credence

ASL Series Testers Application Note #3

Using the Dual Operational Amplifier Loop Option for the ASL 1000 Test Platform to Test a General Purpose Operational Amplifier

Contributed by: Ken Dickson

February 19, 2002

Abstract

This application note explains the various blocks of circuitry on the Dual Operational Amplifier Loop (DOAL) card. It then illustrates how to use the DOAL card to perform specific tests on an LM358 operational amplifier (op-amp). The tests covered are representative of tests commonly performed on op-amps and include:

- Swing (SW)
- Output drive (IOUT)
- Slew rate (SR)
- Input offset voltage (VOS/VIO)
- Input bias (IB)
- Gain bandwidth (GBW)
- Open loop voltage gain (AVD)
- Supply voltage rejection (SVR)
- Common mode rejection ration (CMR)

The availability of the LM358 and the simplicity of the test hardware and software make this application an ideal starting point for anyone wanting to test op-amps on the ASL 1000.

Introduction

The DOAL card option for the ASL 1000 test platform is used to test a broad spectrum of operational amplifiers. This application note describes how to connect to the DOAL to perform basic tests on a single op-amp. The op-amp tested is an LM358, a general purpose and readily available amplifier. The block diagram in Figure 1 shows the DOAL circuit blocks (enclosed by dashed lines).

Note: The text associated with portions of the blocks are syntactically correct from a software standpoint. Gray boxes are indicative of connections available on the test head. Test points, labeled TP, are available on the back edge of the DOAL card. These are generally black or white loop connectors and are labeled with the test point number and usually some descriptive text.

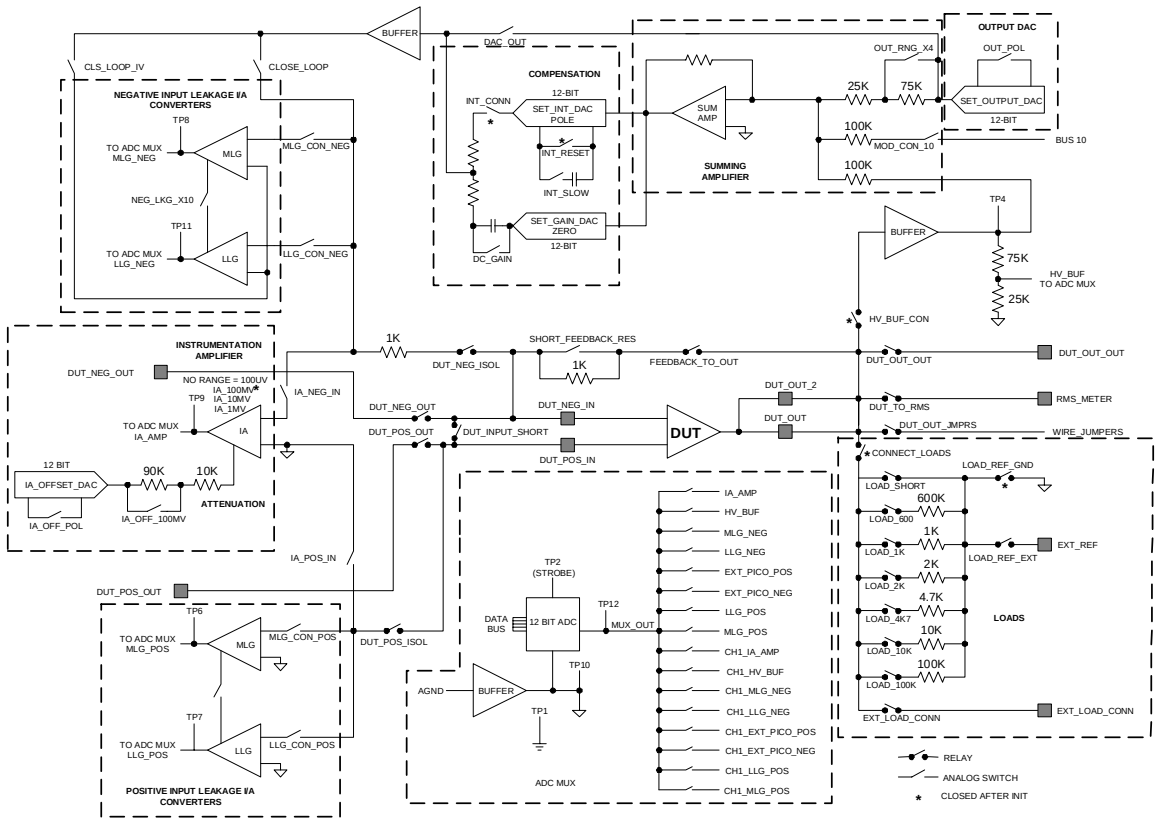


Figure 1. DOAL Channel 0 Block Diagram

DOAL Circuit Blocks

The DOAL card contains two generally independent loops suitable for testing two op-amps in parallel (simultaneously). Both loops are programmed concurrently by the same commands with the exception of the OUTPUT_DAC and measurement ADC, which are shared by the loops. Tests using them may have to be done sequentially.

Note: For this application note, only one amplifier was tested, so discussions are limited to Channel 0 to simplify software and drawings.

The op-amp loop allows the user to program the desired Device Under Test (DUT) output voltage. It does this by summing the output of the DUT with a desired programmed voltage

level from an OUTPUT_DAC creating an error signal. The OUTPUT_DAC is left justified with the lower 4 bits being ignored and is programmable from 0-65535 in increments of 16. Polarity is controlled by the OUT_POL switch (positive = off, negative = on). The Digital-to-Analog Converter (DAC) is 12 bits (4096 levels).

The error signal is then applied to a compensation circuit consisting of the INT DAC and GAIN DAC. These DACs are used respectively to set POLE and ZERO values to stabilize and maximize response of the loop. The output of the compensation circuit is then fed to the negative DUT input, completing the loop. The positive DUT input is usually referenced to ground. Both DACs are left justified with the lower 4 bits being ignored and a programmable range from 0-65535 in increments of 16. The DACs are 12 bits (4096 levels).

Output loads are provided on the DOAL card by resistors of various values. These resistors can be either ground referenced or externally biased. There is a provision for the user to connect external loads as well.

Most measurements make use of a ground referenced instrumentation amplifier (IA) with 3 programmable range relays of 1 mv, 10 mv and 100 mv. If all relays are open, the range will be 100 uv. The default range after init is the 100 mv range.

An offset DAC is connected to the instrumentation amplifier to null the IA to improve accuracy for ranges lower than 100 mv. The DAC is left justified with the lower 4 bits being ignored and is programmable from 0 to 65535 by increments of 16. Polarity is software selectable using the IA_OFF_POL switch (OFF = POSITIVE, ON = NEGATIVE). The DAC is 12 bits (4096 levels).

