAREF	B23	75	ACD_NVETO_12AM
PADAREF	B24	76	ACD_NVETO_12BP
PADAREF	B25	77	ACD_NVETO_12BM
PADVdd	B26	78	DVDD
PADAREF	B27	79	CHID_17
PADGnd	B28	80	DGND
PADAREF	B29	81	DISC_17
PADAREF	B30	82	IRTN_17
PADAREF	B31	83	CHID_16
PADAREF	B32	84	DISC_16
PADAREF	B33	85	IRTN_16
PADAREF	B34	86	CHID_15
PADAREF	B35	87	DISC_15
PADAREF	B36	88	IRTN_15
PADAREF	B37	89	CHID_14
PADAREF	B38	90	DISC_14
PADAREF	B39	91	IRTN_14
PADAREF	B40	92	CHID_13
PADAREF	B41	93	DISC_13
PADAREF	B42	94	IRTN_13
PADAREF	B43	95	CHID_12
PADAREF	B44	96	DISC_12
PADAREF	B45	97	IRTN_12
PADAREF	B46	98	CHID_11
PADAREF	B47	99	DISC_11
PADAREF	B48	100	IRTN_11
PADAREF	B49	101	CHID_10
PADAREF	B50	102	DISC_10
PADAREF	B51	103	IRTN_10
PADAREF	B52	104	HLD_WOR_BIAS
PADAREF	R52	105	CHID_09
PADAREF	R51	106	DISC_09
PADAREF	R50	107	IRTN_09
PADAREF	R49	108	CHID_08
PADAREF	R48	109	DISC_08
PADAREF	R47	110	IRTN_08
PADAREF	R46	111	CHID_07
PADAREF	R45	112	DISC_07
PADAREF	R44	113	IRTN_07
PADAREF	R43	114	CHID_06
PADAREF	R42	115	DISC_06
PADAREF	R41	116	IRTN_06
PADAREF	R40	117	CHID_05

PADAREF	R39	118	DISC_05
PADAREF	R38	119	IRTN_05
PADAREF	R37	120	CHID_04
PADAREF	R36	121	DISC_04
PADAREF	R35	122	IRTN_04
PADAREF	R34	123	CHID_03
PADAREF	R33	124	DISC_03
PADAREF	R32	125	IRTN_03
PADAREF	R31	126	CHID_02
PADAREF	R30	127	DISC_02
PADAREF	R29	128	IRTN_02
PADAREF	R28	129	CHID_01
PADAREF	R27	130	DISC_01
PADAREF	R26	131	IRTN_01
PADVdd	R25	132	DVDD
PADAREF	R24	133	CHID_00
PADAREF	R23	134	DISC_00
PADAREF	R22	135	IRTN_00
PADGnd	R21	136	DGND
PADInC	R20	137	PHAD_17
PADInC	R19	138	PHAD_16
PADInC	R18	139	PHAD_15
PADInC	R17	140	PHAD_14
PADInC	R16	141	PHAD_13
PADInC	R15	142	PHAD_12
PADInC	R14	143	PHAD_11
PADInC	R13	144	PHAD_10
PADInC	R12	145	PHAD_09
PADInC	R11	146	PHAD_08
PADInC	R10	147	PHAD_07
PADInC	R9	148	PHAD_06
PADInC	R8	149	PHAD_05
PADInC	R7	150	PHAD_04
PADInC	R6	151	PHAD_03
PADInC	R5	152	PHAD_02
PADInC	R4	153	PHAD_01
PADInC	R3	154	PHAD_00
PADVdd	R2	155	DVDD
PADAREF	R1	156	BIAS_RCVR
PADGnd	T52	157	DGND
PADAREF	T51	158	IRTN_HL_DISC
PADAREF	T50	159	OR_HL_DISC
PADAREF	T49	160	BIAS_DRV_H
PADAREF	T48	161	GAFE_HOLDP
PADAREF	T47	162	GAFE_HOLDM
PADAREF	T46	163	GAFE_STROBEP
PADAREF	T45	164	GAFE_STROBEM
PADAREF	T44	165	GAFE_DATP
PADAREF	T43	166	GAFE_DATM
PADAREF	T42	167	GAFE_CLKP
PADAREF	T41	168	GAFE_CLKM
PADAREF	T40	169	BIAS_DRV_L
PADInC	T39	170	GAFE_RET_DATA
PadOut	T38	171	GAFE_RST
PADOut	T37	172	ADC_CLK
PADOut	T36	173	NADC_CS
PADOut	T35	174	DAC_CLK
PADOut	T34	175	DAC_DATA
PADInC	T33	176	DAC_READBACK
PADOut	T32	177	NDAC_CS
PADOut	T31	178	NDAC_CLR
PADOut	T30	179	HITMAP_TEST

