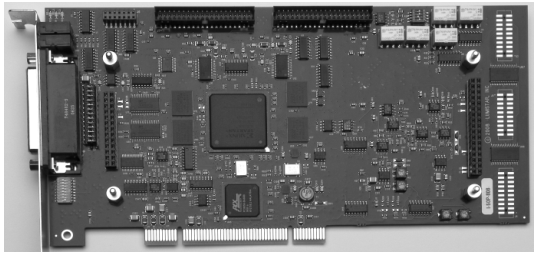
# “P2” Platform
# PCM Decommutator
# LS-50-P2 (R5)

# Technical Manual



| | |
|---|---|
| Document: | U500501 |
| Editor: | D. Spielman |
| Date: | August 2008 |
| Release: | Initial |

Lumistar, Inc.
5870 El Camino Real
Carlsbad, CA 92008
(760) 431-2181
www.lumi-star.com

Lumistar, Inc.
5870 El Camino Real
Carlsbad, CA 92008
(760) 431-2181
(760) 431-2665 Fax
www.lumi-star.com

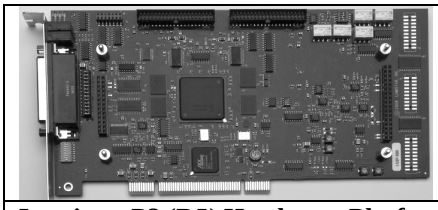
**TABLE OF CONTENTS**

**List of Tables**

**List of Figures**

# 1 Introduction

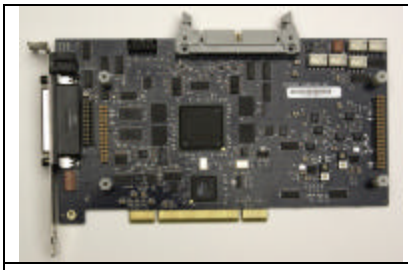
## 1.1 General



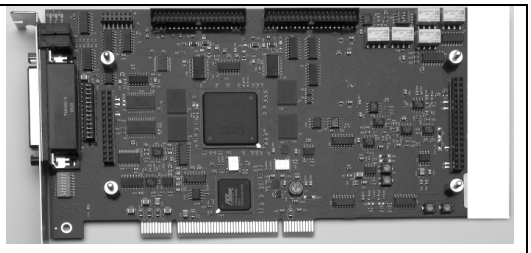
**Lumistar P2 (R5) Hardware Platform**

Technological advances in the capabilities of Field-Programmable Gate Arrays (FPGA) have enabled Lumistar to release this new "P2" hardware platform. The P2 platform allows the full functionality of up to two, LS-50 PCM decommutator/simulators, OR[1], up to two LS-70 High-Performance PCM simulators with dynamic data stream creation capabilities to be housed on a single reduced-length PCI card.

This P2 platform is capable of hosting one daughter-card, which may be an LS-40 PCM Bit Synchronizer, an LS-38 70MHz FM/SOQPSK Demodulator, or one or two of several types of low-power FM or SOQPSK RF Test Transmitters. Contact the factory for specific details.



**R3 Variant**



**R4 Variant**

## 1.2 Lumistar Universal Daughterboard Family

The P2 hardware platform may be equipped with one of a family of optional daughter-cards to add functionality. Current options include:

- Tunable Bit Synchronizer
- 70 MHz FM/SOQPSK Demodulator
- FM or SOQPSK Test Transmitters

**Lumistar LS-40-DB Bit Synchronizer**

**Lumistar LS-38 Multi-Mode Demodulator**

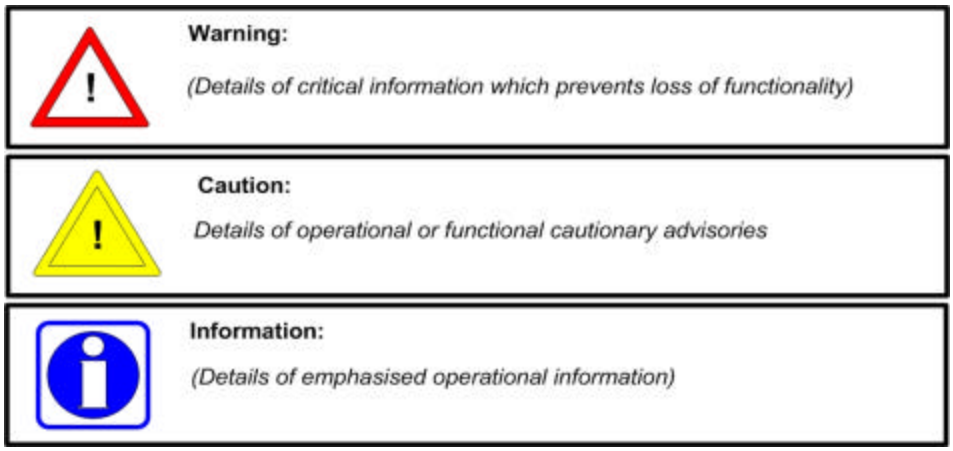
[1] One or two LS-50s, OR, one or two LS-70s, NOT both (a LS-50 & LS-70)

## 1.3 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-50 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.



## 1.4 Specifications

| Table 1-1 PCM Decommutator Specifications | |
|---|---|
| Input Data Rate | <100.0 bps to >33 Mbps |
| Input Signals | PCM Data and Symbol-Rate clock<br>(>15 Mbps: NRZ-L data & 0-degree clock) |
| Input Levels | Single-ended TTL & RS-422 |
| Word Length | Variable from 3 to 16 bits per word on a word-by-word basis |
| CRC checker | CRC16/CCITT |
| Minor Frame Length | 2 to 16,383 words per minor frame |
| Major Frame Length | Up to 1,024 minor frames per major frame |
| Bit Order | MSB or LSB-first (word-by-word basis) |
| Frame Sync Pattern | Up to 64 bits (any pattern, including "don't care" bits (X) may be used) |
| Frame Sync Location | Beginning or end of the frame |
| Frame Sync Strategy | Adaptive mode (search-lock-verify) & burst mode (search-lock) |
| Sync Error Tolerance | 0 to 15 bits (selectable) |
| Sync Slip Window | 1 or 3 bits wide (selectable) |
| Data Polarity | Normal, inverted or automatic |
| Major Frame Sync | FCC (FAC), SFID or URC |
| URC Location | Any 32 bit window within the first minor frame not including the last bit in the minor frame |
| SFID Location | Any series of contiguous bits not including the last bit in the minor frame |
| Time Reader Input Format | IRIG A, B, or G, 1 v p-p nominal. |
| Data Outputs | Automatic time tags for PCM data blocks<br>Time accessible in register space |
| System Output | Buffered output with status, time, & data.<br>Buffer size up to 64K words. |

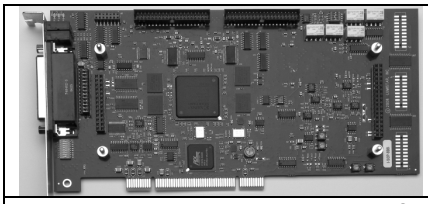
| Table 1-2 General PCM Simulator Specifications | |
|---|---|
| Minor Frame Length | 2 to 16,384 words per minor frame |
| Major Frame Length | Up to 1024 minor frames per major frame |
| Bit Order | MSB or LSB-first |
| Frame Sync Pattern | Up to 256 words. Normal or FAC |
| Major Frame Sync | FCC or SFID |
| Common Words | LS50: May be a single value or selected from a group of one minor frame. |
| Unique Words (LS50) | Seven may be programmed in any mainframe, super-commutated, or sub-commutated channel. Data may be changed while operating. |
| Waveform Words | Five may be programmed to appear in every frame at the same location. Data may be changed while operating. |
| PRN Data | Output may be pre-empted by one PRN generator with forced error for link bit error rate (BER) tests. Generator can produce an 11, 15, 17, 19, 21, 23, and 25-bit forward or reverse PRN sequence. |
| Outputs | PCM Data, symbol-rate clock & minor frame strobe.<br>Slave Clock out for sharing asynchronous embedded formats with a slave simulator. |
| Output Levels (Logic) | Single-ended TTL & RS-422 |
| Baseband Output | Software-controlled adjustable <200mv to 8V p-p.<br>Standard pre-modulation filters:<br>5-pole 0.25, 0.5, 1, 3, 6, 8, 12, 15MHz |
| Output Data Rate | 64 bps to 33 Mbps (NRZ codes)<br>64 bps to 15 Mbps (all codes)<br>Internal or external clock |
| Data Rate Stability | We use the best crystal oscillator money can buy. |
| PCM Codes | NRZ-L/M/S; Bi-Phase-L/M/S; DM-M/S; $M^2$, RNRZ-L-11/15, k=7 Convolutional Rate ½, 1/3 |
| Word Length | Variable from 3 to 16 bits per word on a word-by-word basis |
| CRC Generator | CRC16/CCITT forward/reverse |
| Frame Sync Pattern | Up to 256 words (any series of 0s or is 1s may be used) |
| Major Frame Sync | FCC (FAC), SFID |
| Master/Slave | TTL-level interfaces available to serve as a master or slave simulator for asynchronous embedded PCM format simulation. |
| Time Generator Output | IRIG A, B, or G |