The DOAL contains leakage measurement circuitry in the form of I to V converters. These allow current measurements of from 1 to 10 ua (MLG) and 10 na (LLG) full scale. The medium-leakage (MLG) circuitry has a 1 ua and a 10 ua range selected by a X10 switch (OFF = 1 uA, ON = 10 uA). The MLG and low-leakage (LLG) I/V converters can be connected to the instrumentation amplifier for measurement.

Stabilizing the feedback loop

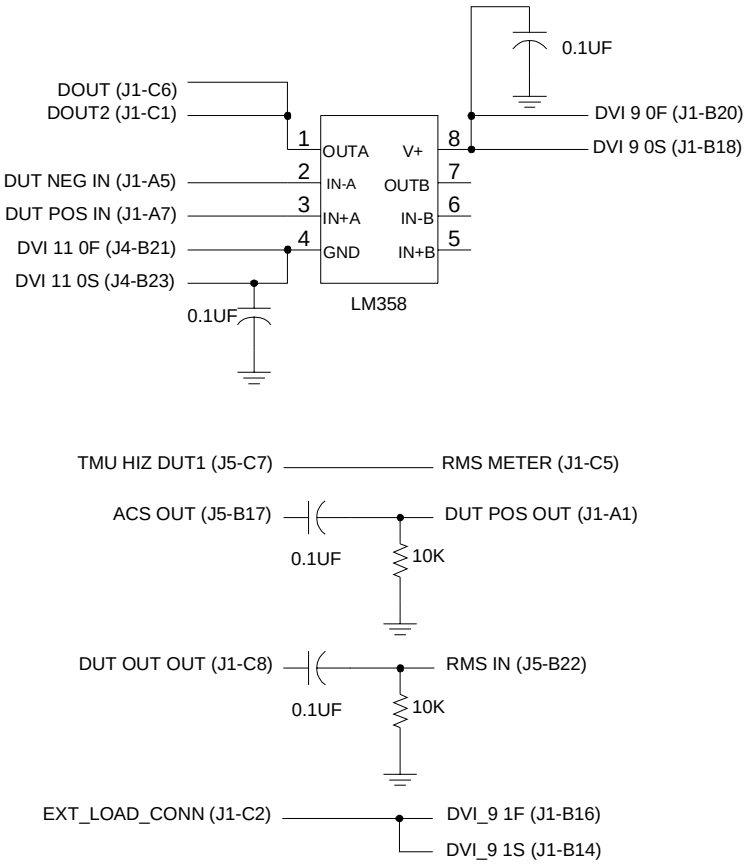
For tests that require the feedback loop, compensation must be adjusted in order to ensure that the DUT does not oscillate and to maximize loop response to reduce settling delays. Programming the INT DAC (POLE) and GAIN DAC (ZERO) accomplish this compensation. The adjustment should be made whenever test conditions change (for example, power supply, output voltage, and connecting other circuitry such as I/V converters). Usually, the ZERO value is between one-half and one-sixth the POLE value.

Operational Amplifier Tests

The following tests are designed to show as simply as possible how to connect the DOAL to test a single operational amplifier for a representative group of generic tests. There are more

complex and stringent tests required for other operational amplifiers, which are not covered here. A schematic of the DUT board, shown in Figure 2, used for this exercise shows the DOAL connections to the device to be tested, all required components, supplies, grounding, and the ASL 1000 system configuration requirements.

.



GND sense connections: connect J5-A26,C26 to GND plane
GND connections: solder all or some of these pins to connect GND plane to GND:
J5-A31,C31; J5-A21,C21; J5-A17,C17; J1-B2,B4; J2-B2,B4;
J6-A32,C32;J6-A30,C30; J6-A28,C28; J6 A26,C26; J6-A24,C24; J6-A22,C22;
J6-A20,C20; J6-A18,C18;

SYSTEM CONFIGURATION:

Resource	Slot
ACS	5
TMU	6
OAL	8
DVI	9
DVI	11
MUX	20

Figure 2. LM358 Single Op-Amp Demo Board

Swing Test

Swing is the ability of an op-amp to swing between the power supply rails. Figure 3 identifies the required connections for the swing test.

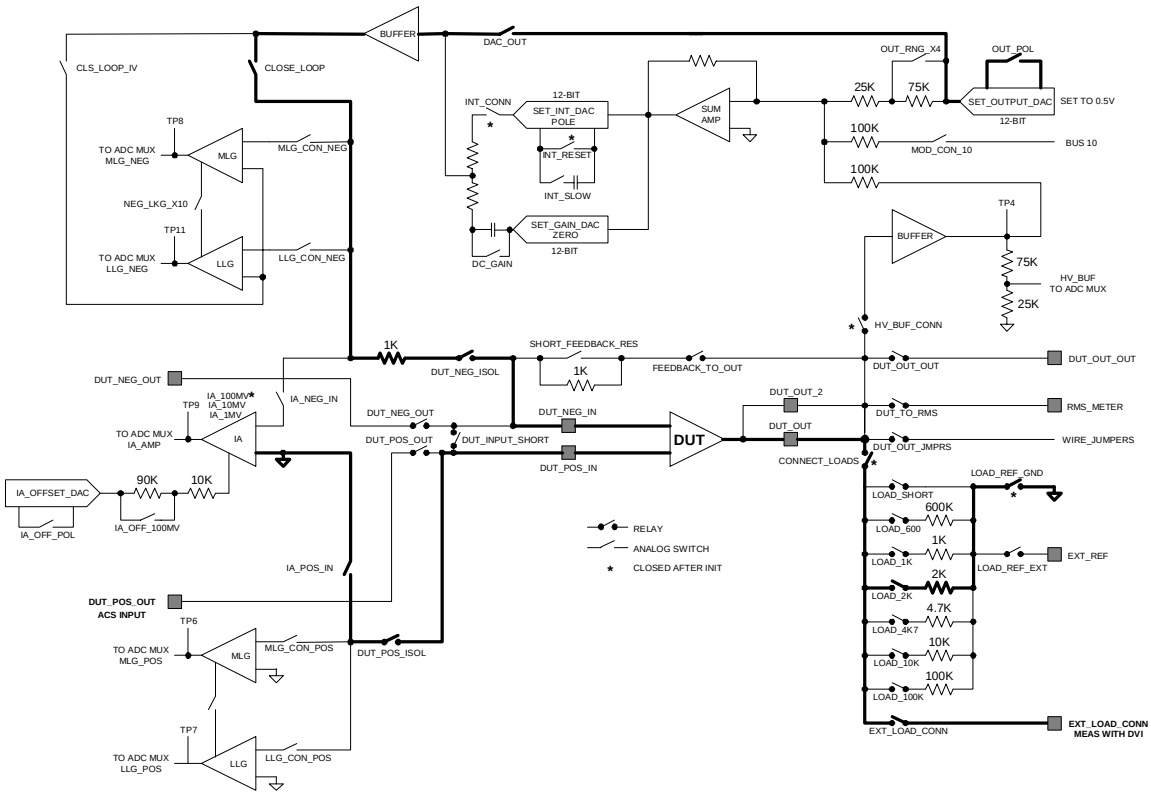


Figure 3. Swing Test

Procedure

With feedback disabled, use the following procedure to toggle the op-amp between the supply rails and measure the output to determine how close to the rails it can go.

