LS-71

Hardware User's Manual

16/32-Channel Analog Output

Document:	U710102
Editor:	D. Spielman
Date:	8/09/2006
Release:	Rev-1

Lumistar, Inc.
2701 Loker Ave. West, Suite 230
Carlsbad, CA 92010
(760) 431-2181
www.lumi-star.com

Lumistar, Inc.
2701 Loker Avenue West
Suite 230
Carlsbad, CA 92010
(760) 431-2181
(760) 431-2665 Fax
www.lumi-star.com

TABLE OF CONTENTS

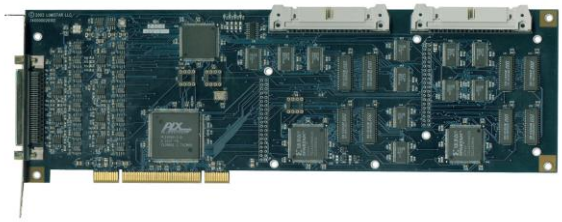
List of Tables

List of Figures

1 Introduction

1.1 General

The Lumistar LS-71 16/32-Channel Analog Output PCI card enables real-time displays of rapidly changing data using strip-chart displays and similar technology. Table-Lookup techniques permit virtually any linear or nonlinear Engineering Unit (EU) conversion between the data and the Digital/Analog Converter (DAC) Outputs.



The LS-71 supports in the base configuration one decommutator parallel output port and up to 32 analog output channels provided in groups of sixteen.

The Word Selector at the input port works for telemetry formats with up to 64K Words/Major Frame. Lookup tables may be any power of two in length up to 64K. A generous format memory allows space for over two hundred thousand lookup table entries, which may be either dedicated or shared among the outputs. The word selector can also be programmed to concatenate segmented parameters together prior to referencing lookup tables. For quick test setups, the word selector can send scaled raw data from the input directly to the outputs without EU conversion. Individual bits can be selected to simulate latched event outputs. A scheme of logical and physical channel numbers permits up to four cards to be daisy-chained for 128-channel capability, with all the cards programmed identically.

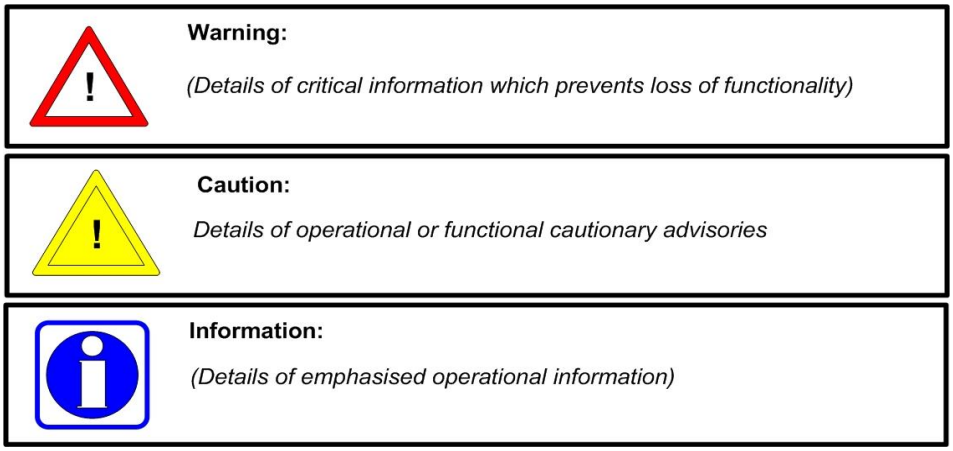
The analog output voltage range can be globally set to ±10V, 0..10V, ±5V, 0..5V, or ±2.5V.

1.2 Manual Format and Conventions

This manual contains the following sections:

- Chapter 1 provides a brief product overview and technical specifications
- Chapter 2 provides installation and configuration instructions
- Chapter 3 provides info on the LS-71 LDPS software
- Chapter 4 provides programming information

Throughout this document, several document flags will be utilized to emphasis warnings or other important data. These flags come in three different formats: Warnings, Cautions, and Information. Examples of these flags appear below.



1.3 Components and Part Numbers for the LS-71 Family

The Lumistar LS-71 1Analog Output card is factory configured to support either 16, or 32 output channels coming from one, or two separate decommutators. An optional signal breakout box assembly is also available from the factory. Use the item part numbers from the table below when ordering.

Table 1-1 LS-71 Family - Components & Part Numbers

Part Number	Description
LS-71-P16	16-Channel Card. Supports one decom
LS-71-P16D	16-Channel Card. Supports two decoms
LS-71-P32	32-Channel Card. Supports one decom
LS-71-P32D	32-Channel Card. Supports two decoms
LS-71-PA16	16-Channel Breakout Box Assembly
LS-71-PA32	32-Channel Breakout Box Assembly

1.4 Software Support

For Microsoft *Windows 98/NT/2000/XP* users, Lumistar provides an Application Program Interface (API) for the Analog Output card in the form of a Dynamic Link Library (DLL.) Lumistar also provides the Lumistar Data Processing System (LDPS) software described in paragraph 3 starting on page 12. The LDPS is composed of two major application programs - the Server and the Client. The Server program is used to setup and acquire data from various sources. The server formats the data into a normalized format, archives it, and passes the data on to the client application for further processing and/or display.

1.5 Specifications

Table 1-2 Specifications	
Inputs	a) Direct Processor Write to individual channels. b) Lumistar Decommutator Systems Parallel Output. c) Option: Up to three more Decommutator Outputs.
Major Frame Length	65,536 words.
Outputs (Daughtercard)	16 Output Channels.
Outputs	16 Output Channels. Option: 32 Channels 68-pin Output Connector
Output Range	Programmable 20V, 10V, 5V span. Zero-scale output –10V, -5V, -2.5V, 0V
Output Impedance	50Ω nominal
Resolution	0.000122V all output ranges
Relative Channel Accuracy	1%
Linearity	+/- 1 LSB
Update Rate	Per Output Channel: 750ksps Per Input Port: Raw 2.5MW. Processed 1.5MW Per Card: 5MW
Reset	System reset sets all outputs to 0V. Software reset sets all outputs to midscale.
Processing	a) Scaled Raw Data Output. b) EU conversion by table lookup. Lookup tables are decoupled from output channels and may be any size 2^n where 1 < n < 16 not to exceed 211,968 total table entries.
Calibration	System software controlled n-step sequence. (n < 4096.)
Power Dissipation	TBD
Form Factor	Full-length PCI slot.
Operating Temperature	0° to 50° C

2 Installation

2.1 Addressing

The LS-71 16/32-Channel Analog Output PCI card occupies both PCI I/O space and memory space. ***No address switch is used; the address is determined by the system.*** The LS-71 card occupies128 bytes of I/O space. The LS-71 will respond to any access in its I/O space. Accesses to the first byte will return an ASCII identifier string.

The amount of memory space taken up by the LS-71 depends on its memory addressing mode. DAC cards are normally shipped in a "flat" addressing mode, where four bits of a bankswitch register map the on-board memory into a 128Kbyte bus window in PCI memory space. The configuration can be changed to activate three additional low-order bits of the bankswitch register to place the bus window in 16Kbytes of MS-DOS real memory space. The user's system may not allow this mode, but Lumistar uses it for testing purposes.

2.2 Physical Installation

The LS-71 PCI card can be installed in any physical slot where it fits. Remove and discard the blanking plate from the chosen slot and carefully insert the card.

2.3 Indicators

The LS-71 card has four chip LED indicators along its top edge. These are board ID indicators, connected to a static register and are intended for use by device drivers in environments where multiple LS-71 cards are present to identify which card is which.

2.4 Interface

2.4.1 Analog Output

The LS-71 card uses a 68-pin connector for analog output. Pin assignments are shown in Table 2–1. The optional LS-71-PA signal breakout box shown in Figure 2-1 on page 10 may be used to access individual analog channels. The breakout box assembly may be mounted in a standard 19-inch wide panel and comes with an interconnecting cable for connector J1 (68-pin).

	**Under no circumstances should the user plug a SCSI device into this 68-pin connector!**

Table 2-1 J1 I/O Connector Pinout			
Pin	**Signal**	**Pin**	**Signal**
1	Analog Output 0	35	Analog Output 16
2	Ground	36	Ground
3	Analog Output 1	37	Analog Output 17
4	Ground	38	Ground
5	Analog Output 2	39	Analog Output 18
6	Ground	40	Ground
7	Analog Output 3	41	Analog Output 19
8	Ground	42	Ground
9	Analog Output 4	43	Analog Output 20
10	Ground	44	Ground
11	Analog Output 5	45	Analog Output 21
12	Ground	46	Ground
13	Analog Output 6	47	Analog Output 22
14	Ground	48	Ground
15	Analog Output 7	49	Analog Output 23
16	Ground	50	Ground
17	Analog Output 8	51	Analog Output 24
18	Ground	52	Ground
19	Analog Output 9	53	Analog Output 25
20	Ground	54	Ground
21	Analog Output 10	55	Analog Output 26
22	Ground	56	Ground
23	Analog Output 11	57	Analog Output 27
24	Ground	58	Ground
25	Analog Output 12	59	Analog Output 28
26	Ground	60	Ground
27	Analog Output 13	61	Analog Output 29
28	Ground	62	Ground
29	Analog Output 14	63	Analog Output 30
30	Ground	64	Ground
31	Analog Output 15	65	Analog Output 31
32	Ground	66	Ground
33	Ground	67	Ground
34	Ground	68	Ground

Figure 2-1 Optional LS-71-PA16/32 Signal Break Out Panel

2.4.2 Word Selector Input

The LS-71 has one word selector input J3. J3 is a 40-pin 0.1" pitch dual-inline header. The header has latching ejectors intended for insulation-displacement cable connectors sans strain relief. The pinout and signaling is compatible with the J2 "Over-the-Top-Bus" (OTB) parallel output connector on Lumistar PCM decommutator systems and shown in Table 2–3. It is possible to connect up to four LS-71 cards, daisy-chaining through their J3 connectors.