PADOut	T29	180	TRIG_TST
PADInC	T28	181	FREE_ID
PADInC	T27	182	PWR_ON_RST
PADOut	T26	183	VETO_ENA_TST
PADAREF	T25	184	LVDS_PRESETADJ
PADOut	T24	185	HV_ENABLE_1
PADOut	T23	186	HV_ENABLE_2
PADGnd	T22	187	DGND
PADAREF	T21	188	ACD_NVETO_17AP
PADAREF	T20	189	ACD_NVETO_17AM
PADAREF	T19	190	ACD_NVETO_17BP
PADAREF	T18	191	ACD_NVETO_17BM
PADAREF	T17	192	ACD_NVETO_16AP
PADAREF	T16	193	ACD_NVETO_16AM
PADAREF	T15	194	ACD_NVETO_16BP
PADAREF	T14	195	ACD_NVETO_16BM
PADAREF	T13	196	ACD_NVETO_15AP
PADAREF	T12	197	ACD_NVETO_15AM
PADAREF	T11	198	ACD_NVETO_15BP
PADAREF	T10	199	ACD_NVETO_15BM
PADAREF	T9	200	ACD_NVETO_14AP
PADAREF	T8	201	ACD_NVETO_14AM
PADAREF	T7	202	ACD_NVETO_14BP
PADAREF	T6	203	ACD_NVETO_14BM
PADAREF	T5	204	ACD_NVETO_13AP
PADAREF	T4	205	ACD_NVETO_13AM
PADAREF	T3	206	ACD_NVETO_13BP
PADAREF	T2	207	ACD_NVETO_13BM
PADVdd	T1	208	DVDD

Appendix 9: GAFE Pin List

Tanner IO Cell	MOSIS Bond Pad Name	GAFE Package Pin Number	GAFE Signal Name
PADTBD	L1	1	TBD
PADTBD	L1	1	TBD
PADTBD	L1	1	TBD
PADTBD	L1	1	TBD
PADTBD	L1	1	TBD
PADTBD	L1	1	TBD
PADTBD	L1	1	TBD

Appendix 10: GARC Command Mnemonics and Functions

The following table represents each of the available GARC commands. Additionally, all GAFE commands are passed through the GARC. These command patterns are detailed in the document discussing the GAFE logic. There are two types of GARC commands – trigger commands (4 bits in length) and configuration commands (34 bits in length). The AEM-ACD ICD contains the authoritive formats for all command types, but they are repeated in the appendices of this document for convenience. The table below defines the command mnemonics that will be used in this test.

Note that a GAFE will process a write command either for the address hard-wired to the chip address pins or to an address of 'h1F, the GAFE broadcast address. A GAFE will process a read command only for an address identical to the hard-wired address. It is an operational constraint that, for any given ACD circuit board, each GAFE must have a unique address.

GARC Cmd No.	ACD Command Mnemonic	Rd/Wr Status	Select GARC=0 GAFE=1	Function Block	Register Number	No. of Data Bits	Command Function
1	Trigger_ZS	N/A	N/A	N/A	N/A	N/A	ACD Trigger, Zero-Suppression Enable
2	Trigger_NOZS	N/A	N/A	N/A	N/A	N/A	ACD Trigger, No Zero-Suppression
3	GARC_Reset	W	0	0	1	0	Generates reset for GARC and GAFE registers
4	Veto_Delay_Wr	W	0	0	2	5	Sets Delay from Disc_In to VETO Out
5	Veto_Delay_Rd	R	0	0	2	5	Reads contents of Veto_Delay register
6	GARC_Cal_Strobe	W	0	0	3	0	Sends Calibration Strobe signal to all GAFEs
7	HVBS_Level_Wr	W	0	0	8	12	Sets GARC register value from which HVBS may be commanded in the science mode
8	HVBS_Level_Rd	R	0	0	8	12	Reads contents of HVBS Level register
9	SAA_Level_Wr	W	0	0	9	12	Sets GARC register value from which HVBS may be commanded when in the SAA
10	SAA_Level_Rd	R	0	0	9	12	Reads contents of SAA Level register
11	Use_HV_Normal	W	0	0	10	0	Sends 12 bit value in HVBS Level register to the MAX5121 DAC
12	DAC_HVReg_Rd	R	0	0	10	0	Reads MAX5121 DAC Config Register
13	Use_SAA_Normal	W	0	0	11	0	Sends 12 bit value in SAA Level register to the MAX5121 DAC
14	DAC_SAAReg_Rd	R	0	0	11	0	Reads MAX5121 DAC Config Register
15	Hold_Delay_Wr	W	0	0	12	7	Sets GARC Hold Delay
16	Hold_Delay_Rd	R	0	0	12	7	Reads back value of GARC Hold Delay register
17	Veto_Width_Wr	W	0	0	13	3	Sets width of the VETO signals
18	Veto_Width_Rd	R	0	0	13	3	Reads back width of the VETO width register
19	HitMap_Width_Wr	W	0	0	14	4	Sets width of HitMap pulses
20	HitMap_Width_Rd	R	0	0	14	4	Reads back value in the HitMap width register
21	HitMap_Deadtime_Wr	W	0	0	15	3	Sets stretch at end of HitMap pulses
22	HitMap_Deadtime_Rd	R	0	0	15	3	Sets stretch at end of HitMap pulses
23	Look_At_Me	W	0	1	4	16	GARC interface selection command (primary/secondary)
24	HitMap_Delay_Wr	W	0	1	8	5	Sets delay from Disc_In to HitMap pulse
25	HitMap_Delay_Rd	R	0	1	8	5	Reads back contents of the HitMap_Delay register
26	PHA_EN0_Wr	W	0	1	9	16	Enables and Disables PHA readout enable, channels 0 -15