| Table 1-3 Mechanical Specifications | |
|---|---|
| Form Factors | 7.55" long "Desktop" PCI (2.2 M33, D32) |
| Power Dissipation | 6.25 Watts |

| Table 1-4 Environmental Specifications | |
|---|---|
| Temperature (Operating) | 0 to 50 $^{o}$C |
| Temperature (Non-Operating) | -25 to +70 $^{o}$C |
| Humidity (Operating) | 10% to 90% Non-Condensing |
| Humidity (Non-Op) | Packaging must prevent contact with moisture and contaminants |
| Special Handling | Standard ESD methods required |

# 2 Installation

## 2.1 Addressing



**Lumistar P2 (R5) Hardware Platform**

The P2 Platform Board occupies both PCI I/O space and memory space. ***No address switch is used, as the address is determined by the system.*** 128 bytes of I/O space are always occupied. The card will respond to any access in its I/O space. The first 64 bytes of that space are assigned to the main ("Ch 0") channel, and accesses to the first byte will return an ASCII identifier string. If a second channel ("Ch 1") is configured, it will respond with its own identifier string, located 64 bytes up from the base address of the card.

The amount of memory space taken up by the card is circumstantial. If Ch 1 is configured, then twice as much space will be occupied and the upper half of that space will access Ch 1. The other factor is the memory-addressing mode recognized by the buffer memory. PCI cards are normally shipped in a "flat" addressing mode wherein the 128 KByte buffer memory is mapped one-to-one into PCI memory space. The configuration can be changed to activate a bankswitch register and maps the selected bank into 16Kbytes of MS-DOS real memory space. The user's computing environment may not allow for the use of this mode, but Lumistar uses it for testing purposes. In either case, if Ch 1 is present, it will map either 128K or 16K higher in system memory space.

## 2.2 Physical Installation

The P2 Platform board can be installed in any physical slot where it fits. Remove and discard the blanking plate from the chosen slot (save the screw(s)!) and carefully insert the card.

## 2.3 Indicators

Multiple LED indicators are provided. These indicators are shown in Figure 2-1 on page 11. Three chip LEDs are board ID indicators. These are connected to a static register and are intended for use by device drivers in environments where multiple cards are present to identify which board is assigned to which data stream.

Two rows of three indicators are visible through the faceplate and used as status indicators. For the LS-50 configuration, indicators [4..5] on the card are controlled by the decommutator. Indicator 4 is a minor frame lock indication. Indicator 5 is a major frame lock indication. Indicator 6 lights when the IRIG time reader detects a valid IRIG time carrier.

If Ch 1 is configured, Indicators 7 through 9 function in the same manner as indicators 4 through 6, reflecting the status of Ch 1. These indicators are shared with the daughter-card. Indicator 7 is a signal present indication. Indicator 8 is a bit synchronizer lock indication. For LS-40 modules, indicator 9 lights if the estimated $E_b/N_0$ exceeds 5dB. If

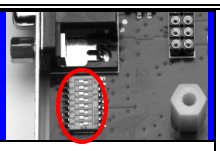
Ch 1 and the daughter-card interface are both in use, the indications are wire-ored together.

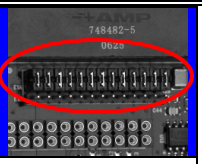
## 2.4 Interface

For I/O, the P2 hardware platform uses a 44-position female high-density subminiature "D" type connector designated J1. This connector has three rows of pins. Pin assignments for J1 are shown in Table 2-4 on page 10. On the board, there are far more signals than there are pins on J1. This situation creates the unavoidable pin-sharing complexity seen here. The first row of pins (1..15) are reserved for Ch 0. The second row of pins (16..29) are grounds. Pin 30 is a TTL-compatible 1PPS output from the IRIG time reader. The third row of pins are connected to the center row of patch array E1, adjacent to the J1 connector. Patch array E1 (see Figure 2-2 on page 12) allows the J1 pins to be dedicated to either a daughter-card module or to Ch 1 via the installation of a 2mm shunt patch.

An 8-position switch (designated SW1 on Figure 2-2) adjacent to the faceplate also affects the J1 pin-outs. The actions of this switch are defined in Table 2-1 on page 8. A four-position switch (designated SW2 on Figure 2-2) near the top board edge is applicable if an LS-40 Bit Synchronizer daughter-card is installed. Switch SW2 determines what termination, if any, is provided for the selected mezzanine input. (Unselected inputs are open.) The actions of SW2 are defined in Table 2-3 on page 9.

The LS-38 module does not provide all the I/O capability of the LS-40, so certain output signals are "manufactured" on-board. References to "LS-38" in these tables apply only to boards that have been configured to host an LS-38 module. Such boards also include an unreferenced SMA faceplate connector, used as the LS-38 70MHz input. References to "LS-40" apply only to boards hosting an LS-40.

| **Table 2-1 Switch SW1 Definition** |  |
|---|---|
| **SW1 Section** | **Definition** |
| 1 | Off: J1-11 is Ch 0 Decom Slave Clock Out<br>J1-13 is Ch 0 Decom Slave Data Out<br>On: J1-11 is Ch 0 Sim RS-422 Symbol Clock(–)<br>J1-13 is Ch 0 Sim RS-422 PCM Out(–) |
| 2 | Off: E1A-22 is Ch 1 Decom Slave Clock Out.<br>E1A-26 is Ch 1 Decom Slave Data Out<br>On: E1A-22 is Ch 1 Sim RS-422 Symbol Clock(–)<br>E1A-26 is Ch 1 Sim RS-422 PCM Out(–)<br>If used, E1A-22 is usually associated with J1-41<br>If used, E1A-26 is usually associated with J1-43 |
| 3 | On: J1-36 is Ch 0 Sim External Baseband Input |
| 4 | On: J1-37 is Ch 1 Sim External Baseband Input |
| 5 | Off: J1-5 is Ch 0 Decom Aux Data In/ RS-422 Status In(+)<br>On: J1-5 is Ch 0 Sim Slave Clock Output |
| 6 | Off: E1A-10 is Ch 1 Decom Aux Data In/ RS-422 Status In(+)<br>On: E1A-10 is Ch 1 Sim Slave Clock Output<br>If used, E1A-10 is usually associated with J1-35 |
| 7 | Off: J1-15 is IRIG Time Reader Input<br>On: J1-15 is also IRIG Time Generator Output |
| 8 | On: J1-15 is terminated into 100 ohms to ground. |