1. Connect the OUTPUT_DAC to dut_neg in bypassing the compensation network.
2. Connect a 2 K load.
3. Set the OUTPUT_DAC to 0.5 V to get output to rail negative.
4. Measure the negative swing voltage.
5. Close the Out_Pol switch to get the output to rail positive.
6. Measure the positive swing voltage.

Software Code

The following code is used to determine swing.

```
// Definitions & Variables
float swing_pos, swing_neg;

oal_8->close_relay(EXT_LOAD_CONN);
oal_8->close_switch(DAC_OUT);
oal_8->close_switch(IA_POS_IN);
oal_8->close_switch(CLOSE_LOOP);
oal_8->open_switch(INT_CONN);
oal_8->close_relay(DUT_POS_ISOL);
oal_8->close_relay(DUT_NEG_ISOL);
dvi_9->set_current(DVI_CHANNEL_0, 0.2);
dvi_11->set_current(DVI_CHANNEL_0, 0.2);
dvi_9->set_meas_mode(DVI_CHANNEL_1, DVI_MEASURE_VOLTAGE);
dvi_9->set_voltage(DVI_CHANNEL_1, 40.0);//for meas range
dvi_9->set_current(DVI_CHANNEL_1, 0.1e-9);//high impedence

// VCC = 30V, load 2K
oal_8->close_relay(LOAD_2K);
dvi_9->set_voltage(DVI_CHANNEL_0, 30.0); //vplus
dvi_11->set_voltage(DVI_CHANNEL_0, 0.0);//gnd
oal_8->dac_output_voltage(ours->input_volts);
delay(ours->meas_delay);
swing_neg = dvi_9->measure_average(ours->samples);
oal_8->close_switch(OUT_POL); //switch polarity of DAC
delay(ours->meas_delay);
swing_pos = dvi_9->measure_average(ours->samples);

// Power down
power_down();

// Datalog
do_dlog(func, 0, swing_neg, ours->fail_bin, "swing neg", POWER_UNIT);
do_dlog(func, 1, swing_pos, ours->fail_bin, "swing pos", POWER_UNIT);
```

Output Drive (IOUT) Test

Output drive is the current sourcing and sinking capability of the op-amp output. Figure 4 identifies the required connections for the output drive test.

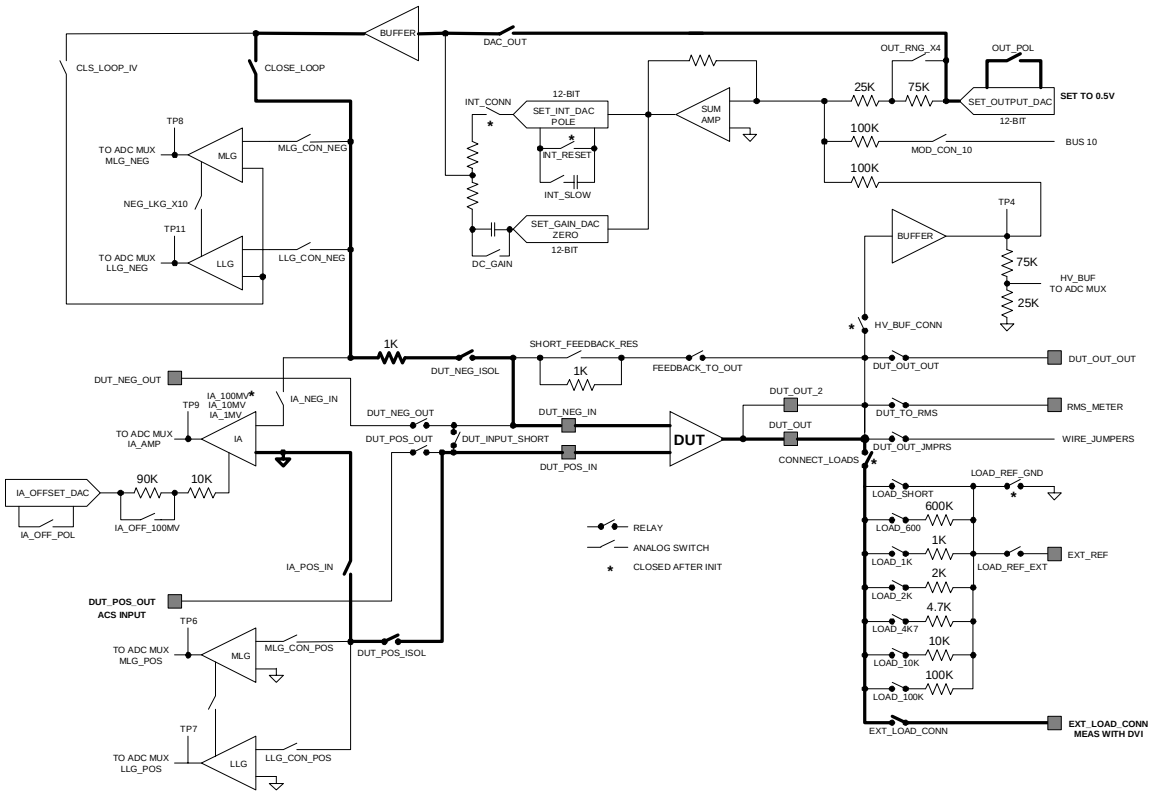


Figure 4. IOUT Test

Procedure

Switch the output rail to rail with a load midway between the rails. Use the following procedure to measure the source and sink currents at the negative and positive rails respectively.

1. Connect the OUTPUT_DAC to dut_neg in bypassing the compensation network.
2. Connect an external load using a dual voltage/current source instrument (DVI) as a load.
3. Set the load to 7.5 V, 100 ma (halfway between supply rails).
4. Set the OUTPUT_DAC to 0.5 V to get output to rail negative.
5. Measure the sink current.
6. Close the Out_pol switch to get the output to rail positive.
7. Measure the source current.

Software Code

The following code is used to determine the sourcing and sinking capability of the op-amp output.

```
// Definitions & Variables
float isource, isink;

oal_8->close_relay(EXT_LOAD_CONN);
oal_8->open_relay(HV_BUF_CONN);
oal_8->close_switch(DAC_OUT);
oal_8->close_switch(IA_POS_IN);
oal_8->close_switch(CLOSE_LOOP);
oal_8->open_switch(INT_CONN);
oal_8->close_relay(DUT_POS_ISOL);
oal_8->close_relay(DUT_NEG_ISOL);
dvi_9->set_current(DVI_CHANNEL_0, 0.2);
dvi_11->set_current(DVI_CHANNEL_0, 0.2);
dvi_9->set_meas_mode(DVI_CHANNEL_1, DVI_MEASURE_CURRENT);

//Sink Current
dvi_9->set_voltage(DVI_CHANNEL_0, ours->v_plus); //VCC
dvi_11->set_voltage(DVI_CHANNEL_0, ours->v_minus); //GND
dvi_9->set_voltage(DVI_CHANNEL_1, 7.5); //load voltage, midway between rails
dvi_9->set_current(DVI_CHANNEL_1, 100.0e-3); //load clamp
oal_8->dac_output_voltage(ours->input_v); // set output lo for sink
delay(ours->meas_delay);
isink = dvi_9->measure_average(ours->samples);

//Source Current
oal_8->close_switch(OUT_POL); //switch polarity of DAC, output hi for source
delay(ours->meas_delay);
isource = dvi_9->measure_average(ours->samples);

// Power down
power_down();

// Datalog
do_dlog(func, 0, isink,  ours->fail_bin, "isink",   POWER_MILLI);
do_dlog(func, 1, isource, ours->fail_bin, "isource", POWER_MILLI);
```

Slew Rate (SR) Test

Slew rate is the maximum rate at which the output can change (typically in terms of V/usec). Figure 5 identifies the required connections for the slew test.