27	PHA_EN0_Rd	R	0	1	9	16	Reads back contents of the PHA_EN0 register
28	VETO_EN0_Wr	W	0	1	10	16	Enables and Disables VETO signals, channels 0 -15
29	VETO_EN0_Rd	R	0	1	10	16	Reads back contents of the VETO_EN0 register
30	PHA_EN1_Wr	W	0	1	12	2	Enables and Disables PHA readout enable, channels 16 & 17
31	PHA_EN1_Rd	R	0	1	12	2	Reads back contents of the PHA_EN1 register
32	VETO_EN1_Wr	W	0	1	13	2	Enables and Disables VETO signals, channels 16 & 17
33	VETO_EN1_Rd	R	0	1	13	2	Reads back contents of the VETP_EN1 register
34	Max_PHA_Wr	W	0	1	15	5	Sets the Maximum number (limit) of PHA to be returned in a single event data packet
35	Max_PHA_Rd	R	0	1	15	5	Reads back the contents of the Max_PHA register
36	GARC_Mode_Wr	W	0	2	8	11	Sets values of GARC mode bits
37	GARC_Mode_Rd	R	0	2	8	11	Reads back the value of the GARC mode register
38	GARC_Status	R	0	2	9	6	Reads back the value of the GARC status register
39	GARC_Cmd_Reg	R	0	2	10	16	Reads back the value of the GARC command register
40	GARC_Diagnostic	R	0	2	11	16	Reads back the value of the GARC diagnostic register
41	GARC_Cmd_Rejects	R	0	2	12	8	Reads back the number of rejected commands since reset
42	FREE_Board_ID	R	0	2	13	8	Reads back the FREE board serial number
43	GARC_Version	R	0	2	14	3	Reads back the GARC ASIC version number
44	PHA_Thresh00_Wr	W	0	3	8	12	Writes PHA ZS threshold for channel 00
45	PHA_Thresh00_Rd	R	0	3	8	12	Reads back PHA ZS threshold register for channel 00
46	PHA_Thresh01_Wr	W	0	3	9	12	Writes PHA ZS threshold for channel 01
47	PHA_Thresh01_Rd	R	0	3	9	12	Reads back PHA ZS threshold register for channel 01
48	PHA_Thresh02_Wr	W	0	3	10	12	Writes PHA ZS threshold for channel 02
49	PHA_Thresh02_Rd	R	0	3	10	12	Reads back PHA ZS threshold register for channel 02
50	PHA_Thresh03_Wr	W	0	3	11	12	Writes PHA ZS threshold for channel 03
51	PHA_Thresh03_Rd	R	0	3	11	12	Reads back PHA ZS threshold register for channel 03
52	PHA_Thresh04_Wr	W	0	3	12	12	Writes PHA ZS threshold for channel 04
53	PHA_Thresh04_Rd	R	0	3	12	12	Reads back PHA ZS threshold register for channel 04
54	PHA_Thresh05_Wr	W	0	3	13	12	Writes PHA ZS threshold for channel 05
55	PHA_Thresh05_Rd	R	0	3	13	12	Reads back PHA ZS threshold register for channel 05
56	PHA_Thresh06_Wr	W	0	3	14	12	Writes PHA ZS threshold for channel 06
57	PHA_Thresh06_Rd	R	0	3	14	12	Reads back PHA ZS threshold register for channel 06
58	PHA_Thresh07_Wr	W	0	4	8	12	Writes PHA ZS threshold for channel 07
59	PHA_Thresh07_Rd	R	0	4	8	12	Reads back PHA ZS threshold register for channel 07
60	PHA_Thresh08_Wr	W	0	4	9	12	Writes PHA ZS threshold for channel 08
61	PHA_Thresh08_Rd	R	0	4	9	12	Reads back PHA ZS threshold register for channel 08
62	PHA_Thresh09_Wr	W	0	4	10	12	Writes PHA ZS threshold for channel 09
63	PHA_Thresh09_Rd	R	0	4	10	12	Reads back PHA ZS threshold register