| Table 2-2 E1 Patch Definition | | | | |  |
|---|---|---|---|---|---|
| **E1A** | **Signal** | **E1A** | **Signal** | **E1B** | **(LS40) Signal** |
| 26 | See Table 2–1 SW1-2 | 25 | J1-43 | 13 | NRZ Out(–) |
| 24 | Ch 1 Sim PCM Out(+) | 23 | J1-42 | 12 | Input 6 / Input 2(+)* |
| 22 | See Table 2–1 SW1-2 | 21 | J1-41 | 11 | Clock Out(–) |
| 20 | Ch 1 Sim Symbol Clk(+) | 19 | J1-40 | 10 | Input 8 / Input 4(+) |
| 18 | Ch 1 Sim Baseband Out | 17 | J1-39 | 9 | Input 4 / Input 4(–) |
| 16 | Ch 1 TTL Aux Clock In<br>LS-38 Tape Out | 15 | J1-38 | 8 | Tape Out |
| 14 | Ch 1 Sim Slave Data In<br>Ch 1 Sim TTL Ext Data In | 13 | J1-37 | 7 | Input 3 / Input 3(–) |
| 12 | Ch 1 Sim Ext Clock In<br>Ch 1 RS-422 Status In(–) | 11 | J1-36 | 6 | Input 1 / Input 1(–) |
| 10 | See Table 2–1 SW1-6 | 9 | J1-35 | 5 | Input 7 / Input 3(+) |
| 8 | Ch 1 TTL Ext Sync In<br>Ch 1 RS-422 Data In(–) | 7 | J1-34 | 4 | Input 5 / Input 1(+) |
| 6 | Ch 1 TTL Ext Status In<br>Ch 1 RS-422 Data In(+) | 5 | J1-33 | 3 | Lock Out<br>Lock Out |
| 4 | Ch 1 TTL Data In<br>Ch 1 RS-422 Clock In(–) | 3 | J1-32 | 2 | NRZ Out(+) |
| 2 | Ch 1 TTL Clock In<br>Ch 1 RS-422 Clock In(+) | 1 | J1-31 | 1 | Clock Out(+) |
| *LS-40 Input 2 / Input 2(–) is J1-9; differential Input 2 is usually unusable. | | | | | |

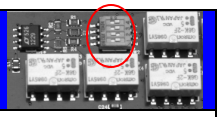
| Table 2-3 Switch SW2 Definition |  |
|---|---|
| **Section** | **Definition** |
| 1 | On: Selected input is terminated into 50 ohms to ground.<br>Use for single-ended inputs only. |
| 2 | On: Selected input is terminated into 75 ohms to ground.<br>Use for single-ended inputs only. |
| 3 | On: Selected inputs terminate into 120 ohms to each other.<br>Use for differential inputs only. |
| 4 | Not used. |

**Table 2-4 J1 I/O Connector Pin-out**

| Pin | Signal(s) | Pin | Signals(s) |
|---|---|---|---|
| 1 | Ch 0 TTL Clock In<br>Also Ch 0 RS422 Clock In(+) | 31<br>E1A-1 | E1A-2: Ch 1 TTL Clock In<br>Also Ch 1 RS422 Clock In(+) |
| | | | E1B-1: LS40 Clock Out(+) |
| | | | LS38: Clock Out(+) |
| 2 | Ch 0 TTL Data In<br>Also Ch 0 RS422 Clock In(–) | 32<br>E1A-3 | E1A-4: Ch 1 TTL Data In<br>Also Ch 1 RS422 Clock In(–) |
| | | | E1B-2: LS40 NRZ Out(+) |
| | | | LS38: NRZ Out(+) |
| 3 | Ch 0 TTL Ext Status In<br>Also Ch 0 RS422 Data In(+) | 33<br>E1A-5 | E1A-6: Ch 1 TTL Ext Status In<br>Also Ch 1 RS422 Data In(+) |
| | | | E1B-3: LS40 Lock Out |
| 4 | Ch 0 TTL Ext Sync In<br>Also Ch 0 RS422 Data In(–) | 34<br>E1A-7 | E1A-8: Ch 1 TTL Ext Sync In<br>Also Ch 1 RS422 Data In(–) |
| | | | E1B-4: LS40 Input 5 / Input 1(+) |
| 5 | Ch 0 Decom Aux Data In<br>Also RS422 Status In(+)<br>SW1-5 On: Ch 0 Sim Slave Clock Out | 35<br>E1A-9 | E1A-10: Ch 1 Decom Aux Data In<br>Also RS422 Status In(+)<br>SW1-6 On: Ch 1 Sim Slave Clock Out |
| | | | E1B-5: LS40 Input 7 / Input 3(+) |
| 6 | Ch 0 Sim External Clock In<br>Ch 0 RS422 Status In(–)<br>Ch 0 Alternate Clock In | 36<br>E1A-11 | E1A-12: Ch 1 Sim Ext Clock In<br>Ch 1 RS422 Status In(–) |
| | | | E1B-6: LS40 Input 1 / Input 1(–) |
| | | | SW1-3 On: Ch 0 Sim Ext Baseband In |
| 7 | Ch 0 Sim Slave/TTL Ext Data In<br>Also Ch 0 Alternate Data In | 37<br>E1A-13 | E1A-14: Ch 1 Sim Slave/Ext Data In<br>Ch 1 Sim TTL Ext Data In |
| | | | E1B-7: LS40 Input 3 / Input 3(–) |
| | | | SW1-4 On: Ch 1 Sim Ext Baseband In |
| 8 | Ch 0 TTL Aux Clock In | 38<br>E1A-15 | E1A-16: Ch 1 TTL Aux Clock In<br>LS38: Tape Out |
| | | | E1B-8: LS40 Tape Out |
| 9 | Ch 0 Sim Baseband Out<br>Also LS40 Input 2 | 39<br>E1A-17 | E1A-18: Ch 1 Sim Baseband Out |
| | | | E1B-9: LS40 Input 4 / Input 4(–) |
| 10 | Ch 0 Sim Clock Out(+) | 40<br>E1A-19 | E1A-20: Ch 1 Sim Clk(+) |
| | | | E1B-10: LS40 Input 8 / Input 4(+) |
| 11 | Ch 0 Decom Slave Clock Out<br>SW1-1 On: Ch 0 Sim RS422 Clock(–)<br>LS70: Ch 0 Sim BitRate Clock Out | 41<br>E1A-21 | E1A-22: Ch 1 Decom Slave Clock Out.<br>SW1-2 On: Ch 1 Sim RS422 Clock(–) |
| | | | E1B-11: LS40 Clock Out(–) |
| | | | LS38: Clock Out(–) |
| 12 | Ch 0 Sim PCM Out(+) | 42<br>E1A-23 | E1A-24: Ch 1 Sim PCM Out(+) |
| | | | E1B-12: LS40 Input 6 |
| 13 | Ch 0 Decom Slave Data Out<br>SW1-1 On: Ch 0 Sim RS422 PCM (–)<br>LS70: Ch 0 Sim NRZL Data Out | 43<br>E1A-25 | E1A-26: Ch 1 Decom Slave Data Out<br>SW1-2 On: Ch 1 Sim RS422 PCM (–) |
| | | | E1B-13: LS40 NRZ Out(–) |
| | | | LS38: NRZ Out(–) |
| 14 | Ch 0 Sim Scope Trigger Out. | 44 | Ch 1 Sim Scope Trigger Out. |
| 15 | IRIG Time. SW1-7 On: IRIG Time Out<br>SW1-8 On: 100 ohms to ground | 16-29 | Ground. |
| | | 30 | IRIG reader 1pps Out. |

The P2 platform board products are shipped with two mating pigtail cables to interface with the 44-position J1 connector. Both cables provide single-ended signals. The LS-40 bit sync interface cable is documented in Figure 2-3 on page 13. The LS-38 demodulator interface cable is documented in Figure 2-4 on page 15.