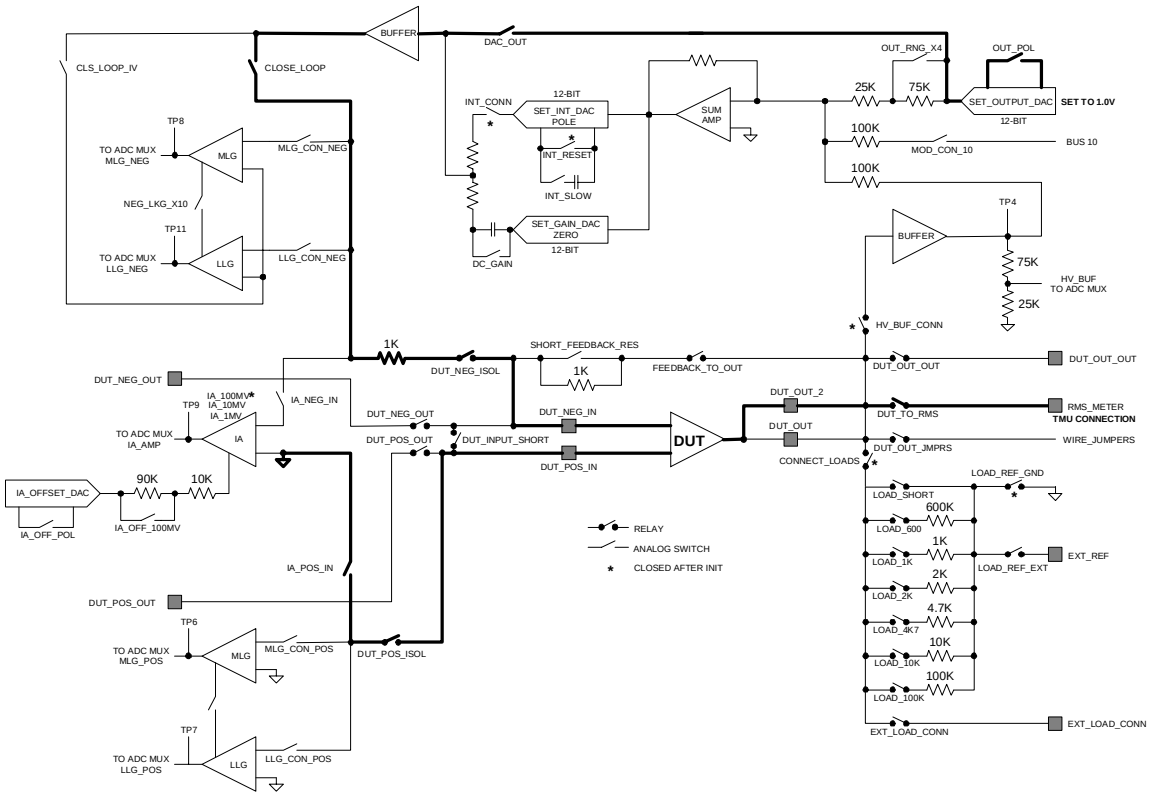


Figure 5. Slew Test

Procedure

The following procedure is used to perform the slew test.

1. With feedback disabled, rail the output by programming a voltage on the OUTPUT_DAC.
2. Set up the TMU for a rise/fall time measurement and toggle the OUT_POL switch to get the output to go to the other extreme.
3. Set the OUTPUT_DAC to 1 V.
4. Bypass the compensation circuitry.
5. Connect the buffered OUTPUT_DAC to the negative input.
6. Connect the positive input to ground (GND).
7. Connect the output to the time measurement unit (TMU) instrument.
8. Set up the TMU for rise time measurement and arm it.

9. Close the OUT_POL switch to set the output positive.
10. Read the TMU value.
11. If the value is 0, assign a non-zero value to prevent divide by zero errors later.
12. Scale the result in microseconds (time *= 1e6). Rise time = (high threshold value – low threshold value)/time.
13. Set up the TMU for a fall time measurement.
14. Open the OUT_POL switch to set the output negative.
15. Read the TMU value.
16. If the value is 0, assign a non-zero value to prevent divide by zero errors later.
17. Scale the measurement in microseconds (time *= 1e6).
18. Calculate the result. Fall time = (high threshold value – low threshold value)/ time.

Software Code

The following code is used to determine slew rate.

```
// Definitions & Variables
float rise_time, fall_time;

// power up and test
dvi_9->set_voltage(DVI_CHANNEL_0, ours->v_plus);
dvi_11->set_voltage(DVI_CHANNEL_0, ours->v_minus);
dvi_9->set_current(DVI_CHANNEL_0, 0.02);
dvi_11->set_current(DVI_CHANNEL_0, 0.02);
oal_8->close_relay(DUT_NEG_ISOL);
oal_8->close_relay(DUT_POS_ISOL);
oal_8->open_relay(HV_BUF_CONN);
oal_8->close_switch(DAC_OUT);
oal_8->close_switch(IA_POS_IN);
oal_8->close_switch(CLOSE_LOOP);
oal_8->dac_output_voltage(1);
oal_8->close_relay(OUT_TO_RMS);
tmu_6->close_relay(TMU_HIZ_DUT);
delay(1);
tmu_6->start_trigger_setup(ours->low_thresh,POS_SLOPE,TMU_HIZ);
tmu_6->stop_trigger_setup(ours->high_thresh,POS_SLOPE,TMU_HIZ);
delay(1);
tmu_6->arm();
oal_8->close_switch(OUT_POL);// set output positive
```

```
delay(ours->meas_delay);
if((rise_time = tmu_6->read()) == 0.0)
rise_time = 1e-12;
rise_time *= 1e6;
rise_time = (ours->high_thresh - ours->low_thresh) / rise_time;

tmu_6->start_trigger_setup(ours->high_thresh,NEG_SLOPE,TMU_HIZ);
tmu_6->stop_trigger_setup(ours->low_thresh,NEG_SLOPE,TMU_HIZ);
delay(1);
tmu_6->arm();
oal_8->open_switch(OUT_POL);// set output negative
delay(ours->meas_delay);
if((fall_time = tmu_6->read()) == 0.0)
fall_time = 1e-12;
fall_time *= 1e6;
fall_time = (ours->high_thresh - ours->low_thresh) / fall_time;

// Power down
power_down();

// Datalog
do_dlog(func,0,rise_time,ours->fail_bin,"rise time", POWER_UNIT);
do_dlog(func,1,fall_time,ours->fail_bin,"fall time", POWER_UNIT);
```

VOS/VIO Test

VOS or VIO is the input voltage required to force the output to zero volts (typically 100 uv to 2 mv). Figure 6 identifies the required connections for the VOS/VIO test.