Pin 1 of the Lumistar decom output connector is located at the end with the "J2" designator. For earlier decoms, the cable between decommutator and DAC will need a half-twist!

For more complex systems where the LS-71 card is to also be driven from a second PCM decommutator, an optional additional word selector is available. This second word selector input is J5, and has the same pinout as J3.

Table 2-2 Word Selector Input Pinout

Pin	Signal	Pin	Signal
1	Ground	2	Ground
3	OD1	4	OD9
5	OD2	6	OD10
7	OD3	8	OD11
9	OD4	10	OD12
11	OD5	12	OD13
13	OD6	14	OD14
15	OD7	16	OD15
17	OD8	18	OD16
19	Ground	20	Ground
21	WdStb	22	Ground
23	FrmStb	24	Ground
25	MFStb	26	Ground
27	Clock	28	Ground
29	1stBIT	30	Ground
31	Not Used	32	Ground
33	Not Used	34	Ground
35	Ground	36	Ground
37	Ground	38	Ground
39	Ground	40	Ground

3 Operation of the LS-71 With The LDPS Software

The Lumistar LS-71 16/32-Channel Analog Output Card can be set up and controlled by using the Lumistar Data Processing System (LDPS) software (shown below left).

Figure 3-1 LDPS Setup Display for the LS-71

The LDPS is composed of two major application programs - the Server and the Client. The Server program is used to setup and acquire data from various sources. The server formats the data into a normalized format, archives it, and then pass the data on to the client application for further processing and/or display. The Client is mainly a data processing and presentation program, with hooks to allow new display and processing routines to be added by the user. The server and client applications can run together on the same computing platform, or on different platforms interconnected via a Local Area Network (LAN), Wide Area Network (WAN), or the Internet. This user's manual will focus primarily on the server side application.

The LS-71 is designed to be used in conjunction with the Lumistar LS-50-P Multifunction PCM Decommutator board and uses an over-the-top communication bus directly from the decommutator to the LS-71 to avoid any glitches in the analog outputs due to PCI bus activity. The LS-71 design allows two LS-50-P Multifunction boards to be used with a single LS-71 board allowing tagged data from either data stream to be directed to any LS-71 channel.

To initially configure the LS-71, perform the following steps:

1. Run the LDPS server program and from the System menu shown below, select "Devices" and then "Manage" *(System→ Devices→ Manage)*
2. From the System Manager shown below left, select the "Enable" check box next to the LS71 button. The "Ls71_8x" button will then become active (not grayed out). Note the red rectangle around the button - this indicates that the application has not yet started. Note also the "Sim" check box next to the "Enable" check box. Checking this box allows the LDPS application to operate when a LS-71 board is not installed in the system.
3. From the System Manager, click the "Ls71_8x" button. This will launch the "Ls71_8x (DAC)" display shown below right. Note that the red rectangle around the button has changed to green indicating that the application is now running.
4. To setup and configure the LS-71 card(s), follow the procedures outlined in paragraphs 3.1.

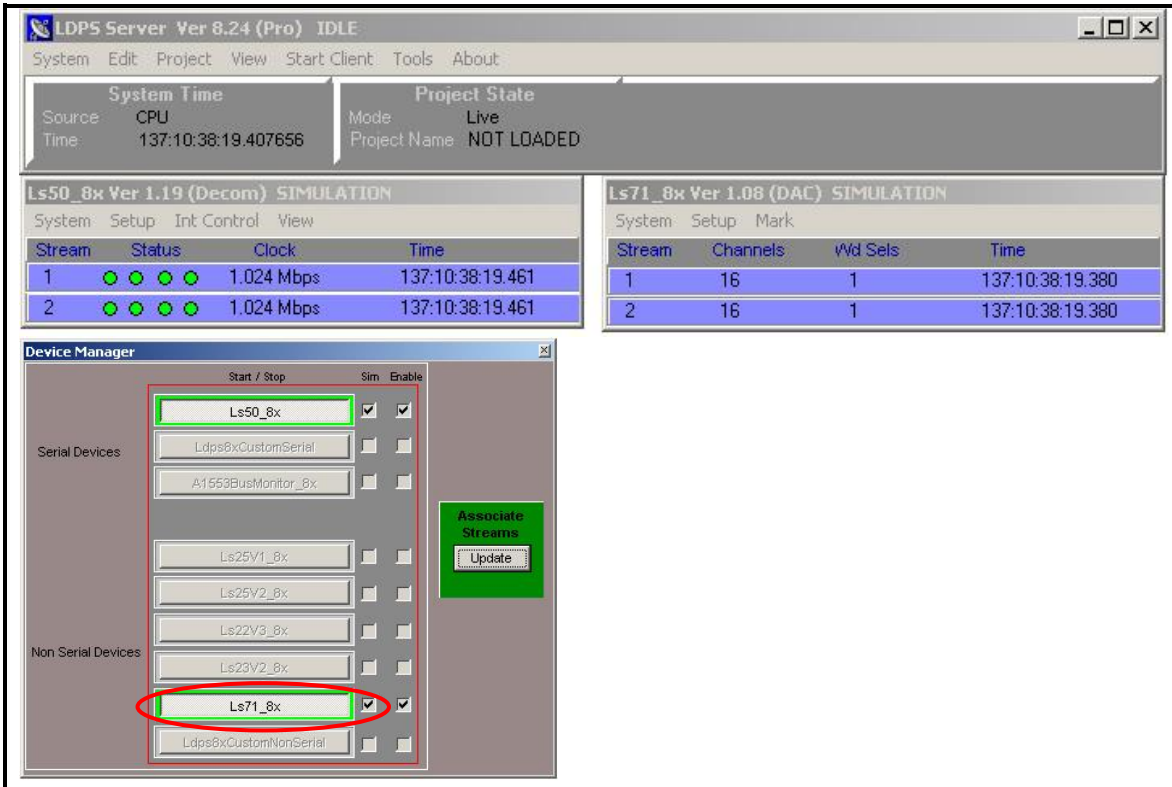


Figure 3-2 LDPS Server Application Windows

3.1 Configuring The LS-71 Hardware

From the "Ls71_8x (DAC)" display shown below center in Figure 3-3, click "Setup" and then "Stream 1" *(Setup →Stream 1).* The resulting "DAC Card Number X Setup" display shown below left in Figure 3-3 is divided into several regions. To invoke the controls for this display, simply select a particular menu item, or place the mouse curser in the region and right click. The resulting menus are shown in Figure 3-4 on page 15 are discussed in detail in the following paragraphs.

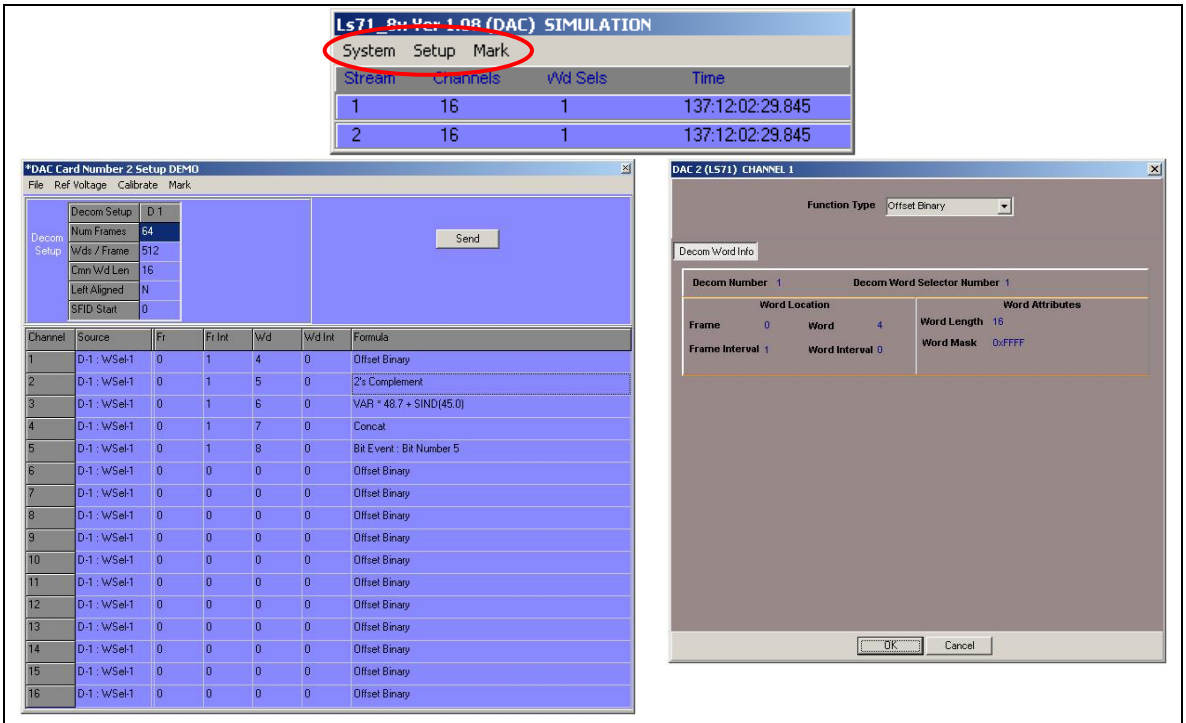


Figure 3-3 LDPS Configuration Menus/Controls for the LS-71

The "Ls71_8x (DAC)" display shown above center in Figure 3-3 has three menu items: "System", "Setup", and "Mark" (as indicated by red oval). For each LS-71 card installed, the number of channels per/card has will also be displayed. In the example discussed here, two 16-channel LS-71 cards are installed.

Selecting the "Mark" item will deflect the output to 0 scale, then ¾ scale, then 0 scale for each channel on all LS-71 cards installed.

To associate each stream in the system with the corresponding hardware, click "System" and then "Flash Board ID Leds" *(System → Flash Board ID Leds).* Select the stream number from the drop-down list and the board ID LEDs of the corresponding LS-71 card will begin to flash several times.