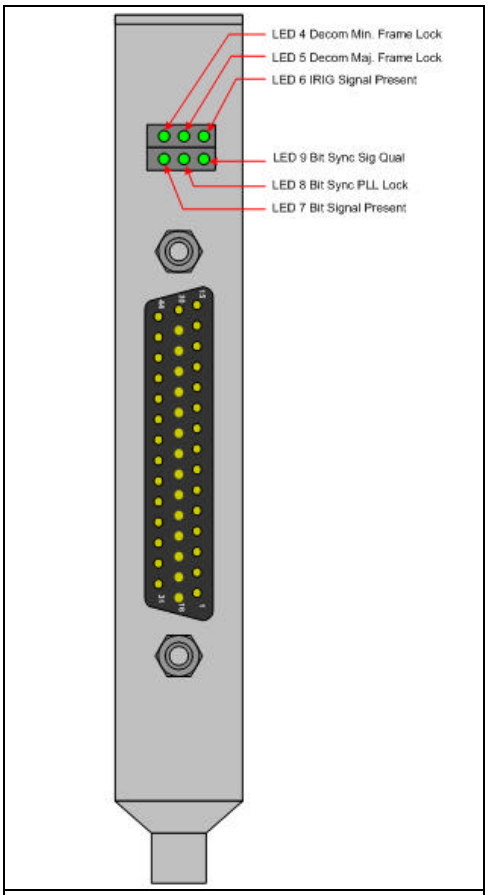
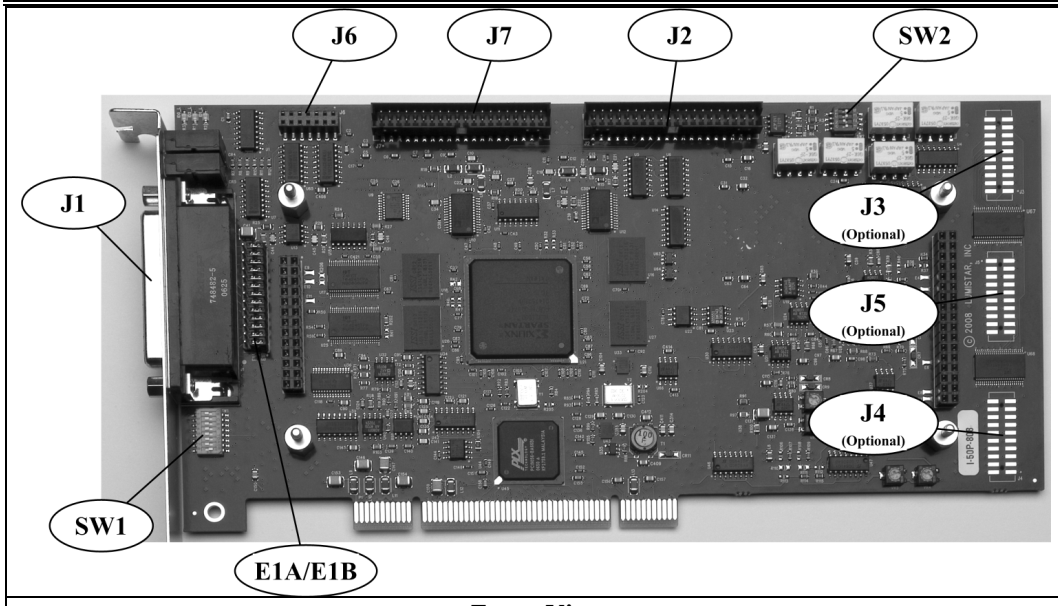
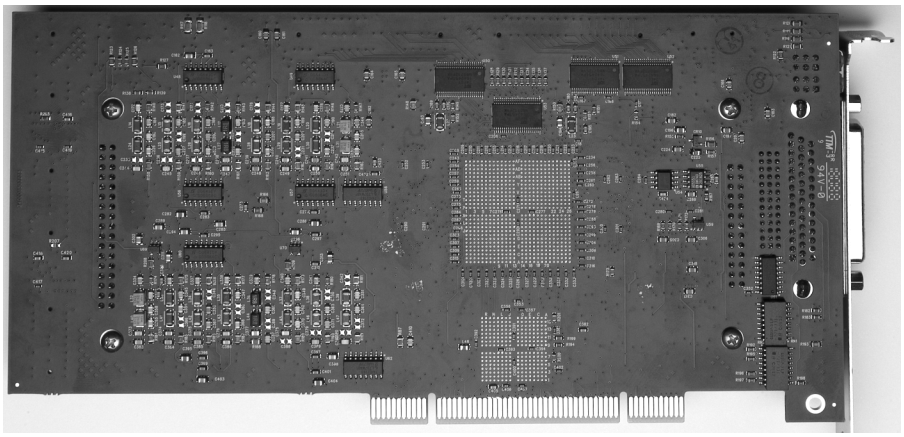



**Figure 2-1 Front Plates.**

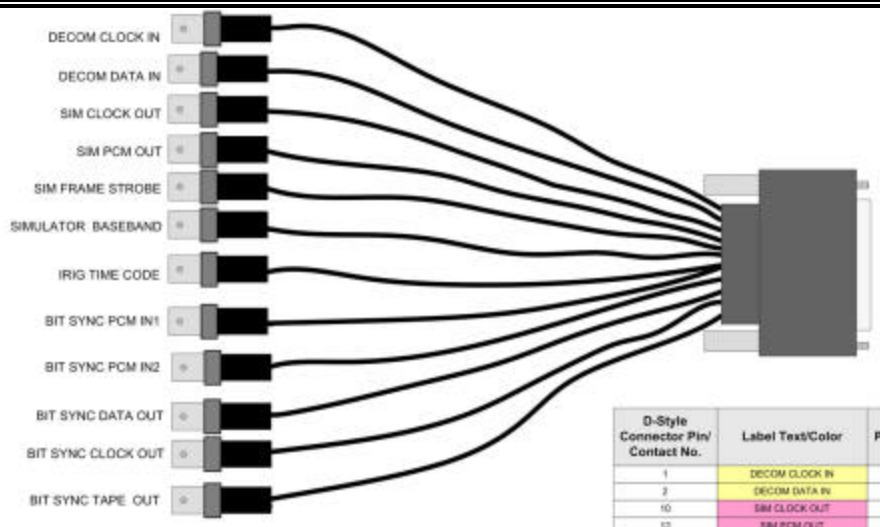



**Front View**



**Rear View**

**Figure 2-2 LS-50-P2 (R5) Card Major Component Call-Outs**



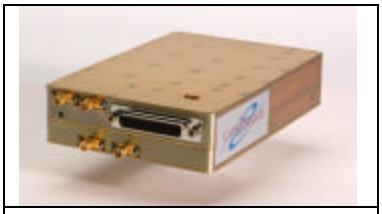

| D-Style Connector Pin/ Contact No. | Label Text/Color | Pigtail Type | Signal Details |
|---|---|---|---|
| 1 | DECOM CLOCK IN | BNC-F | DECOM CLOCK IN |
| 2 | DECOM DATA IN | BNC-F | DECOM DATA IN |
| 10 | SIM CLOCK OUT | BNC-F | SIM CLOCK |
| 12 | SIM PCM OUT | BNC-F | SIMULATOR NRZ-L |
| 14 | SIM FRAME STROBE | BNC-F | SIMULATOR FRAME STROBE |
| 15 | IRIG TIME CODE | BNC-F | IRIG TIME CODE INPUT/OUTPUT |
| 9 | SIMULATOR BASEBAND | BNC-F | SIMULATOR BASEBAND |
| 32 | BIT SYNC DATA OUT | BNC-F | BIT SYNC DATA OUT |
| 31 | BIT SYNC CLOCK OUT | BNC-F | BIT SYNC CLOCK OUT |
| 36 | BIT SYNC PCM IN1 | BNC-F | BIT SYNC PCM INPUT 1 |
| 37 | BIT SYNC PCM IN2 | BNC-F | BIT SYNC PCM INPUT 2 |
| 38 | BIT SYNC TAPE OUT | BNC-F | BIT SYNC PCM OUTPUT |
| 16,17,20,21,22,23,24, 25,26,29, | (N/A) | (N/A) | Ground |

**Figure 2-3 LS-50P2/LS-40 Pigtail Connector Assembly (Single-Ended Signals) [PN: C050204]**

## 2.5 Parallel Output

The P2 hardware platform also provides a parallel output port. This output appears at connector J2 as shown in Figure 2-2 on page 12. The pin-out for J2 is shown in Table 2-5 on page 14. If a Ch 1 decommutator is configured, it has its own parallel output on connector J7. The Ch 1 pin-out is identical to the J2 pin-out.