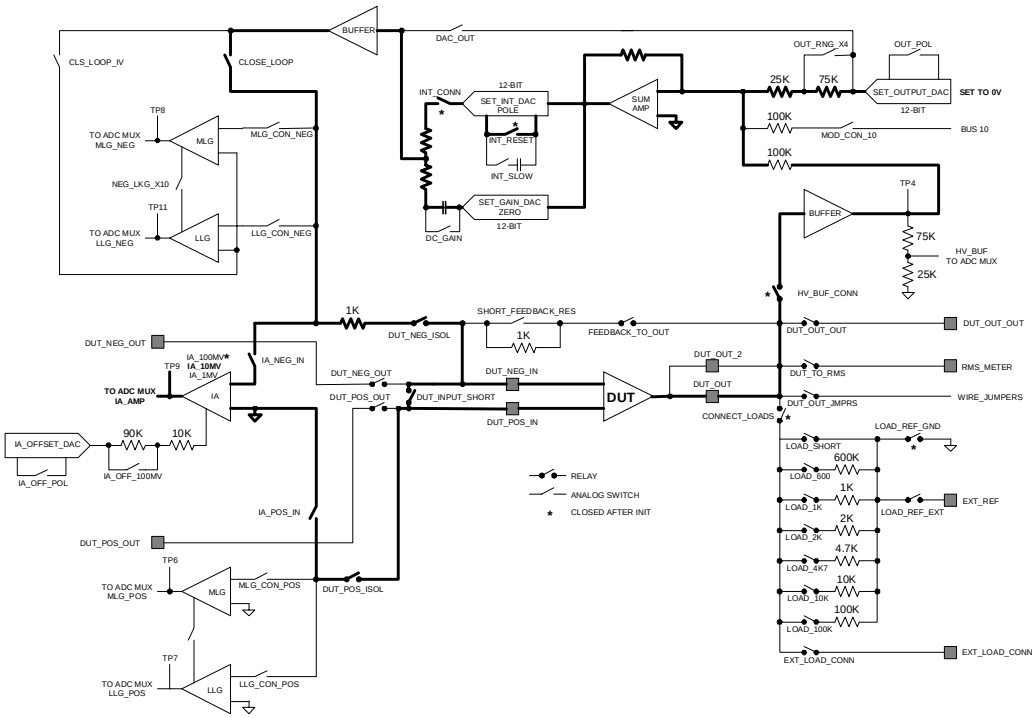


Figure 6. VOS/VIO Test

Procedure

Force the output to 0 V using the feedback loop and OUTPUT_DAC. Use the following procedure to measure the difference between the inputs with the instrument amplifier.

1. Connect the feedback loop.
2. Open the INT_RESET switch on the INT_DAC.
3. Set the pole and the zero values required to achieve circuit stability.
4. Measure the offset with the DUT inputs shorted.
5. Apply a reference voltage to the OUTPUT_DAC.
6. Measure the offset voltage and subtract it from the previously measured value.

Software Code

The following code is used to determine VOS or VIO.

```
// Definitions & Variables
int temp, cal;
float vos;

// set up for 10mV measure range
oal_8->open_switch(IA_100MV);
oal_8->close_switch(IA_10MV);

// do cal with inputs shorted
oal_8->close_relay(DUT_POS_ISOL);
oal_8->close_relay(DUT_NEG_ISOL);
oal_8->close_relay(DUT_INP_SHORT);
oal_8->close_switch(IA_POS_IN);
oal_8->close_switch(IA_NEG_IN);
oal_8->select_adc_mux(IA_AMP);
delay(ours->meas_delay);
cal = oal_8->convert_read_adc();

// set up to test Vos
oal_8->open_relay(DUT_INP_SHORT);
oal_8->close_switch(CLOSE_LOOP);
dvi_9->set_current(DVI_CHANNEL_0, 0.02);
dvi_11->set_current(DVI_CHANNEL_0, 0.02);

// set compensation
oal_8->set_int_dac_ch0 (ours->pole_stab);
oal_8->set_gain_dac_ch0(ours->zero_stab);

// power up and get measurement
dvi_9->set_voltage(DVI_CHANNEL_0, 3.0);
dvi_11->set_voltage(DVI_CHANNEL_0, -3.0);
oal_8->set_output_voltage(0);
oal_8->open_switch(INT_RESET);
delay(ours->meas_delay);
temp = oal_8->convert_read_adc() - cal;
vos = temp / 3276800.0;   // 32768 = 10mV

// Power down
power_down();

// Datalog
do_dlog(func, 0, vos, ours->fail_bin, "vos", POWER_MILLI);
```

Input Bias (IB) Test

Input bias is the input current that flows into each DUT input as a consequence of normal operation (typically 500 pa to 100 na). Figure 7 identifies the required connections for the input bias test.