To setup the LS-71 hardware for each stream, click "Setup" and then "Stream 1" *(Setup →Stream 1).* The resulting "DAC Card Number X Setup" display is shown above left in Figure 3-3.

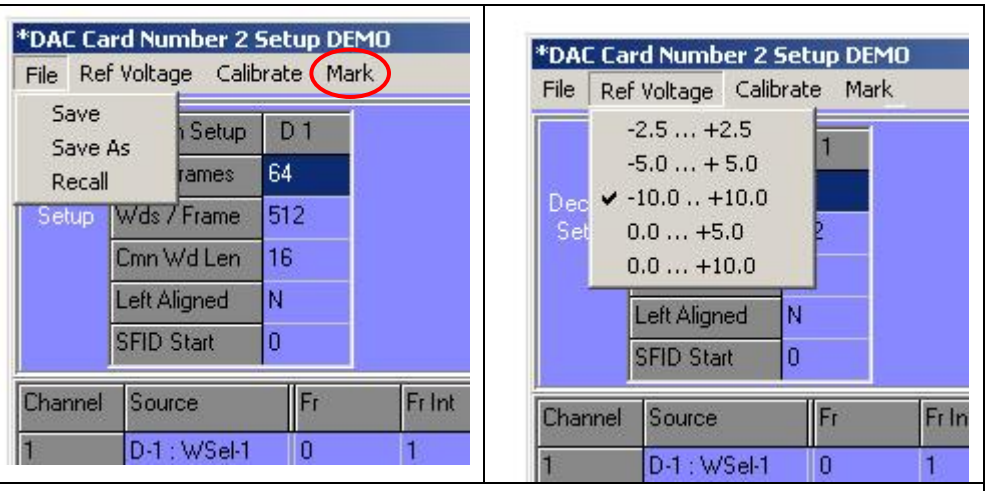


Figure 3-4 The LS-71 Configuration Menus

The LS-71 hardware configuration display shown above in Figure 3-4 has four menu items: "File," "Ref Voltage," "Calibrate" and "Mark." Both "File" and "Ref Voltage" have sub-menus, the others don't. All of the LS-71 hardware configuration functions are described in more detail in the following paragraphs.

3.1.1 File Menu

The File Menu options are shown in Figure 3-5 on page 16. The File Menu allows the user to "Recall" previously defined LS-71 hardware configurations for streams in the system. The configuration(s) may be used as is, or the user may change and "Save" the configuration changes to the current file, or "Save As" to a new/different file. After a configuration file has been recalled and/or saved, the name of the file will appear in the title bar of the window as shown in Figure 3-4 above left (red oval). Note the file extension (.LSDAC) for the LS-71 hardware configuration file.

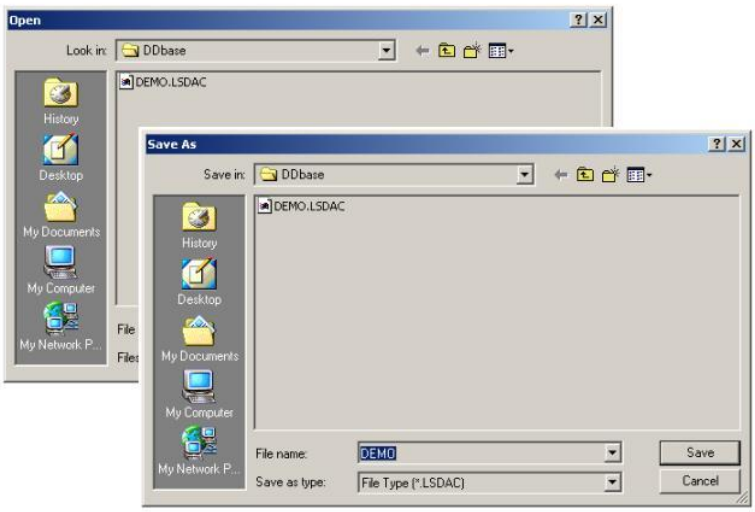


Figure 3-5 File Menu Options

3.1.2 Reference Voltage Menu

The Reference Voltage Manu shown right allows the user to select the output voltage range for all channels of a specific LS-71 card. Both unipolar and bipolar output voltage ranges are available. The available ranges listed below include:

1. –2.5 Vdc to + 2.5 Vdc
2. 5.0 Vdc to + 5.0 Vdc
3. –10.0 Vdc to + 10.0 Vdc
4. 0 Vdc to + 5.0 Vdc
5. 0 Vdc to + 10.0 Vdc

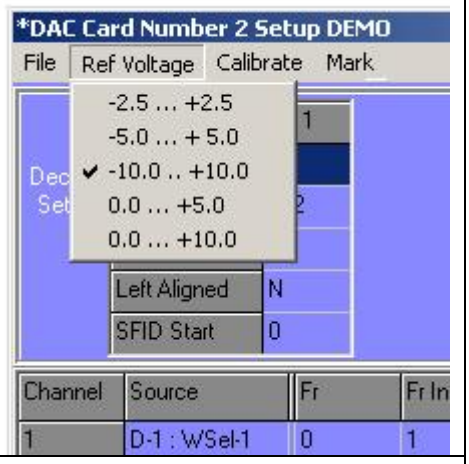


Figure 3-6 Reference Voltage Menu

3.1.3 Calibrate Function

The Calibrate function is used to drive all the channels on a particular LS-71 card to certain positions. This is done to align and/or adjust the behavior of strip chart recorders or other such devices used to monitor the analog signals produced by the LS-71. Several permutations of the "DAC CARD X CALIBRATION" window are shown below in Figure 3-7. The particular window displayed will be a function of the type of Deflection mode selected.

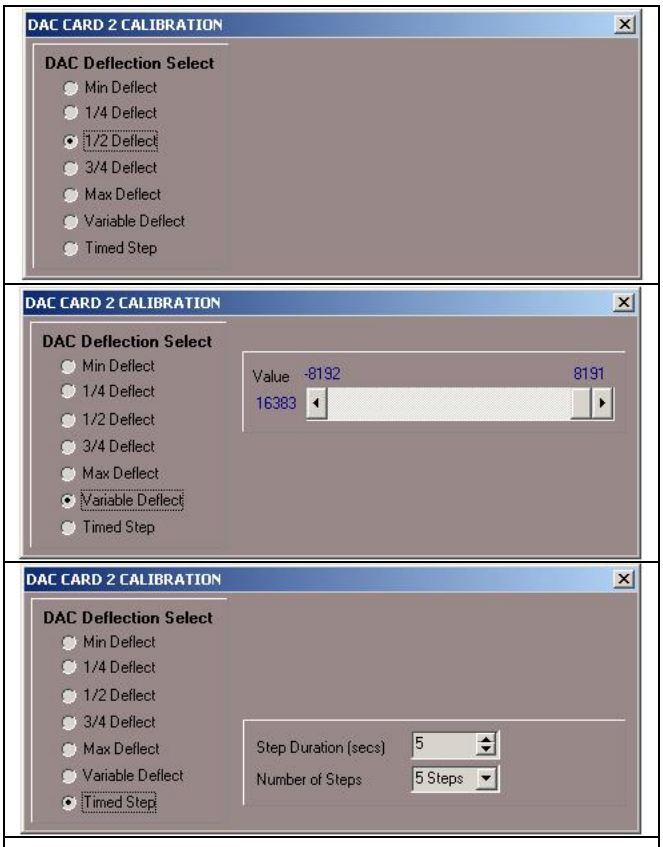


Figure 3-7 Calibrate Menus

As shown above, the user may select one of the following deflection modes:

1. Min Deflect – Drives all outputs to 0, -10, or –5 volts, the minimum output of the LS-71 as specified by the Ref Voltage (see paragraph 3.1.2 on page 16).
2. ¼ Deflect – Drives all outputs to one fourth of the maximum output of the LS-71.
3. ½ Deflect - Drives all outputs to one half of the maximum output of the LS-71as specified by the Ref Voltage.
4. ¾ Deflect - Drives all outputs to three fourths of the maximum output of the LS-71as specified by the Ref Voltage.
5. Max Deflect - Drives all outputs to the maximum output of the LS-71as specified by the Ref Voltage.

6. Variable Deflect – Displays a slider bar that allows the user to control the output to any range between minimum deflection and maximum deflection (as specified by the Ref Voltage).
7. Timed Step – Allows the user to set the duration in seconds and the number of steps to deflect the channels from minimum deflection to maximum deflection.

3.1.4 Mark Control

Selecting the "Mark" item will deflect the output to 0 scale, then ¾ scale, then 0 scale for each channel on all LS-71 cards installed.

3.1.5 Decom Setup

Below the menus in the LS-71 hardware configuration display shown below left in Figure 3-8 is the decommutator information and setup area. This area displays the decom setup for a particular stream number. The user may individually enter/modify each of the parameters shown in the display and save them permanently using the File command (see paragraph 3.1.1 on page 15). The user may also recall a previously defined LS-50 decom setup file by double clicking in decom setup area or right clicking and selecting the "Set To Decom Setup" button. The resulting file dialog box is shown below right in Figure 3-8 (red oval). Note the file extension (.B50) for the LS-50 decommutator.