## 2.6 Optional RF Device Control Output



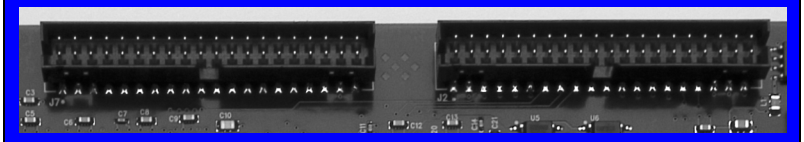
**LS-25-B**

The P2 hardware platform provides for optional control of an external RF device such as the Lumistar LS-25-B drive-bay receiver shown right. If Ch 1 is not configured, then the external module for Ch 0 may be connected to J3. In the R3 and R4 variants of the P2 hardware platform, J3 is a 14-pin 53780-type Molex connector located at the top right board edge on the rear of the card. In the R5 variant, J3 is an 87832-type Molex 2x10 header located on the right board edge on the front of the card. The pin-out of J3 is shown in Table 2–6 for pins 1 through 20. If Ch 1 is configured, then J3 is connected to the external module, and the external module associated with Ch 0 is connected to J5. In this context, transmit control refers to RS-232 signals intended to connect a transmitter or other such device with RS-232 control.

**LS-40-B**

For boards with serial numbers 750 and up, three 2x10, 2mm-pitch male header connectors (J3, J4, and J5) are located on the right board edge on the front of the card. J3 and J4 are associated with the Ch 0 PCM Simulator. J5 is associated with the Ch 1 PCM Simulator. If Ch1 is configured as a PCM decommutator, then J5 is turned around and serves to control an external LS-40-B dive-bay PCM bit synchronizer associated with the Ch 1 decommutator. In this scenario, the "alternate" pin assignments in Table 2–6 come into effect.

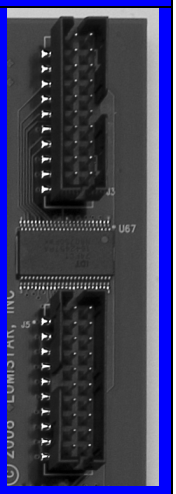
**Table 2-5 Parallel Output Pin-out**



| J2 (J7) Pin | Signal | J2 (J7) 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 | Lock | 32 | Ground |
| 33 | MFLock | 34 | Ground |
| 35 | Ground | 36 | Ground |
| 37 | Ground | 38 | Ground |
| 39 | Ground | 40 | Ground |

**Table 2-6 Optional Connector J3/J5 RF Control Pin-out**

| Pin | J3 Signal | J5 Alternate Signal |
|---|---|---|
| 1 | PCM Data Out | Bit Sync NRZL Out |
| 2 | Ground | Ground |
| 3 | Symbol Clock Out | Bit Sync Clock Out |
| 4 | Ground | Ground |
| 5 | RS232 Control Out | RS232 Control Out |
| 6 | RS232 Control In | RS232 Control In |
| 7 | Ground | Ground |
| 8 | Attenuator Control Serial Data | Bit Sync !Lock Status |
| 9 | Attenuator Control Serial Clock | Bit Sync !Signal Present Status |
| 10 | Ground | Ground |
| 11 | Attenuator Control Chip Select | Bit Sync !Signal Quality Status |
| 12 | RF Mode Switch Control | Device !Present Status |
| 13 | Ground | Ground |
| 14 | RF Output Amplifier Control | Not defined |
| 15 | Ground | Ground |
| 16 | Secondary Att Control Serial Data | Not defined |
| 17 | Ground | Ground |
| 18 | Secondary Att Control Chip Select | Not defined |
| 19 | ADC Chip Select (J4 Only) | Not defined |
| 20 | ADC Data Output (J4 Only) | Not defined |



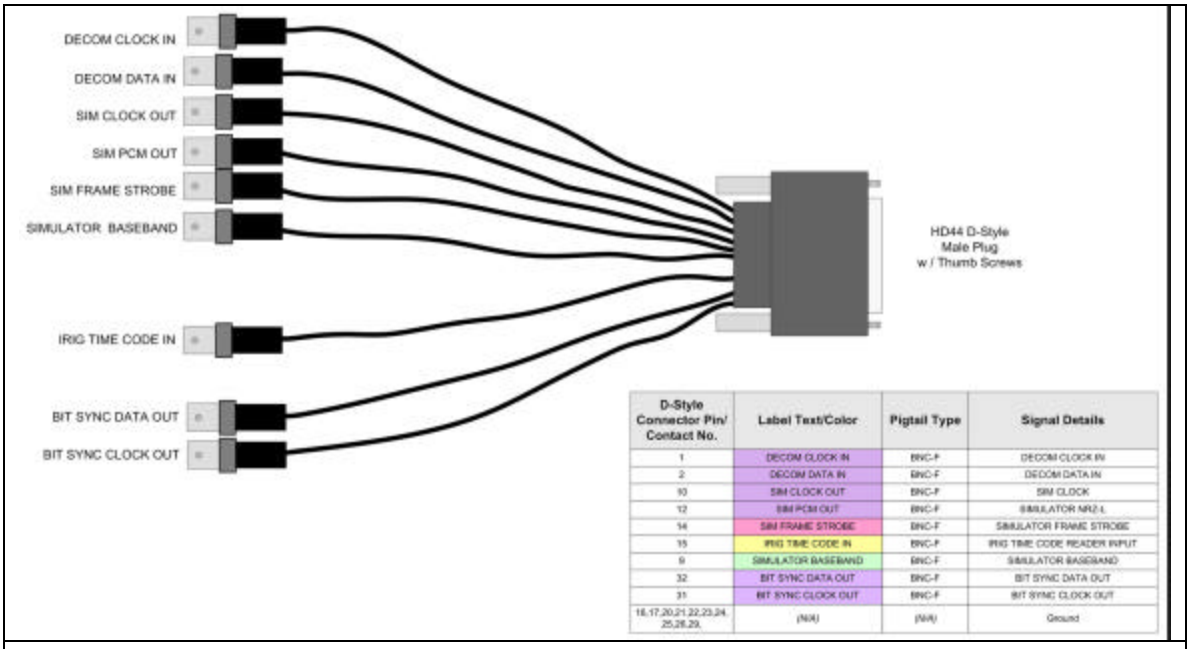



**Figure 2-4 LS-50/LS-38 Pigtail Connector Assembly (Single-Ended Signals) [PN: C050203]**

# 3 Operation of the P2 Platform Board With The LDPS Software

The LS-50 Multi-function PCM Decommutator incarnation of the P2 hardware platform can be setup and controlled by using the Lumistar Data Processing System (LDPS) software (shown below).

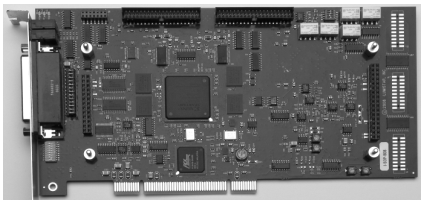


**Figure 3-1 LDPS Status Display for the LS-50**

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 (such as the LS-50). The server archives the data, formats the data into a normalized format, 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).

To initially configure the LS-50, 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 resulting System Manager shown below left, select the "Enable" check box next to the Ls50P2 button. The Ls50P2 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 as shown in the figure allows the LDPS application to operate when a LS-50 board is not installed in the system.
3. From the System Manager, click the Ls50P2 button. This will launch the "Ls50P2 Ver X.YY (Decom)" 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-50 card, follow the procedures outlined in paragraphs 3.1.