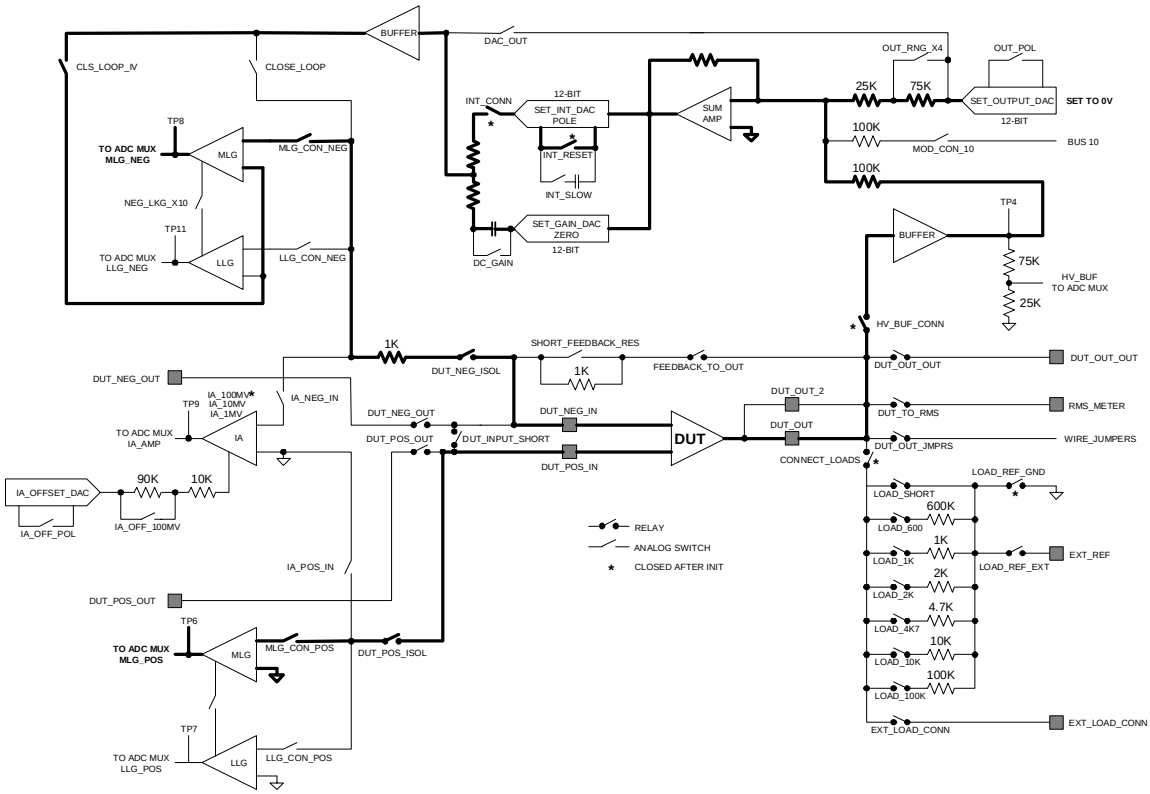


Figure 7. IB Test

Procedure

Using the following procedure, connect and stabilize the feedback loop. Connect the MLG I/V converters to the positive and negative inputs. You must shift power supplies to move GND on the positive input to obtain voltages other than GND on the positive and negative inputs. Force a value on the OUTPUT_DAC that corresponds to the power supply shift.

1. Connect feedback loop and any required output load.
2. Open INT_RESET switch on INT_DAC.
3. Set pole and zero values required for circuit stability.
4. Apply reference voltage to OUTPUT_DAC.

5. Measure positive leakage using MLG_POS (1 uA range) and ADC.
6. Scale result for 1 uA range: pos_lkg = (value-32768.0)/(32768*1e-6).
7. Measure negative leakage using MLG_NEG (1 uA range) and ADC.
8. Scale result for 1 uA range: neg_lkg = (value-32768.0)/(32768*1e-6).
9. Calculate offset current: pos_lkg-neg_lkg.

Software Code

The following code is used to determine input bias.

```
// Definitions & Variables
unsigned int i;
long temp;
float ib_pos, ib_neg, ib_offset;

// setup OAL
oal_8->close_relay(DUT_NEG_ISOL);
oal_8->close_relay(DUT_POS_ISOL);
oal_8->close_switch(MLG_CON_POS);
oal_8->close_switch(MLG_CON_NEG);
oal_8->select_adc_mux(MLG_POS);
oal_8->set_output_voltage(ours->v_output);
oal_8->close_switch(CLS_LOOP_IV);
oal_8->open_switch(INT_RESET);

// set compensation
oal_8->set_int_dac_ch0 (ours->pole_stab);
oal_8->set_gain_dac_ch0(ours->zero_stab);

// power up dut
dvi_9->set_current(DVI_CHANNEL_0, 0.02);
dvi_11->set_current(DVI_CHANNEL_0, 0.02);
dvi_9->set_voltage(DVI_CHANNEL_0, ours->v_plus);
dvi_11->set_voltage(DVI_CHANNEL_0, ours->v_minus);

// get POS measurements
delay(ours->meas_delay);
temp = 0;
for (i=0; i<ours->samples; i++)
{
temp += oal_8->convert_read_adc();
}
temp /= ours->samples;
ib_pos = (float(temp) - 32768.0)/(32768*1e6);//32768 is mid scale, convert from uA for datalog

// get NEG measurements
oal_8->select_adc_mux(MLG_NEG);
delay(ours->meas_delay);
temp = 0;
for (i=0; i<ours->samples; i++)
```

```
{
temp += oal_8->convert_read_adc();
}
temp /= ours->samples;
ib_neg = (float(temp) - 32768.0)/(32768*1e6);//32768 is mid scale, convert from uA for datalog
ib_offset = ib_pos - ib_neg;

// power down
power_down();

// Datalog
do_dlog(func, 0, ib_pos,ours->fail_bin, "ib pos", POWER_NANO);
do_dlog(func, 1, ib_neg,ours->fail_bin, "ib_neg", POWER_NANO);
do_dlog(func, 2, ib_offset, ours->fail_bin, "ib offset", POWER_NANO);
```

Gain Bandwidth (GBW) Test

Gain bandwidth is the frequency for which an op-amp has a voltage gain equal to 1. Figure 8 identifies the required connections for the gain bandwith test.

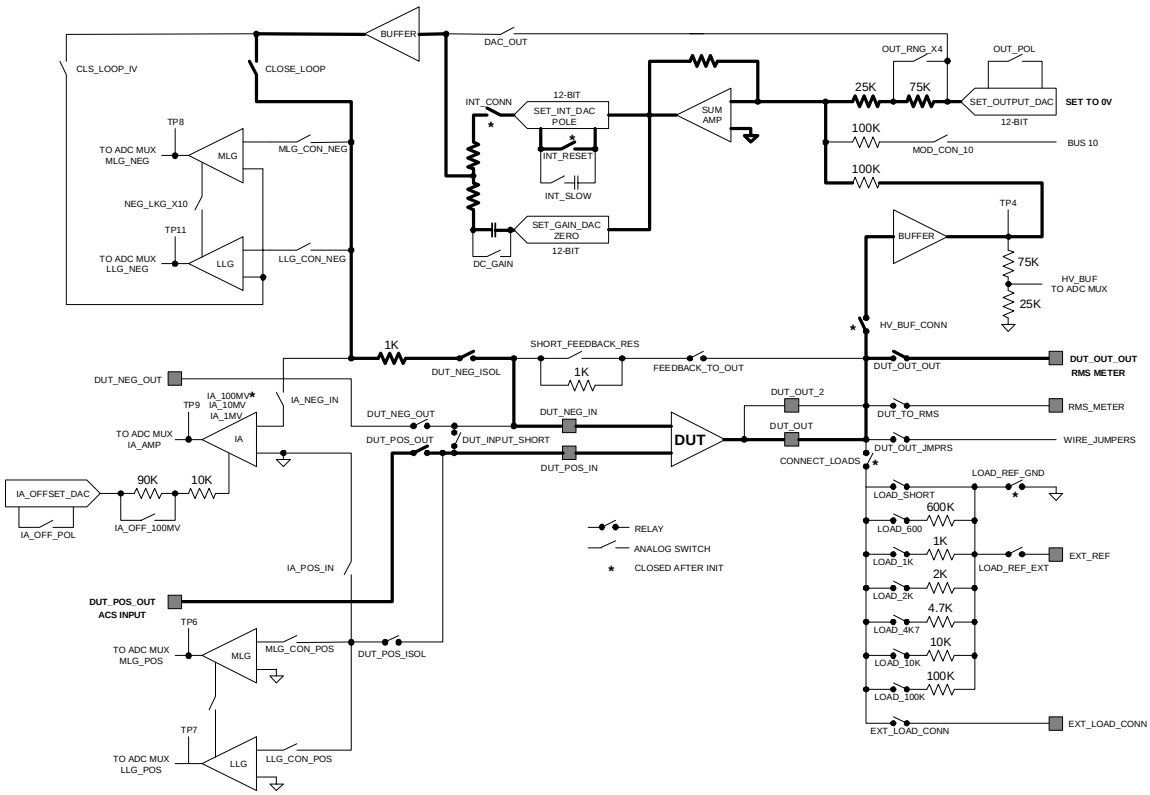


Figure 8. GBW Test

Procedure

Using the following procedure, connect and stabilize the feedback loop. Connect the AC source to the positive input. Start a frequency. Measure the output and calculate the GBW.