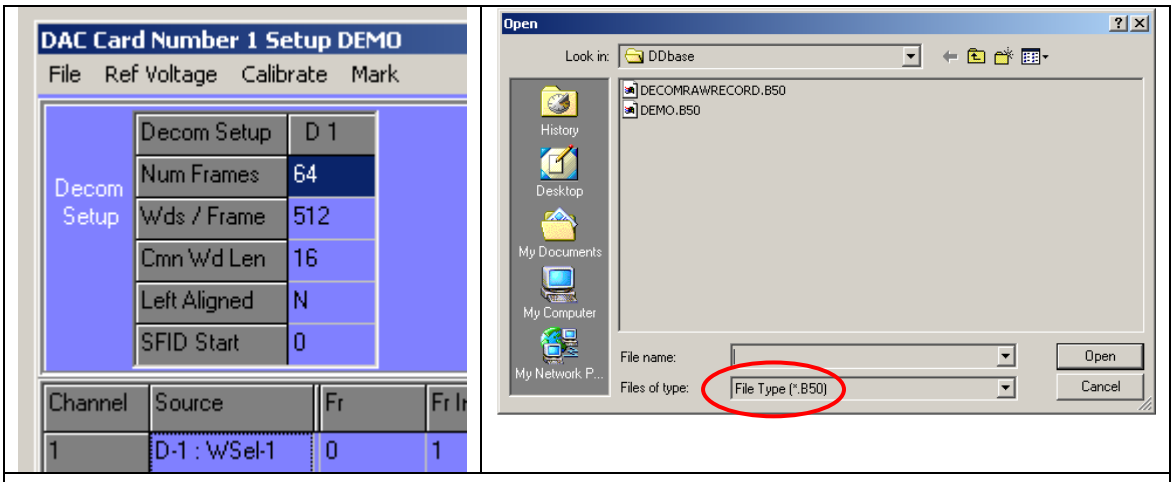


Figure 3-8 Decom Setup

The parameters displayed in the decom setup area include:

A. Num Frames – Set to the number of minor frames per major frame.
B. Words Per Frame – Set to the number of words per minor frame.
C. Common Word Length – Set to the number of bits in most of the words for the decommutator stream.

D. Left Aligned – Indicate (Y/N) here if the output of the decommutator is set for left aligned data
E. SFID Start – The SFID counts from 0 or 1

Figure 3-9 LS-71 Setup Dialog Box

3.1.5.1 Channel Setup

The grid on the lower portion of the LS-71 setup dialog box shown in Figure 3-9 above displays the information the user sets up for each channel on the LS-71 card. The LS-71 channel numbers are listed in the left column. The six parameters along the top of the grid have the following meaning:

a) Source – The source of the data feeding the channel. This is the data coming from the decommutator word(s) and the word selector.

b) Fr – The minor frame number of the word driving the LS-71, as configured by the decommutator.
c) Fr Int – The minor frame interval of the word driving the LS-71, as configured by the decommutator.
d) Wd – The word number of the word driving the LS-71, as configured by the decommutator.
e) Wd Int – The word interval of the word driving the LS-71, as configured by the decommutator.
f) Formula – This is the formula, or function type applied to the LS-71.

3.1.5.2 Edit Channel Dialog Box

To set up each channel on the LS-71 card with the required configuration, select the channel number on the Setup Display, and double click, or right click on the channel number and click the "Edit Channel" button that results. Either way, a new channel setup window will be displayed as shown in Figure 3-10 on page 21. The Edit Channel dialog box is where the user specifies the source of the LS-71 data, the function to be applied to the channel, the decommutator word selection(s), any masks to be applied, and any formulas to be applied.

NOTE: For all channels on a LS-71 card, a decommutator word can only be used once, for all channels. They cannot be repeated. For instance, the user cannot use Word 1 of Frame 3 in channel 2 and for channel 7.

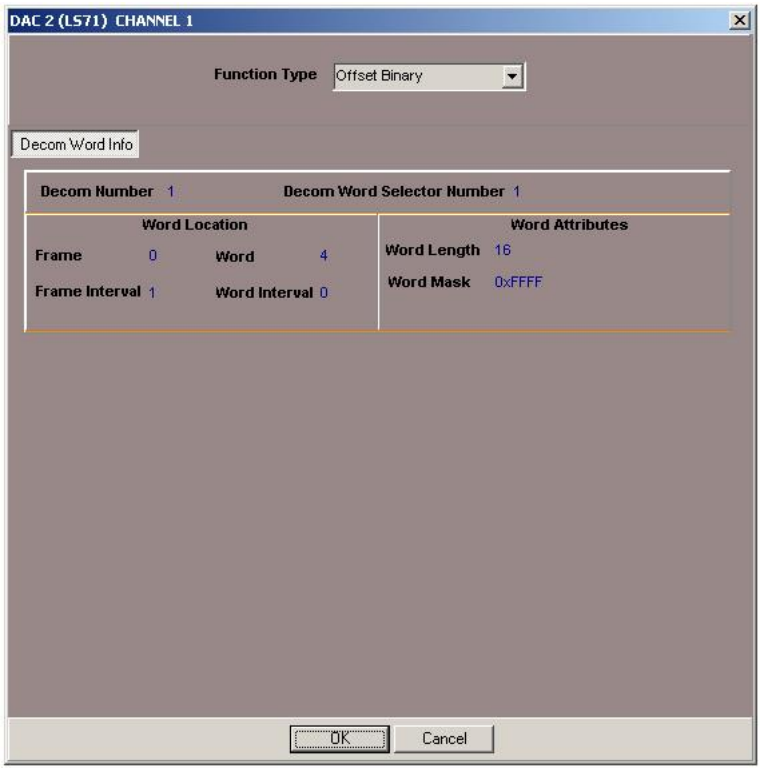


Figure 3-10 Edit Channel Dialog Box

The Edit Channel Dialog Box elements of Figure 3-10 above are as follows:

Function Type – The selected channel can have the data output from the decommutator adjusted by several methods (see paragraph 3.1.5.2.1 on page 22)

Decommutator Word Information – The user selects a word location from the decommutator that the function type described above will be applied to. For the Concatenate function type (see paragraph 3.1.5.2.3 on page 23), the user selects two decommutator words. The user will also select the Word Selector number that the data coming into the LS-71 will use. The specific decommutator word information that the user will specify are as follows:

a) Frame – The minor frame number of the selected decommutator word, as defined in the decommutator setup.
b) Frame Interval – The minor frame interval of the selected decommutator word, as defined in the decommutator setup.
c) Word – The word number of the selected decommutator word, as defined in the decommutator setup.

d) Word Interval – The word interval of the selected decommutator word, as defined in the decommutator setup.
e) Word Length – The number of bits in the selected decommutator word, as defined in the decommutator setup.
f) Word Mask – The word mask is applied to the selected decommutator word and is entered as a hex value. Also if a math function will use a bit in the decommutator word, the user should ensure that the bit is also set in the mask. (To use all bits, set the mask to 0xFFFF)

3.1.5.2.1 Word Function Types

The specified channel can have the data output from the decommutator adjusted by several methods and expressed in several formats as follows:

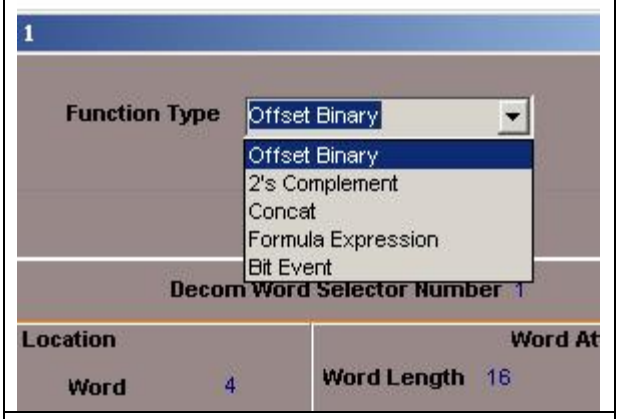


Figure 3-11 Word Function Types

1. Offset Binary – Essentially the raw output data from the decommutator applied to the LS-71 after any defined mask is applied.
2. Twos Compliment – The raw output data from the decommutator is expressed in twos compliment format after any defined mask is applied.
3. Concatenate – Two decommutator words are concatenated as follows: ((Word1 & Mask1) << (bits in Word2)) + (Word2 & Mask2)
4. Formula – An extremely powerful capability where the user selects a specific word from the decommutator and applies a formula using the math engine (see paragraph 3.1.5.2.4 on page 24). The formula is applied to the decommutator output prior to going to the LS-71.
5. Bit Event – This is a simple method to deflect the channel to ¾ scale if a specific bit is set to TRUE (1).

3.1.5.2.2 Offset Binary & 2's Complement Word Formats

Offset Binary Representation is a method for representing signed numbers and is popular in A/D and D/A conversions. To represent a binary number as offset binary, begin calculating by assigning half of the largest possible number as the zero value. A positive integer then is represented as the absolute value of the integer added to the zero number. A negative integer is represented as the absolute value of the integer subtracted from the zero number.

For example:

Largest value for 8-bit integer = $2^8 = 256$
Offset binary zero value = 256÷2 =128(decimal) =1000 0000(binary)

Signed Integer	Offset Binary
+5	1000 0101
+4	1000 0100
+3	1000 0011
+2	1000 0010
+1	1000 0001
0	1000 0000
-1	0111 1111
-2	0111 1110
-3	0111 1101
-4	0111 1100
-5	0111 1011

The Two's complement representation allows the use of binary arithmetic operations on signed integers, yielding the correct 2's complement results. Positive 2's complement numbers are represented as the simple binary. Negative 2's complement numbers are represented as the binary number that when added to a positive number of the same magnitude equals zero. Note: The most significant (leftmost) bit indicates the sign of the integer; therefore it is sometimes called the sign bit.

Integer		2's Complement
Signed	Unsigned	
5	5	0000 0101
4	4	0000 0100
3	3	0000 0011
2	2	0000 0010
1	1	0000 0001
0	0	0000 0000
-1	255	1111 1111
-2	254	1111 1110
-3	253	1111 1101
-4	252	1111 1100
-5	251	1111 1011

To calculate the 2's complement of an integer, invert the binary equivalent of the number by changing all of the ones to zeroes and all of the zeroes to ones (also called 1's complement), and then add one. If the sign bit is zero, then the number is greater than or equal to zero, or positive. If the sign bit is one, then the number is less than zero, or negative.