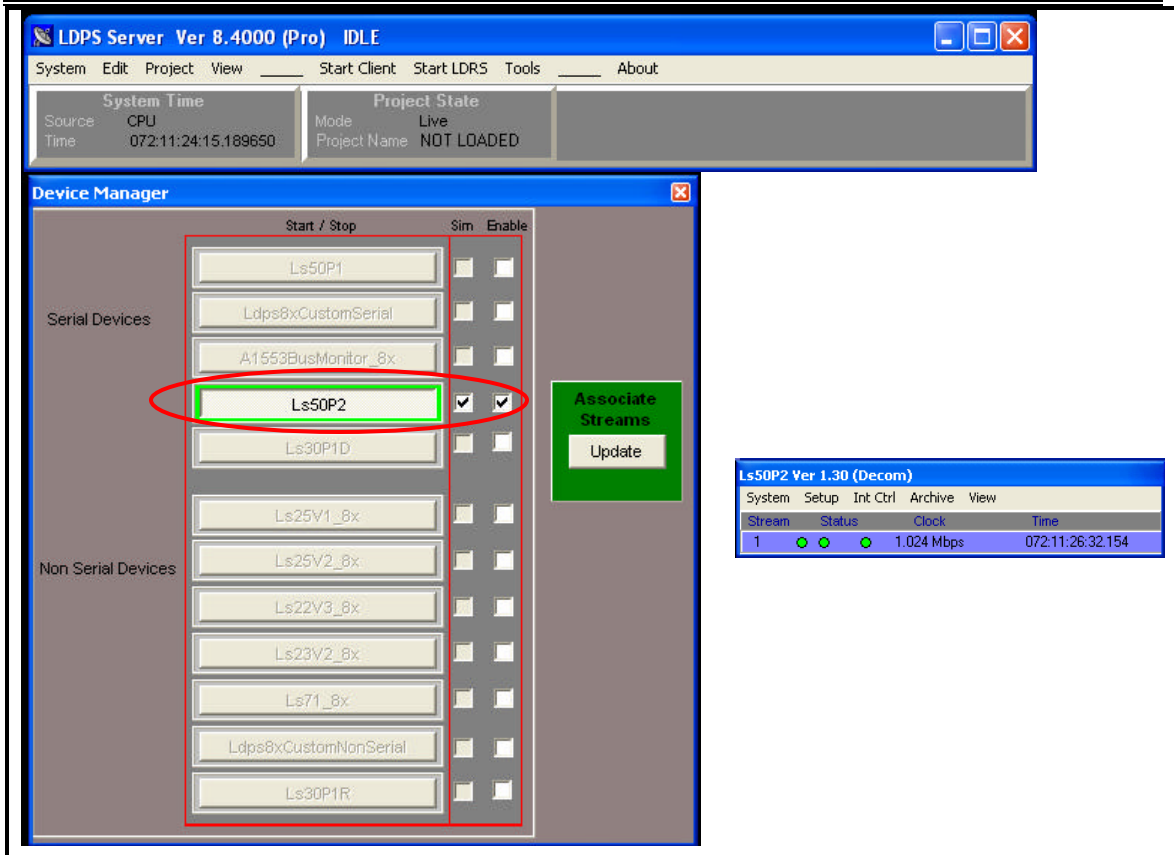



**Figure 3-2 LDPS Server Application Windows**

## 3.1 Configuring The LS-50 Hardware

From the "Ls50P2 Ver X.YY (Decom)" display[2] shown below in Figure 3-3, click *"Setup"* and then *"Stream 1" (Setup ? Stream 1***).**

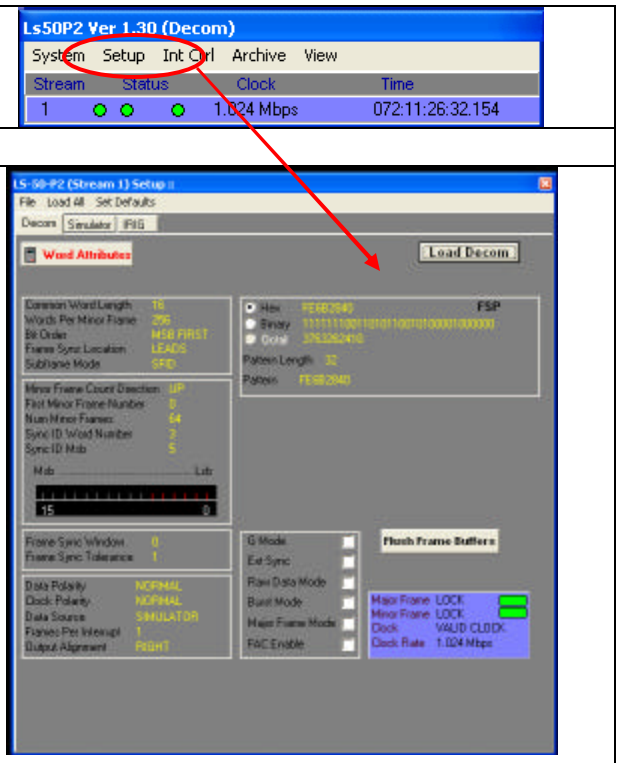
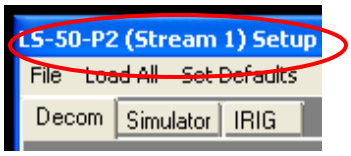


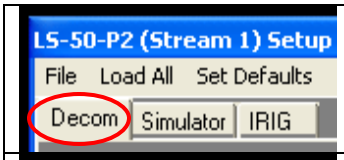
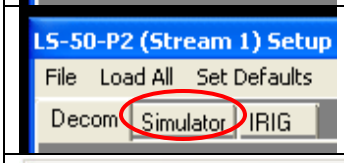
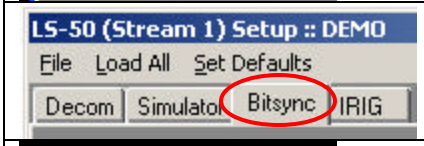
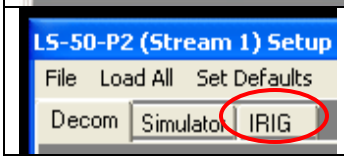

**Figure 3-3 Configuration Menus/Controls for the LS-50**

The "LS-50 (Stream 1) Setup" display shown above in Figure 3-3 is divided into several regions. Below the window header are the *"File," "Load All,"* and *"Set Defaults"* commands (more about these later). Each of the LS-50's four main functions have their own setup tab. To completely configure the LS-50, visit each tab in turn and configure the functions. After the setup configuration is complete, save the settings by invoking the *"File ? SaveAs"* command. To download the configuration to the LS-50 hardware, invoke the *"Load All"* command. To recall a previously defined LS-50 setup configuration, invoke the *"File ? Recall"* command and select the appropriate file from the file menu and then download the configuration to the LS-50 hardware by invoking the *"Load All"* command.

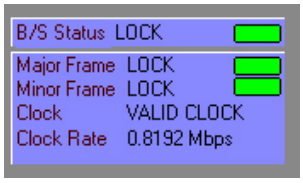



[2] This figure shows the server setup window in "Simulation" mode, where the LS-50 hardware is not installed in the system. When actual LS-50 hardware is installed, the server setup window appears as shown in Figure 3-14 on page 61

To invoke the controls for any of the tabs in the display, simply place the mouse curser in a region and right click. The resulting menus for the Decom tab are shown in Figure 3-4 on page 20 and are discussed in detail in the following paragraphs. The configuration setup for the Decommutator, Simulator, Bit Synchronizer, and IRIG Timecode functions are described in detail as indicated in the table below.

| | |
|---|---|
|  | See paragraph 3.1.1 on page 20 for more info on the Decommutator. |
|  | See paragraph 3.1.2 on page 38 for more info on the PCM Simulator. |
|  | See paragraph 3.1.3 on page 50 for more info on the Bit Synchronizer. |
|  | See paragraph 3.1.4 on page 56 for more info on the IRIG Timecode Reader/Generator. |

Each tab has a button control to load the setup information for the portion of the card displayed with the tab. Changes made with any of the controls will not take affect until this button is pressed.