1. Connect the feedback loop.
2. Open the INT_RESET switch on the INT_DAC.
3. Set the pole and the zero values required for circuit stability.
4. Apply a reference voltage to the OUTPUT_DAC (0 V).
5. Connect the positive input to the ACS via DUT_POS_OUT.
6. Connect the ACS RMS METER to the DUT output via DUT_OUT_OUT.
7. Set the ACS to output a 100 khz sine wave of 10 mVRMS amplitude.
8. Measure the RMS out.
9. Calculate the GBW: GBW = input_freq * (rms_out /rms_in).

Software Code

The following code is used to determine gain bandwith.

```
// Definitions & Variables
float rms_out, GBW;

// OAL setup
oal_8->close_switch(INT_RESET);
oal_8->close_relay(DUT_NEG_ISOL);
oal_8->close_switch(CLOSE_LOOP);
oal_8->close_relay(DUT_POS_OUT);
oal_8->close_relay(DUT_OUT_OUT);
oal_8->open_switch(INT_RESET);
oal_8->set_output_voltage(ours->output_dc);

// set compensation
oal_8->set_int_dac_ch0 (ours->pole_stab);
oal_8->set_gain_dac_ch0(ours->zero_stab);

// Apply power
dvi_9->set_current(DVI_CHANNEL_0, 0.3);
dvi_11->set_current(DVI_CHANNEL_0, 0.3);
dvi_9->set_voltage(DVI_CHANNEL_0, ours->v_plus);
dvi_11->set_voltage(DVI_CHANNEL_0, ours->v_minus);
```

```
// Setup the ACS for output
acs_5->set_path(ACS_BASE_FILTER,ACS_1_VOLT_RANGE);
acs_5->ldram(ours->input_freq);
acs_5->go_rms();
acs_5->close_relay(DUT_SIG_OUT);
acs_5->set_level(ours->input_rms,0,ACS_RMS_LEVEL);

// Setup the ACS for rms measure
acs_5->close_relay(D_RMS_CH1);
acs_5->close_relay(SEL_ADC_IN0);
acs_5->set_meas_mode(ACS_CHANNEL_1, ACS_1V_RMS_RANGE, ACS_DC_COUPLING);
delay(ours->meas_delay);
rms_out = acs_5->measure(10);
GBW =  ours->input_freq * (rms_out / ours->input_rms);

// Power down
power_down();

// Datalog
do_dlog(func, 0, GBW, ours->fail_bin, "GBW", POWER_MEGA);
```

Open Loop Voltage Gain (AVD) Test

Open loop voltage gain is the gain from inputs to output with no feedback. Figure 9 identifies the required connections for the open loop voltage gain test.

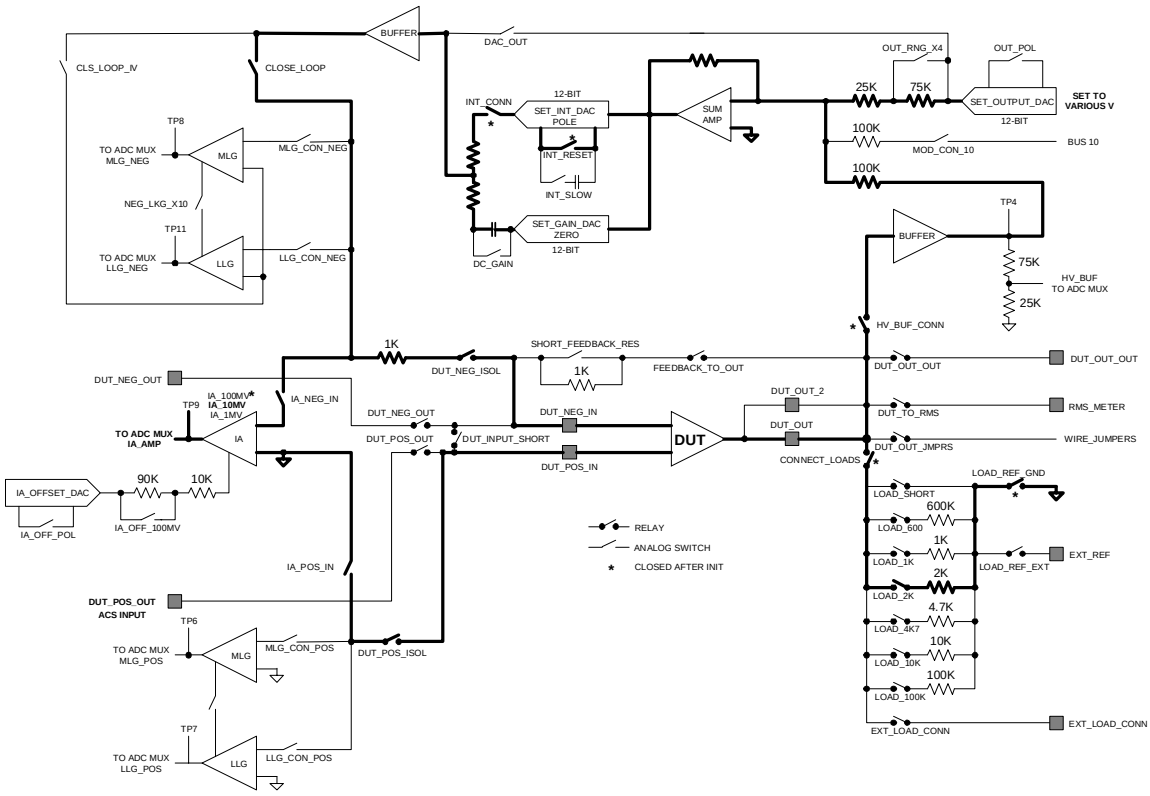


Figure 9. AVD, SVR, CMR Tests

Procedure

Using the following procedure, measure the inputs at two different output voltages, and then compare the difference of the inputs to the difference of the outputs.