3.1.5.2.3 Concatenate Function

The Concatenate Function dialog box is shown in Figure 3-12 on page 24. The resulting concatenation stores the result in the MSWord or LSWord selected by the user and is then output. The two decommutator words are concatenated as follows:

((Word1 & Mask1) << (bits in Word2)) + (Word2 & Mask2)

Note: "<<" in the above expression denotes left shift N number of bits, where N=word length.

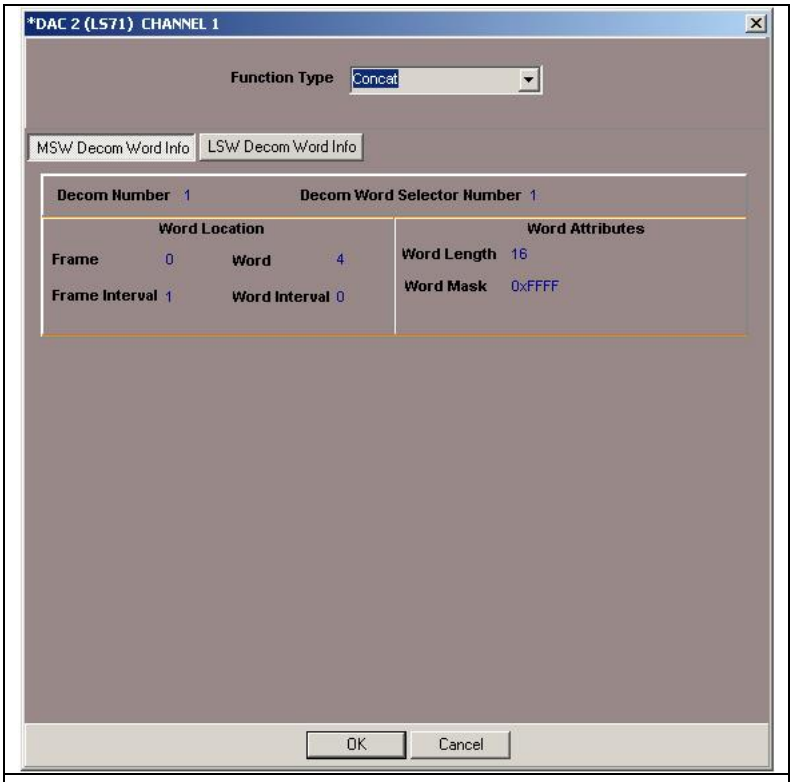


Figure 3-12 Concatenate Dialog Box

3.1.5.2.4 Formula Expression Dialog Box

The Formula Expression dialog box is shown in Figure 3-13 on page 25. In addition to the decommutator information (see paragraph 3.1.5.2 on page 20), the user also enters the formula that will be applied to the selected decommutator word before it is output. There is a help window on the bottom to assist the user in entering the formula. In this example, the formula entered is indicated by the red oval.

In specifying the formula, the decommutator word becomes the variable. The variable name is "VAR." For example, to enter a simple mx+b linear equation, the user would type "VAR * 2.0 + 0.777," where 2.0 is m, VAR is x, and 0.777 is b. The math engine will also allow the user to enter conditional statements.

To verify the syntax of the formula, click the Syntax Check button (yellow oval in the figure). This will check to ensure that parenthesis and other syntax rules are followed. If the formula is valid, then NO ERROR will be displayed next to the button. If the syntax is invalid, then the error message will be displayed next to the button and the cursor will be placed in the position in the formula the error was encountered.

For certain calculated parameters, the resultants are often compared to minimum and maximum values of the parameter. For example if the decommutator word represents a pitch

angle, the user might specify –90.0 degrees for the minimum value. The result of the calculation will produce a minimum deflection on the LS-71 channel when –90.0 or less is solved.

To enter the Min and Max Scaled values, place the mouse cursor next to the labels and right click. Select the "Minimum Scaled Value" or "Maximum Scaled Value" and enter the values in the input box. Note, that the Min and Max Scaled values may not be entered if the "Unity Gain" check box is selected.

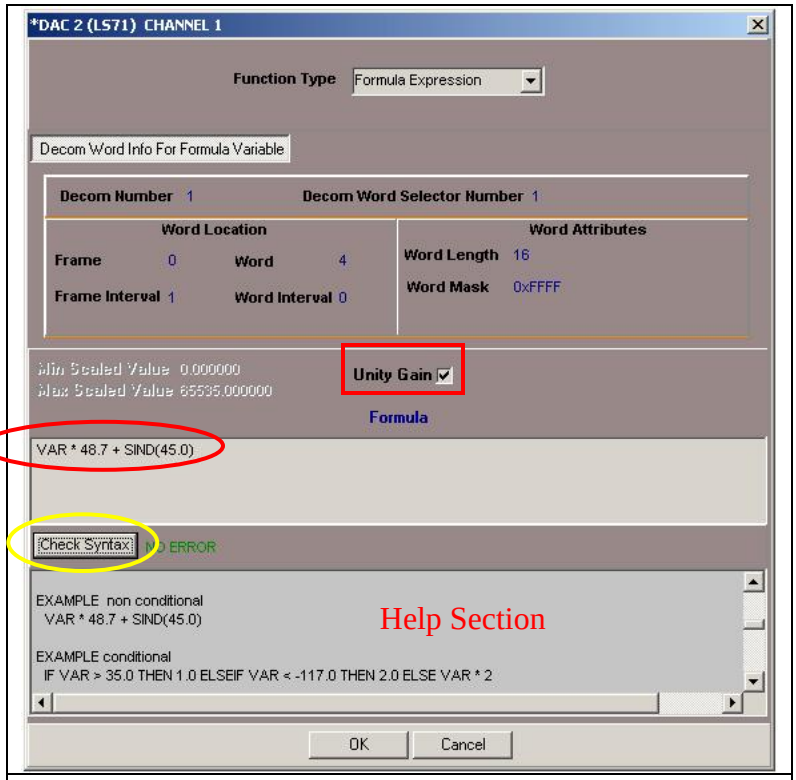


Figure 3-13 Formula Expression Dialog Box

If the user wishes to normalize the results of the calculation, then the "Unity Gain" check box should be selected (red rectangle in the figure above). When selected, the application will solve for the Max Scaled Value and the Min Scaled Value. In this scenario, the Max Scaled Value will become $(2^{numbits}) - 1$, and the Min Scaled Value will become 0.0.

The processing of the channel prior to output to the LS-71 using this function type can be summarized as follows:

1. The selected decommutator word is collected from the decommutator.
2. The selected decommutator word is logically "anded" with the mask bits specified by the user.

3. Prior to the load process of the LS-71, the formula entered by the user is pre-solved by the math engine for every possible bit combination of the raw decommutator word (which is the word length of the word, or 14 bits, which ever is less). The individual results are populated in the look-up table during the load process and during operation, the value derived in step #2 above will drive the look-up table.
4. The result of step #3 above is multiplied by the scale factor determined by the result of Max Scaled Value – Min Scaled Value.
5. The result is output to the LS-71 channel.

3.1.5.2.5 Bit Event Dialog Box

The Bit Event dialog box is shown in Figure 3-14 below. When the Bit Event function type is selected, the Event Bit field is added to the Word Attributes area as indicated below via the red oval. The Bit Event function is a simple method to deflect the channel to ¾ scale if a specific bit is set to TRUE (1). To select the bit number that will trigger the event, right click in the Word Attributes area and select Event Bit from the list. Enter the bit position in the resulting dialog box and click OK.

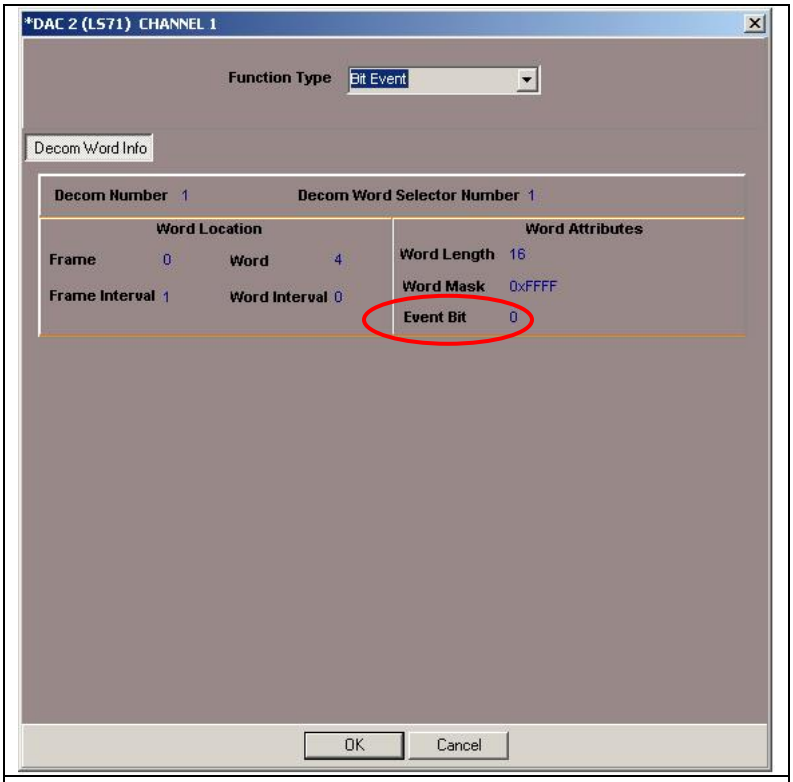


Figure 3-14 Bit Event Dialog Box

4 Programming Information

4.1 Introduction

This chapter is targeted to authors of device drivers, API's, and telemetry applications who will need to know what all the bits do.

The LS-71 is controlled by an array of eight-bit registers, each identified by a register number. This was done to ease moving the implementation across different form factors and addressing schemes.

4.2 Locating a PCI Device

PCI components do not have fixed address assignments. At system startup a power-on routine scans the computer for PCI interfaces and assigns system resources such as address space to them.

On non-PC architectures the user may run into Big/Little-Endian issues. Be mindful of this while troubleshooting.