There is also a window displayed (shown left) showing the status of some of the LS-50's functional states (like frame lock). This status display is updated at a ten Hertz rate. The user may load all four major functions (Decom, Simulator, Bitsync, and IRIG) from the *"Load All"* command on the menu next to the File menu). If any changes are made to an individual setup without loading, a red text will appear below the Load button (shown above right), indicating the displayed data does not match the cards' loaded data.

### 3.1.1 The LS-50 Decommutator Tab

The LS-50 decommutator setup tab and its associated menus and controls are shown in Figure 3-4 below. There are up to seven groups of controls displayed for the decommutator, depending on the setting of other controls. If a project is loaded from the LDPS server (see Figure 3-2 on page 17), then some portions of the window will not be able to be controlled.

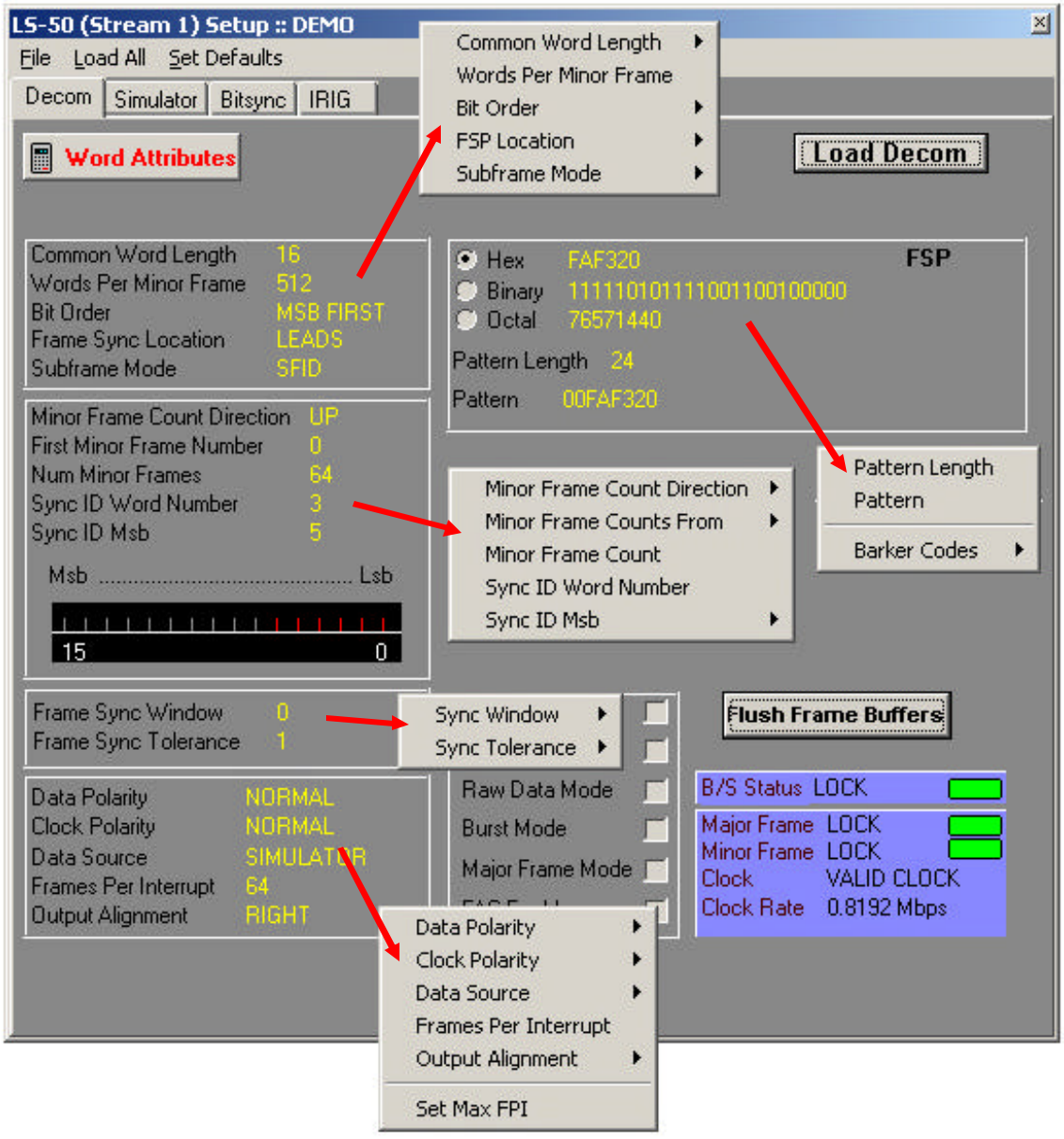



**Figure 3-4 The LS-50 Decom Tab Configuration Menus**

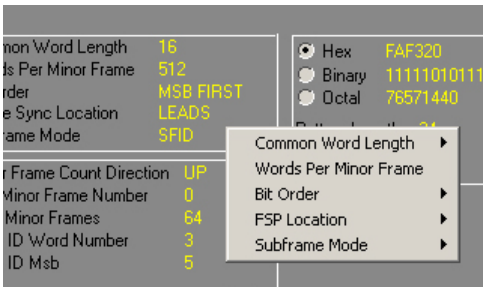
The seven control groups of the LS-50 decom tab include:

1. Major Frame Configuration
2. Minor Frame Configuration
3. Frame Synchronization Pattern (including optional URC)
4. Frame Sync Sensitivity Parameters
5. Data Source Configuration
6. Decommutator Modes
7. Word Attributes Control

| | |
|---|---|
|  | ***Definitions:***<br>• **Frame Synchronization Pattern** – A unique binary bit pattern used to indicate the beginning of a telemetry minor frame.<br>• **Frame Synchronizer** – Correlator & State Machine circuitry that recognizes unique bit patterns indicating the beginning of minor frame data. The frame synchronizer typically "searches" for patterns, "checks" for the recurrence of the pattern in the same position for several frame periods, and then "locks" on the pattern. |

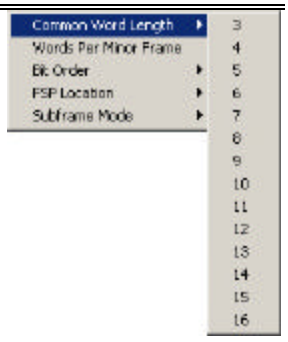
### 3.1.1.1 Major Frame Configuration

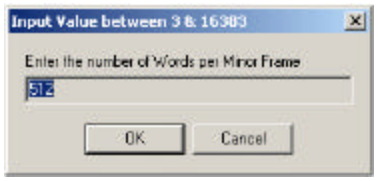
The major frame configuration consists of five controls/parameters that include: common word length, the number of words per minor frame, the bit order of the words in the frame, the frame synchronization patter location, and the subframe synchronization mode.




| | |
|---|---|
|  | ***Definitions:***<br>• **Major Frame** – An integer number of minor frames, not to exceed 256 per the IRIG-106 specification. *The LS-50 however can support up to 1024 minor frames per major frame.*<br>• **Minor Frame** – A fixed length block of data sub-divided into an integer number of fixed-length words. *The LS-50 can support up to 16,383 words per minor frame.* |

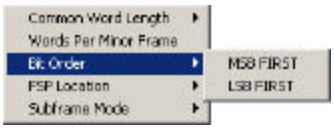
The **Common Word Length** may be set from 3 to 16 bits in length. The common word length defines the length in bits of the *majority* of words that make up a minor frame. Note, not all words in a minor frame need be of the same length. For example, the majority of the words in a minor frame could be 8-bits in length, and thus the common word length would be 8. However, several of the words might be 14 or 16 bits in length and would be individually specified using the Decommutator Word Attributes command function described in paragraph 3.1.1.9 on page 34

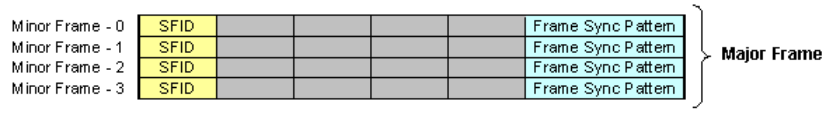


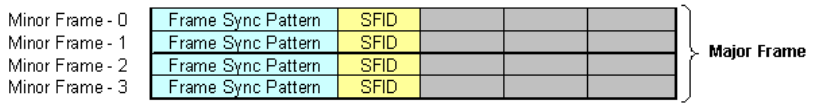

The minor frame length is defined by the user by invoking the **Words Per Minor Frame** command. Here, the user enters the number of words (of length specified by word attributes settings) that make up a minor frame. The minor frame length on the LS-50 may be between 3 and 16,383 words.