1. Connect the feedback loop and any required output load.
2. Open the INT_RESET switch on the INT_DAC.
3. Set the pole and the zero values required for circuit stability.
4. Measure the inputs differentially with the OUTPUT_DAC set to 2 V.
5. Measure inputs differentially with OUTPUT_DAC set to 13 V.
6. Scale the results as needed.
7. AVD = 11 / (input_diff_13 V – input_diff_2 V).

Supply Voltage Rejection Ratio (SVR) Test

Supply voltage rejection ratio is the op-amp's ability to ignore a change in power supply voltage (typically 90 dB). This requires measuring the inputs with the output held constant using two different power supply conditions. Compare the input differences to the supply difference.

Procedure

Using the setup shown in Figure 9, measure the inputs with the output held constant using two different power supply conditions. Compare the input differences to the supply difference using the following procedure.

1. Connect the feedback loop and any required output load.
2. Open the INT_RESET switch on the INT_DAC.
3. Set the pole and the zero values required for circuit stability.
4. Program the OUTPUT_DAC to 2.0 V.
5. Measure the inputs differentially (A).
6. Change the supplies by -16 V.
7. Program the OUTPUT_DAC to 2.0 V.
8. Measure the inputs differentially (B).
9. Scale the ADC results as needed.
10. SVR = 20 * log10 (16 / (A – B)).

Common Mode Voltage Rejection Ratio (CMR) Test

Common mode voltage rejection ratio is the ability of an op-amp to ignore input changes that are common to both op-amp inputs.

Procedure

Using the following procedure, move inputs between supply rails by programming the output to the same voltage relative to the negative supply while moving supply voltages.

1. Connect the feedback loop and any required output load.

2. Open the INT_RESET switch on the INT_DAC.
3. Set the pole and the zero values required for circuit stability.
4. Set the OUTPUT_DAC to 2.0 V (2.0 V above negative rail).
5. Measure the inputs differentially (A).
6. Change the supplies by –18 V.
7. Set the OUTPUT_DAC to –16 V (2.0 V above negative rail).
8. Measure the inputs differentially (B).
9. Scale the ADC results as needed.
10. CMR = 20 * log10 (18 / (A – B)).

Software Code

The following code is used to determine open loop voltage gain, supply voltage rejection, and common mode rejection ration.

```
// Definitions & Variables
int i, tests;
long temp_val;
int adc_val[5];
float AVD, SVR, CMR, inp_change;
// ia offset adjust not needed for this device
oal_8->open_switch(IA_100MV);
oal_8->close_switch(IA_10MV);
oal_8->close_relay(DUT_POS_ISOL);
oal_8->close_relay(DUT_NEG_ISOL);
oal_8->close_switch(IA_POS_IN);
oal_8->close_switch(IA_NEG_IN);
oal_8->select_adc_mux(IA_AMP);
oal_8->close_switch(CLOSE_LOOP);
dvi_9->set_current(DVI_CHANNEL_0, 0.2);
dvi_11->set_current(DVI_CHANNEL_0, 0.2);

// set compensation
oal_8->set_int_dac_ch0 (ours->pole_stab);
oal_8->set_gain_dac_ch0(ours->zero_stab);

// initial power up
dvi_9->set_voltage(DVI_CHANNEL_0, 15.0);// VCC
dvi_11->set_voltage(DVI_CHANNEL_0, 0.0);// GND
oal_8->open_switch(INT_RESET);
oal_8->close_relay(LOAD_2K);
```

```
oal_8->close_relay(CONNECT_LOADS);

// perform tests
for(tests=0; tests<5; tests++)
{
switch(tests)
{
case 0:// Vio 15V sup 2K load Vout = 2.0V
oal_8->set_output_voltage(2.0);
break;

case 1:// Vio 15V sup 2K load Vout = 13V
oal_8->set_output_voltage(13.0);
break;

case 2:// Vio 20V sup no load Vout = 2.0V
oal_8->open_relay(CONNECT_LOADS);
dvi_9->set_voltage(DVI_CHANNEL_0, 20.0);
oal_8->set_output_voltage(2.0);
break;

case 3:// Vio +4 -16V sup no load Vout = 2.0V
dvi_9->set_voltage(DVI_CHANNEL_0, 4.0);
dvi_11->set_voltage(DVI_CHANNEL_0, -16.0);
break;

case 4:// Vio 2.0 -18.0V sup no load Vout = -16.0V
dvi_9->set_voltage(DVI_CHANNEL_0, 2.0);
dvi_11->set_voltage(DVI_CHANNEL_0, -18.0);
oal_8->set_output_voltage(-16.0);
break;
}
delay(ours->meas_delay);
//average results
temp_val = 0;
for(i=0; i<ours->samples; i++)
{
temp_val += oal_8->convert_read_adc();
}
adc_val[tests] = temp_val / ours->samples;
}
// power down
power_down();

// calculate open loop voltage gain (AVD)
inp_change = adc_val[1] - adc_val[0];
inp_change = fabs(inp_change / 3276800.0)*1000;//1000 for V/mv
if(inp_change == 0.0) inp_change = 1.0;
AVD = 11.0 / inp_change;

// calculate supply voltage rejection ratio (SVR)
inp_change = adc_val[3] - adc_val[2];
inp_change = fabs(inp_change / 3276800.0);
if(inp_change == 0.0) inp_change = 1.0;
```

```
SVR = 20.0 * log10(16.0 / inp_change);

// calculate common mode voltage rejection ratio (CMR)
inp_change = adc_val[4] - adc_val[2];
inp_change = fabs(inp_change / 3276800.0);
if(inp_change == 0.0) inp_change = 1.0;
CMR = 20.0 * log10(18.0 / inp_change);

//Datalog
do_dlog(func, 0, AVD, ours->fail_bin_avd, "AVD", POWER_UNIT);
do_dlog(func, 1, SVR, ours->fail_bin_svr, "SVR", POWER_UNIT);
do_dlog(func, 2, CMR, ours->fail_bin_cmr, "CMR", POWER_UNIT);
```

Test Results (Actual Datalog)

Table 1 provides the results of the actual device testing.

Table 1. Test Results

Test#	Test Name	Value	P/F	Unit	Min Limit	Max Limit
001.01.01	vio	0.9912	1	mV	-9.5	9.5
001.02.01	ib pos	-10.8337	1	nA	-200	200
001.02.02	ib neg	-12.2681	1	nA	-200	200
001.02.03	i offset	1.4343	1	nA	-200	200
001.03.01	slew+	0.3910	1	V/uS	0.2	1
001.03.02	slew-	0.3637	1	V/uS	0.2	1
001.04.01	GBW	1.0424	1	MHz	1	5.6
001.05.01	AVD	522.3884	1	V/mV	55	0
001.05.02	SVR	101.7625	1	dB	67	0
001.05.03	CMR	102.2925	1	dB	72	0
001.06.01	isink	31.4781	1	mA	10.5	50
001.06.02	isource	-41.1342	1	mA	-57	-21
001.07.01	Swing_neg	0.0365	1	V	3.52	2
001.07.02	Swing_pos	28.2406	1		28	12