Each PCI component is assigned an array of sixty-four 32-bit registers in what is referred to as configuration space. This area is normally not accessible anywhere in system address space and must be accessed by special means that are system-dependent.

To locate a LS-71 card in your system, perform the following:

1. Initialize an "*index*" value to zero. This index is allowed to grow as large as 255 by the PCI specification, but in practice never get this large.

2. To locate PCI9080 chips, set machine registers:

   AX = 0xB102
   CX = 0x9080
   DX = 0x10B5
   SI = *index*

3. Issue a software interrupt 0x1A. If the system returns from interrupt with the carry flag set, any such devices are already located and no (more) exist. Skip out of the scanning routine. If the carry flag is clear, the BIOS call will have returned a "*handle*" in BX.

4. If the carry flag was clear, read the sub-identifier. Set registers:

AX = 0xB10A
BX = *handle*
SI = 0x2C

5. Issue another software interrupt 0x1A. The interrupt returns a value in ECX. If the value returned is 0xDAC0B00B, the handle points to a LS-71 card and other configuration registers may be accessed to obtain the base addresses. (Otherwise skip to step 7.) Set registers as shown below. Register numbers are:

   Register 0x10 – PLX9080 Runtime Registers Memory Address.
   Register 0x14 – PLX9080 Runtime Registers I/O Address.
   Register 0x18 – PCI DAC Memory Window Address.
   Register 0x1C –PCI DAC Register Address.
   Register 0x3C – PCI DAC IRQ Number (for DMA only.)

   AX = 0xB10A
   BX = *handle*
   SI = *register number*

6. Issue another software interrupt 0x1A. The value returned in ECX is the register value. When reading the IRQ Number register, only the eight LSBs are important. They are the IRQ ("8259") number assigned to the PCI interrupt. If these bits are 0xFF, the system was unable to assign an interrupt for some reason. The LS-71 card generates no interrupts of its own; this value has meaning only if one choose to use the DMA termination interrupt from the PLX9080 DMA controller while loading setups into memory.

   When reading addresses, logically AND the value returned in ECX with 0xFFFFFFF0. This yields the base address. If the LSB of ECX was a zero, the address is in memory space. If the bit was a one, the address is in I/O space. Reload AX, BX, and SI and repeat the call to obtain the necessary addresses. The PLX9080 runtime registers may be accessed via memory or I/O operations at your convenience. Skip out of this routine when all of them have been read.

	Microsoft operating environments are notorious for erasing the configuration registers of some hardware. If the locating procedure described here places the memory address at zero, this is most likely the cause.

7. Increment the index value and try again.

The LS-71 may be configured to place the buffer memory in protected memory space ("flat mode") or in real space ("page mode.") In flat mode the memory occupies 128 Kbytes of

contiguous address space and the PAGE field of the Bankswitch register is ignored. If the LS-71 is in page mode the buffer memory occupies 16 Kbytes of address space and three high-order on-board address bits are supplied by the Bankswitch register.

The PCI DAC card occupies 128 bytes of I/O space.

4.3 Register Summaries

Registers appear at the I/O address obtained by adding the register number to the I/O register address. Register bit assignments are summarized in the following tables and discussed in detail later on in this chapter. In many cases read and write bit assignments for the same register are different. Bits defined with a – dash are meaningless.

All register numbers (#) are hexadecimal.

Table 4-1 and Table 4-2 are memory aids for the programmer. Many bit names have been shortened for typographical purposes and have different (longer) mnemonics elsewhere in this narrative.

Table 4-1 Write Register Summary

Register	#	7	6	5	4	3	2	1	0
Board ID	00	–	–	–	–	Board Number			
DAC Control	02	Dac32	DACNUM			OFFSET		SCALE	
WordSel Control	04	–	B-End	DEPTH		–	–	–	RUN
Bankswitch	06	OVRD	WSSEL		BANK		PAGE		
DAC 0 Data	20	Direct Write DAC Data LSBs							
	21	–	–	Direct Write DAC Data MSBs					
DAC n Data (through)	22 5F	ala 20..21. Register # = 0x20 + 2x DAC #							

Table 4-2 Read Register Summary

Register	#	7	6	5	4	3	2	1	0
Identifier	00	0	"LSDAC"						
Configuration	02	NDACS		–	–	Wsel3	Wsel2	Wsel1	Wsel0

4.4 General

The LS-71 has two functional groups. One group is one or more word selectors. Each word selector connects to a parallel data source (i. e., a PCM decommutator.) The word selector uses table lookup techniques to choose certain words from the stream of parallel data. The word selector then manipulates the values of these words using more table lookup techniques and outputs them in turn.

The output of the word selector are 14-bit unipolar words, each of which has an attached 7-bit logical DAC number.

The other group on the card consists of the DAC controller and the DACs themselves. The DAC controller reads the output of the word selector(s), scales and offsets the data values to the desired output range, and steers the data to a DAC if the logical DAC number points to a physical DAC present on the board.

4.5 Registers

Registers are all 8-bit at even I/O port numbers except as noted.

4.5.1 Board ID Register

The board ID register is meaningful only for PCI cards. This register setting has no effect on the operation of the card. It controls only the state of the board number indicators. In desktop PC implementations, if there are multiple instances of the same physical device, there is no way from BIOS information to tell which is which. This user may use this register as needed.

4.5.2 Identifier Register

When read repeatedly, this register returns a null-bounded ASCII string "**`LSDAC`**" to identify the board.

4.5.3 DAC Control Register

The DAC control register sets the operating range of the outputs and sets the DAC controller logical address. This register is global for all D/A's on card.

	Whenever the user writes to the DAC Control Register, all analog outputs are driven to mid-scale until updated later.

Table 4-3 DAC Control Register		
Bit	**Mnemonic**	**Description**
0..1	SCALE	Control D/A output voltage swing: 00 14-bit D/A's dynamic range 20V. 01 13-bit D/A's dynamic range 10V. 1x 12-bit D/A's dynamic range 5v.
2..3	OFFSET	Sets zero-scale output voltage: 00 Zero-scale D/A output -10V. 01 Zero-scale D/A output 0V. 10 Zero-scale D/A output -5V. 11 Zero-scale D/A output -2.5V.
4..6	DACNUM	The D/A controller expects the word selector to provide a logical D/A channel number in the range 0..127. The four LSBs of logical and physical D/A channels are always equal. An update happens only if the MSBs of the logical D/A number match the DACNUM bits. Otherwise the data is discarded. This permits identical programming of word selectors when a data source is connected to more than one D/A card.
7	DACNUM32	If more than 16 physical D/A channels, set this bit. It causes bit 4 of the DACNUM field to be ignored and bit 4 of the logical channel number becomes bit 4 of the physical number.

SCALE and OFFSET present a variety of nonsense options. Meaningful values are:

0000	±10V
0101	0—10V
0110	0—5V
1001	±5V
1110	±2.5V

Note: Setting the DACNUM option can be complex. For simplicity, set bits 4..7 to 1000 and proceed..

Table 4-4 Configuration Register		
Bit	**Mnemonic**	**Description**
0	WSEL0	If this bit returns 1 the card is broken and nothing read or written will have any meaning.
1	WSEL1	Zero if optional second word selector present.
2	WSEL2	Zero if optional third word selector present.
3	WSEL3	Zero if optional fourth word selector present.
4..5		Meaningless.
6..7	NDACS	Hard-coded per number of physical channels present: 00 = 32. 01 = 8. 10 = 16. 11 = 24.

4.5.4 Configuration Register

This register returns the physical card configuration. All the defined bits are hard-coded.

4.5.5 Word Selector Control Register

There is actually a separate control register for each Word Selector present, but they are all at the same address and are always written at once.

Table 4-5 Word Selector Control Register

Bit	Mnemonic	Description
0	RUN	1 to run data. 0 for memory access or calibration. If running, PCI transactions complete, but memory writes are ignored and memory reads return nonsense.
1..3		Meaningless.
4..5	DEPTH	Set per format and data source: 00 All frames are alike. Format has no major frame structure. 01 Frames range 0..255. 10 Nonsense. 11 Frames range 0..1023 (Lumistar decommutators only.).
6	BIGEND	Swaps Tag lookup so even words in high bytes and odd words in low. For use with other (non-iAPX86) machine architectures.
7		Meaningless.

4.5.6 Bankswitch Register

A bankswitch arrangement is needed to provide enough address bits to map the LS-71 on-board memory (up to 2Mbytes may be present) into the bus memory window. Several units of measure for memory size are implied here:

One "Page" is 16Kbytes of memory. This is the amount of memory accessible at a time for a card in Page Mode.

One "Bank" is eight Pages. This is the amount of memory accessible at a time for a card in Flat Mode.

One "WordSelector" is four Banks. The LS-71 PCI card has at least one, and may optionally have up to four.

Table 4-6 Bank-switch Register		
Bit	**Mnemonic**	**Description**
0..2	PAGE	Selects a Page in the selected Bank of the selected WordSelector. This field is meaningless if the card is in Flat mode.
3..4	BANK	Selects a Bank in the selected WordSelector.
5..6	WSSEL	Selects a WordSelector. Accesses are meaningful only if the WordSelector is physically present (its WSEL bit in the configuration register must be 0.)
7	OVERRIDE	Has no effect on memory reads. For memory writes, overrides the WSSEL field so data is written to all word selectors simultaneously. You may set this bit during initialization, and when loading EU lookup tables if you want to share them among multiple data sources.

4.5.7 Low Data Registers

Individual outputs may be set by writing to the Data registers. Writes to even register numbers in the range 0x20..0x5E update a Low Data register. If this is an 8-bit write, 8 LSBs of data to be written directly to a *physical* DAC, and it doesn't matter which address if in range. If this is a 16-bit write, 14 LSBs are used and D/A updates **now**. Register number = twice D/A channel number + 0x20.

4.5.8 High Data Registers

Writes to odd register numbers in the range 0x21..0x5F update a *physical* DAC. The value is the 6 LSBs concatenated to the last Low Data Register value written. Register number = twice D/A channel number + 0x21.