							for channel 09
64	PHA_Thresh10_Wr	W	0	4	11	12	Writes PHA ZS threshold for channel 10
65	PHA_Thresh10_Rd	R	0	4	11	12	Reads back PHA ZS threshold register for channel 10
66	PHA_Thresh11_Wr	W	0	4	12	12	Writes PHA ZS threshold for channel 11
67	PHA_Thresh11_Rd	R	0	4	12	12	Reads back PHA ZS threshold register for channel 11
68	PHA_Thresh12_Wr	W	0	4	13	12	Writes PHA ZS threshold for channel 12
69	PHA_Thresh12_Rd	R	0	4	13	12	Reads back PHA ZS threshold register for channel 12
70	PHA_Thresh13_Wr	W	0	4	14	12	Writes PHA ZS threshold for channel 13
71	PHA_Thresh13_Rd	R	0	4	14	12	Reads back PHA ZS threshold register for channel 13
72	PHA_Thresh14_Wr	W	0	5	8	12	Writes PHA ZS threshold for channel 14
73	PHA_Thresh14_Rd	R	0	5	8	12	Reads back PHA ZS threshold register for channel 14
74	PHA_Thresh15_Wr	W	0	5	9	12	Writes PHA ZS threshold for channel 15
75	PHA_Thresh15_Rd	R	0	5	9	12	Reads back PHA ZS threshold register for channel 15
76	PHA_Thresh16_Wr	W	0	5	10	12	Writes PHA ZS threshold for channel 16
77	PHA_Thresh16_Rd	R	0	5	10	12	Reads back PHA ZS threshold register for channel 16
78	PHA_Thresh17_Wr	W	0	5	11	12	Writes PHA ZS threshold for channel 17
79	PHA_Thresh17_Rd	R	0	5	11	12	Reads back PHA ZS threshold register for channel 17
80	ADC_TACQ_Wr	W	0	5	12	6	Sets ADC Acquisition Time from Hold to Start of Conversion
81	ADC_TACQ_Rd	R	0	5	12	6	Reads back contents of the ADC_TACQ register
82	GAFE_Mode_Wr	W	1	GAFE Addr	0	16	Writes the GAFE mode register for the ASIC addressed
83	GAFE_Mode_Rd	R	1	GAFE Addr	0	16	Reads the GAFE mode register contents for the ASIC addressed
84	GAFE_DAC1_Wr	W	1	GAFE Addr	1	6	Writes the DAC1 register in the GAFE addressed
85	GAFE_DAC1_Rd	R	1	GAFE Addr	1	6	Reads back the contents of the DAC1 register in the addressed GAFE
86	GAFE_DAC2_Wr	W	1	GAFE Addr	2	6	Writes the DAC2 register in the GAFE addressed
87	GAFE_DAC2_Rd	R	1	GAFE Addr	2	6	Reads back the contents of the DAC2 register in the addressed GAFE
88	GAFE_DAC3_Wr	W	1	GAFE Addr	3	6	Writes the DAC3 register in the GAFE addressed
89	GAFE_DAC3_Rd	R	1	GAFE Addr	3	6	Reads back the contents of the DAC3 register in the addressed GAFE
90	GAFE_DAC4_Wr	W	1	GAFE Addr	4	6	Writes the DAC4 register in the GAFE addressed
91	GAFE_DAC4_Rd	R	1	GAFE Addr	4	6	Reads back the contents of the DAC4 register in the addressed GAFE
92	GAFE_DAC5_Wr	W	1	GAFE Addr	5	6	Writes the DAC5 register in the GAFE addressed
93	GAFE_DAC5_Rd	R	1	GAFE Addr	5	6	Reads back the contents of the DAC5 register in the addressed GAFE
94	GAFE_Version	R	1	GAFE Addr	6	3	Reads back the GAFE ASIC version
95	GAFE_Write_Ctr	R	1	GAFE Addr	7	8	Reads back the contents of the GAFE write counter register
96	GAFE_Reject_Ctr	R	1	GAFE Addr	8	8	Reads back the contents of the GAFE command reject register
97	GAFE_Cmd_Ctr	R	1	GAFE Addr	9	8	Reads back the contents of the GAFE command counter
98	GAFE_Chip_Addr	R	1	GAFE Addr	10	5	Reads back the hardwired address of a GAFE ASIC