4.6 Memory Map

The Word Selector memory is segmented up as shown in Table 4-7 on page 34. Most of the time the word selector treats this memory as organized into segments of 64K 16-bit WORDS, so it is documented as word-addressed here.

The word, "WORD" in all caps implies 16-bits of data, and when used in conjunction with memory, implies a 16-bit address.

Addresses in the table are WORD ranges so when one accesses the memory from the bus one needs to multiply all these addresses by 2.

Table 4-7 Memory Map	
WORD Range	**Definition**
0x00000–0x07FFF	Lookup to convert word# (in *major* frame) to tag#. This part is organized as 64K bytes. Even words in low byte, odd words in high, unless BIGEND set.
0x08000–0x083FF	Lookup to convert word# of first word in Frame [frame] to start (byte!) address in above table so if there is a flitch the card can recover without waiting for the next major frame.
0x08400–0x-BFFF	Accessible, unused. You can put some EU LUTs here.
0x0C000–0x0C03F	Program Space for Tag 0. Program starts at offset 0. By convention, the WORD at offset 0x3F is reserved for SAVE instruction operand address.
0x0C040–0x0FFFF	Program Space for Tags 1..255 organized ala C000-C03F.
0x10000–0x1FFFF	Space for LUTs divided as needed. Tables don't have to be 2^n WORDS long but nothing else makes sense.
0x20000–0x2FFFF	Space for more tables.
0x03000–0x3FFFF	Space for *yet more* tables.

4.7 Philosophy

The block diagram of the LS-71 is shown in Figure 4-1 on page 35. In simple terms, the LS-71 has a FIFO hooked up to the Over The Board (OTB) connector. A bit of logic in one corner of the Word Selector controls the FIFO. A latch is cleared when RUN is turned off. When RUN is set the latch stays cleared until a (minor) FRAME STROBE comes in the OTB connector. That sets the latch. If the latch is set, a WORD STROBE (1-1/2 bits into the word) clocks the OTB DATA into the FIFO.

But at FRAME STROBE time, the FIFO is first clocked an extra time, during the first bit. The significance of this is OTB DATA is the minor frame number during the first bit of each word. The FIFO is seventeen bits wide. Sixteen are connected to OTB DATA. The 17^{th} bit is a flag, set when the frame number is loaded.

The rest of the word selector looks for a non-empty state at this FIFO. If empty, it idles. Otherwise the oldest word is read out.

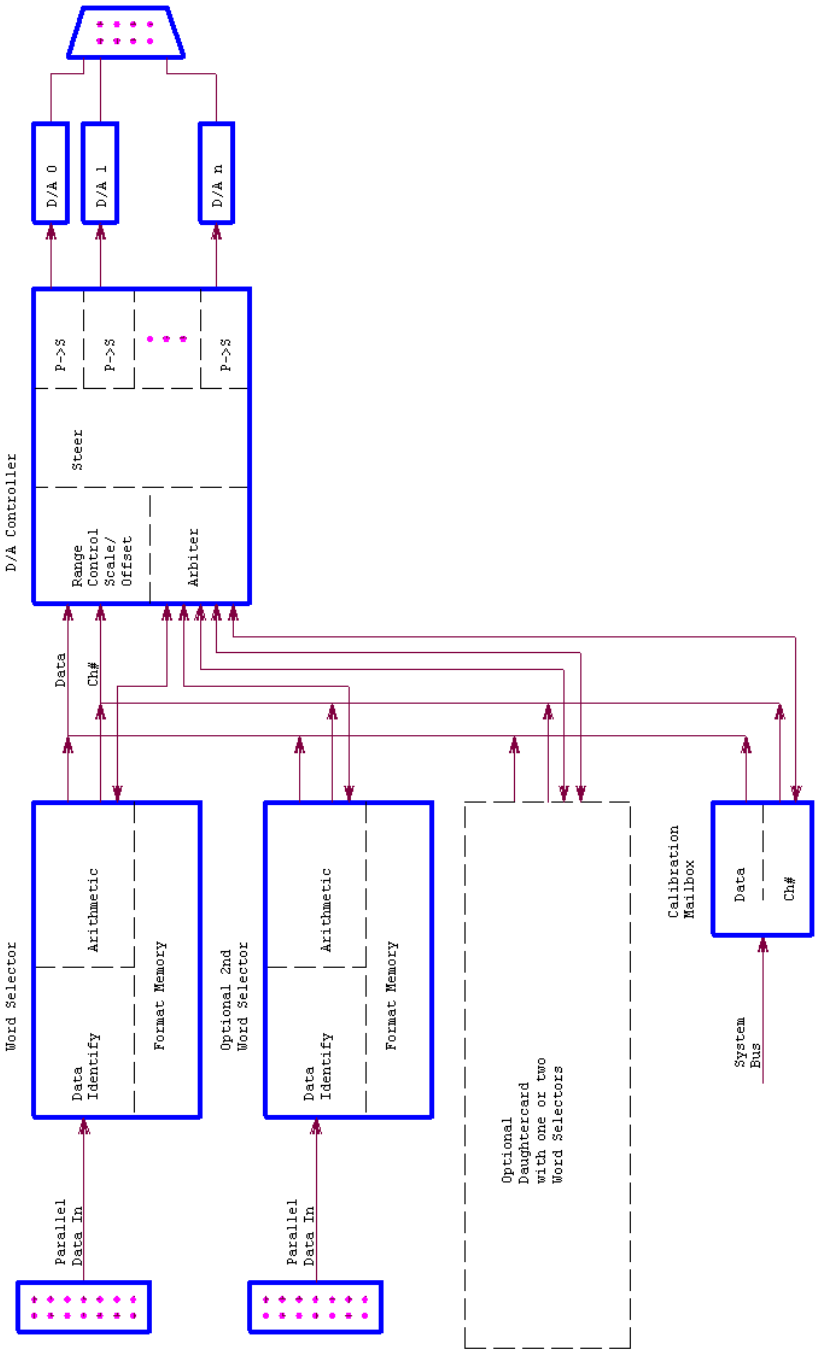


Figure 4-1 LS-71 Block Diagram

If the flag bit is set (which it will be the first time and once a frame thereafter) the LSBs (how many is controlled by the F08 and F10 bits) are stretched into an 18-bit number that looks like this:

```
00 1000 00ff ffff ffff
```
(f is bits from FIFO)

This number is used to address RAM. The part of RAM starting at address 0x08000 + index is loaded with 1 + (frame_length * index). When the card is loaded, location 0x08000 is 1, 0x08001 is 1 + words_per_frame, 0x08002 is 1 + twice words_per_frame, etc. A RAM cycle takes place, and the value is loaded into a 16-bit word counter. Then the word selector idles and waits for the FIFO again.

If the flag bit is clear (usual case) the LSBs are considered to be a data value. They're loaded into what's called the T (for Top) register in the word selector. Then another 18-bit RAM address is formed, looking like this:

```
00 0www wwww wwww www
```

where w is the 15 MSBs of the word counter. A RAM cycle takes place. One byte of the RAM data is selected (high byte if LSB of word counter XOR BIGEND = 1, else low byte.) The result t is loaded into an 18-bit program counter like this:

```
00 11tt tttt tt00 0000
```

Then the word counter is incremented. There are thus 256 possible starting points for processing.

The word selector then starts running the program at the address in the program counter. Each time the program counter is used, it is incremented so instructions are fetched from consecutive addresses like a regular computer. The word selector runs along fetching and executing instructions. These instructions do -- whatever -- to manipulate the data in the T register. Eventually near the end there's usually an instruction in the form:

```
q011 00x0 000d dddd
```
(d is a d/a channel number)

This results in the 14 LSBs of the T register being written in the direction of DAC d. That data, and d, is written to *another* FIFO whose output is watched by the actual DAC controller.

When the word selector executes an instruction with the q bit set, it idles after that instruction, and waits on the FIFO again.

One program space needs to start with a QUIT instruction. Any data item not selected for any DAC needs to point to that one-instruction program:

```
1000 xxxx xxxx xxxx
```

4.8 The Word Selector *"Stack Machine"*

The word selector must assign a tag number to every word in the incoming data stream and executes a program whose starting address is a function of that tag number. Nearly every format will include unselected data, so a tag (0 suggested) needs to be reserved as "untagged." All unselected words have this tag number and as such point to a program that simply quits. There are 256 numbers available.

The word selector processor is a simple stack machine. The term "stack machine" means numbers going in and out appear in a "T" (top entry) register at the top of a stack Behind T is another register called "N" (Next entry.) There are more than enough (sixteen) additional registers behind N. Several basic types of operations occur.

Some operations explicitly do not change the stack.

An operation that merely changes T is called a "put." This includes the automatic placing of raw data from the FIFO in T when a program starts, and any instructions that are specifically defined as manipulating or replacing T.

COPY, LOADK, LOADL perform what is known as a "push." A new value is placed in T, but simultaneously the original contents of T are copied to N, the original contents of N are copied to the register behind it, the original contents of that register are copied to the register behind *it*, etc.

ADD and AND are "binary" operations, meaning they need two operands. These instructions take the values of T and N as their operands and leave their result in T. Simultaneously, N is loaded from the register behind it, that register is loaded from the register behind *it*, etc. The original values of T and N are lost.

When the program for a tag starts, it can assume only that raw data from the input is in T. The program runs until a QUIT appears, then the word selector goes back and waits for the next data item. So untagged words execute a program consisting only of a QUIT.

4.8.1 Instructions

Program instructions are written into the 64-WORD program space for the Tag, starting at offset 0. The area at the end of each program space (typically only the last location) is reserved as a scratch location.

Instructions are divided into OP (8 MSBs) and K (8 LSBs) fields. There's not that many opcodes. They follow by OP field, in a numeric sequence.

Some instructions implicitly specify operand(s) and do not use the K field. For others K is a small address offset or a constant. The LOADL instruction needs a 16-bit operand, so it occupies two instruction locations.

Table 4-8 Instruction Repertoire

Opcode	Mnemonic	Description
0x00	NOP	No-op. Doesn't do anything. K ignored.
0x80	QUIT	Quits processing Tag. K ignored.
0x10	SRL	Shifts T right logically K bits. Only 4 LSBs of K have meaning.
0x11	SRC	Shifts T right circularly K bits. Only 4 LSBs of K have meaning.
0x12	SRA	Shifts T right arithmetically K bits. Only 4 LSBs of K have meaning.
0x20	AND	T is replaced by the logical AND of T and N. The stack is popped and N discarded. K ignored.
0x21	ADD	Same as AND except returns arithmetic sum. Carry status from the add saved in a CF flag. K ignored.
0x22	POP	The stack is popped and T is discarded. K ignored.
0x23	XOR	Same as AND except T is replaced with logical EXCLUSIVE OR ot T and N. K ignored.
0x30	WR	T is copied out to *logical* D/A channel K. K= 0..127 meaningful.
0x32	WRP	Save as WR except stack is popped and T is discarded.
0xB0	WRQ	Same as WR except QUITs afterward.
0x40	JZ	Continue processing with next following instruction if T is non-zero. Otherwise, jump and continue with instruction at K 0..5.
0x41	JNZ	Like JZ except jump if non-zero.
0x42	JN	Like JZ except jump if T MSB set.
0x43	JP	Like JZ except jump if T MSB clear.
0x44	JC	Like JZ except jump if CF set.
0x45	JNC	Like JZ except jump if CF cleared.
0x46	J	Like JZ except jump unconditionally.
0x48	JT	Ends current process and immediately starts the process associated with Tag K. Current process must leave a meaningful value in T (such as the original raw data) for this to be useful.
0x50	LOADI	Replaces T with WORD at memory location T. K bits 6..7 extend address to 18 bits. K bits 0..5 are meaningless. Either to fetch from LUT or to get value from SAVE when Tag was some other number.
0x60	LOADK	Pushes stack and sets T from K. (8 MSBs of T = 0.)
0x61	LOADL	Pushes stack and sets T from next WORD of instruction. This is the only 32-bit long instruction. K ignored.
0x62	COPY	Pushes stack and creates another copy of T. K ignored.
0x70	SAVE	Copies T to memory WORD location 0x0C000 + 0x40 * Tag + K (6 LSBs.) By convention set K to 0x3F except for the unlikely need to save more than one different value.
0x72	SAVEP	Same as SAVE except stack is popped and T is discarded.
0xF0	SAVEQ	Same as SAVE except QUITs afterward.

4.8.2 Microexamples

The instruction set for the LS-71 was designed with the idea that a limited set of operations were to be implemented. NOT on the list was real math to do EU conversions. The proper place for such calculations is the host computer. EU conversions on the LS-71 are implemented via a table lookup (see below).

The user may start with the sample routines below.

-- Throwing away unselected data expeditiously takes one instruction:

```
8000        QUIT
```

-- Executing arbitrary EU conversions from raw data to D/A's. These are done by table lookup, which not only gets rid of the problem of how to describe the math to the word selector, but makes the math itself an offline process. The lookup tables are decoupled from the D/A channels so they may be shared among them. An EU conversion takes 4 instructions:

```
1x0x        Shift Something    ; Align MSB according to table size
6100 xxxx   LOADL TABLE
2000        ADD
50xx        LOADI (TABLE >> 10)
```

The value of "Something" is a function of data alignment, word length, and table size. A table 2^n WORDS long needs the data converted to an n-bit wide index. If the data is left-aligned, the proper shift is an SRL with shift count is 16 – n. If the data is right-aligned, the proper shift is an SRC of 16 + word length – n if n is larger, or an SRL with shift count of word length – n if n is smaller. A shift of zero can be omitted.

-- Concatenating segmented words back together before doing any EU conversions.

The term concatenation refers to the action of juxtaposing a string L1 bits long with another string L2 long bits to yield a result L1 + L2 bits long. This is NOT the same as just adding the two bit strings together.

For this, at least two items from the data stream are needed, (call them "T1" and "T2.") and each must have its own Tag processes. One of them (for the data item earliest in time) runs for Tag T1 and does this:

```
110x        SRC   Something
6100 xxxx   LOADL MASK1  ; or LOADK if 8 LSBs or less
2100        AND
F03F        SAVEQ 0x3F
```

This process uses a circular shift to get the desired bits from the first data item aligned where they belong in the result. MASK1 is a mask with ones only for those desired bits. A logical AND then wipes out all but the desired bits. The program ends with a SAVEQ, which stores the result in memory at the last WORD of the Tag's program space.

The other process for Tag T2 does something complicated like this. Some optimization may be possible but it's hard to describe the circumstances (depends on which segment has the LSBs):

```
110x        SRC   Something
6100 xxxx   LOADL MASK2        ; or LOADK if 8 LSBs or less
2000        AND
6100 xx3F   LOADL 0xC03F + (0x40 * T1)
5000        LOADI 0            ; returns SAVEd value from T1
2100        ADD
                               ; Insert EU here as needed
B0xx        WRQ   DACNUM1      ; send data
```

This process shifts and masks data just like the T1 process, except it leaves a pattern of bits adjacent in the word. The LOADL instruction operand is the WORD address where Tag T1 left its partial result, so the LOADI instruction can fetch it. An ADD puts the two parts together. A logical OR would be more correct, but the processor lacks the operation. Following that, either the raw data can be output, or an EU conversion by table lookup can be inserted.

-- Realigning data for "raw" updates without EUs. For left-aligned data, the program is 2 instructions:

```
1002        SRL   2
B0xx        WRQ   DACNUM4
```

For right-aligned data the program is different:

```
110x        SRC   ((WORDLEN – 14) & 0x0F) ; believe it or not, this works!
B0xx        WRQ   DACNUM4
```

4.9 An Example

Providing a real LS-71 example takes a lot of boilerplate. In the beginning there is a format definition, arbitrarily chosen as:

8-bit words
20 words per frame
16 frames per major frame counting 0..15.

Measurement W is Tag 5, Word 4 of all frames, supercom interval 4.
Measurement X is Tag 6, Word 5 of all frames.

Measurement Y is Tag 7, Word 7 of even frames.
Measurement Z is Tag 8, Word 7 of odd frames.

Measurement W goes to DAC 2 in "raw" form, with no EU conversion.
Measurement X goes to DAC 0. The EU is Y = 106.3 ($Log_{10}(x+1)$).
Measurement Y goes to DAC 1. The EU is Y = X.
Measurement Z goes to DAC 3. The EU is Y = -X + 255.

All EU conversions normalized for domain and range of 0..255 to go with the data.

Lookup Tables here are 256 entries because they need be only as deep as data is wide.

Lookup Table 1 implements Y = X and is placed arbitrarily at WORD address 10100.

Lookup Table 2 implements Y = -X + 255 at WORD address 10200.

Lookup Table 3 implements Y = 106.3 ($Log_{10}(x+1)$) at WORD address 10300.

The decommutator is set to output left-aligned data. The DAC card has 32 DACs present.

The analog output range is 0..10V. Set the DAC Control Register to 0x85.

Tag 0 is reserved for unselected data.

Up to 64K WORDS could be reserved for a single lookup, at the point of absurdity. 4K WORDS maximum is a good compromise. There is room for 13 4K tables and 19 8K tables if you try to use up all the available RAM. Of course, the length of the download will increase linearly with the amount of calculating and downloading you do.

The Frame Lookup area points to the start word number of each minor frame for all the words in the major frame:

```
08000 0001          ; Frame 0
08001 0015          ; Frame 1
08002 0029          ; Frame 2
08003 003D          ; Frame 3
  ...
0800F 012D          ; Frame 15
```

The Tag Lookup Area converts the word number to a Tag number. Remember this memory is byte addressed. Here xx in the LSBs of the first location imply meaninglessness.

```
--------+
00000 00|xx
        +------------
00001 00 00          ; Frame 0
00002 06 05
00003 07 00
00004 00 05
00005 00 00
00006 00 05
00007 00 00
00008 00 05
00009 00 00
--------+
0000A 00|05
        +------------
0000B 00 00          ; Frame 1
0000C 06 05
0000D 08 00
0000E 00 05
0000F 00 00
00010 00 05
00011 00 00
00012 00 05
00013 00 00
    ...
```

Program memory space looks like this:

```
0C000 8000           QUIT ; Tag 0: Quit.

0C140 1002           SRL 2 ; Tag 5: Align 8-bit RA data to 14-bit DAC word
0C141 B002           WRQ 2

0C180 1008           SRL 8 ; Tag 6: Align 8-bit data to 8-bit table (i. e., nothing)
0C181 6100 0300      LOADL TABLE3 ; Lookup using Table3
0C183 2100           ADD
0C184 5040           LOADI (TABLE3 >> 10)
0C185 B000           WRQ 0

0C180 1008           SRL 8 ; Tag 7
0C181 6100 0100      LOADL TABLE1 ; Lookup using Table1
0C183 2100           ADD
0C184 5040           LOADI (TABLE1 >> 10)
0C185 B001           WRQ 1

0C180 1008           SRL 8 ; Tag 8
0C181 6100 0200      LOADL TABLE2 ; Lookup using Table2
0C183 2100           ADD
0C184 5040           LOADI (TABLE2 >> 10)
0C185 B003           WRQ 3
```

Lookup Table Space looks Like this. Tables 1 and 2 are linear, Table 3 is, of course, so nonlinear it suggests more bits were needed for the input parameter:

```
10100 0000          ; Table1
10101 0040
10101 0080
    ...
101FF 3FC0
10200 3FC0          ; Table2
10201 3F80
    ...
102FE 0040
102FF 0000
10300 0000          ; Table3
10301 0800
10302 0CAD
10303 1000
10304 1293
10305 14AE
10306 1675
10307 1800
10308 195B
10309 1A93
    ...
103FD 3FE9
103FE 3FF4
103FF 3FFF
```