By invoking the **Bit Order** command, the user specifies for the common words of the minor frame whether the Most Significant Bit (MSB) is first, as read from left to right, or the Least Significant Bit (LSB) is first, again, read from left to right. Note, not all words in a minor frame need have the same bit order. For example, the majority of the words in a minor frame could have LSB first bit order. However, several of the words might be MSB first and would be individually specified using the Decommutator Word Attributes command function described in paragraph 3.1.1.9 on page 34




Note: The location of the frame synchronization pattern (FSP) may be visualized, as in the examples here, as either being at the beginning of the minor frame, or at the end. The location of the subframe identification (SFID) word(s) is arbitrary.



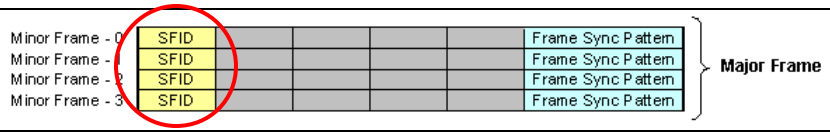

The user specifies the location of the FSP by invoking the **FSP Location** command, and selecting "TRAILS" or "LEADS."

To implement a subframe synchronization scheme, telemetry designers often add one or more "special" words to each minor frame. These special words are used by the frame synchronizer state machine to establish the location of the first minor frame in the major frame. The LS-50 supports three subframe synchronization modes: SFID, FCC, and URC.

***Definition:***

- **SFID** – The most commonly used subframe synchronization method is called Subframe Identification (SFID).

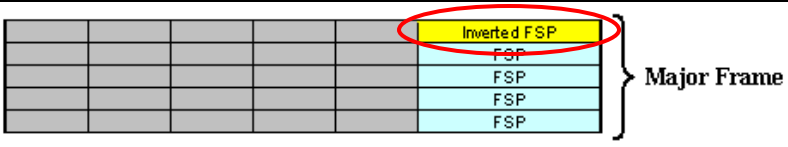
In this method, the synchronization pattern occupies one or more words in each minor frame. The SFID acts as a counter. The pattern value increments or decrements to a specific value and then resets.




***Definition:***

- **FCC** – Another commonly used subframe synchronization method is called Frame Code Complement (FCC).

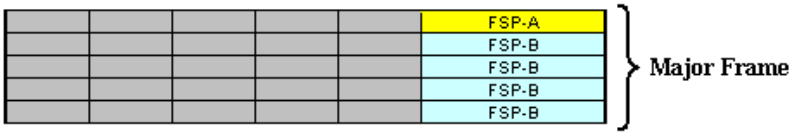
In this method, the complement (inverted) of the synchronization pattern is placed in the FSP location in minor frame-0. All other FSPs are not inverted. Because the complement of the frame synchronization pattern exhibits the same correlation properties as the true pattern, frame sync lock will not be compromised. Minimum sync overhead is attained using this method, although it requires longer subframe acquisition time than the SFID method.



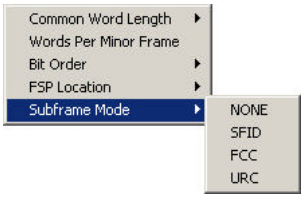
***Definition:***

- **URC** – A less commonly used subframe synchronization method is called Unique Recycle Code (URC).

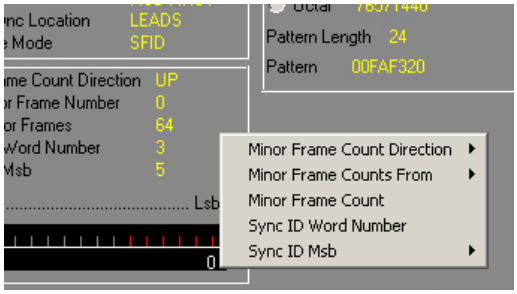
URC is a slight variation on the FCC method. For URC, the beginning of minor frame-0 is identified by a unique synchronization pattern **NOT** related to the primary synchronization pattern.




The user specifies the method of subframe synchronization by invoking the **Subframe Mode** command, and selecting "None," "SFID," "FCC," or "URC." Note, if the user selects the "URC" subframe synchronization mode, then a second frame synchronization pattern setup area will appear on the Decom setup tab as shown in Figure 3-5 on page 26.

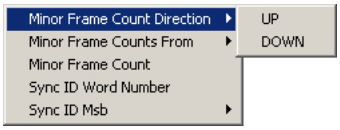


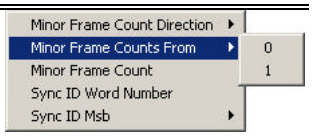

### 3.1.1.2 Minor Frame Configuration



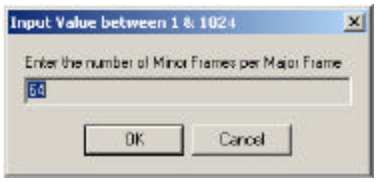

The minor frame configuration consists of five controls/parameters that include: Minor Frame Count Direction, Minor Frame Counts From, Minor Frame Count, Sync ID Word Number, and Sync ID MSB.

As mentioned previously, in the SFID mode, the synchronization pattern occupies one or more words in each minor frame and acts as a counter. The user may specify whether the pattern value increments or decrements from minor frame to minor frame by invoking the **Minor Frame Count Direction** command, and selecting "UP," or "DOWN."

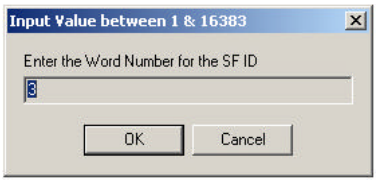




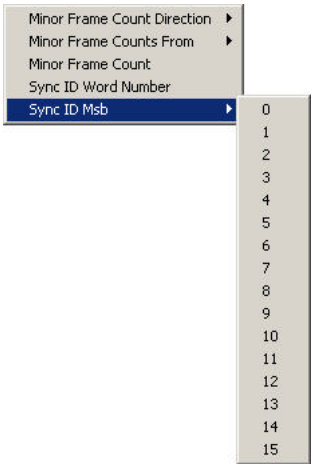

In some telemetry frame designs, the subframe counter in minor frame-0 will initially begin counting from a starting value of zero (0), while in other frame designs, the subframe counter will begin counting from a starting value of one (1). The user specifies one or the other of these two conditions by invoking the **Minor Frame Counts From** command.



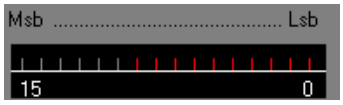

As mentioned previously, the major frame is composed of an integer number of minor frames, and the minor frame is a fixed length block of data sub-divided into an integer number of fixed-length words. By invoking the **Minor Frame Count** command, the user may specify the number of minor frames that make up the major frame. The LS-50 can support up to 1024 minor frames per major frame.




The location of the subframe identification (SFID) word(s) is arbitrary within the minor frame and may be specified by the user by invoking the **Sync ID Word Number** command. As the LS-50 can support up to 16,383 words per minor frame, the user may thus locate the SFID word anywhere within this range, provided it does not overlap or coincide with the frame synchronization pattern location.




As described previously, the SFID word is used as a counter, but it is not always the case that ALL of the bits in the SFID word are used for this purpose. For example, the SFID word might be 16-bits in length, but there might only be 512 minor frames in the major frame. In this scenario, a 9-bit counter ($2^9 = 512$) would be required and the user would specify the location of the counter within the larger 16-bit word by invoking the **Sync ID Msb** command and selecting the appropriate bit position for the most significant bit of the SFID counter. The Sync ID Msb is represented graphically in the minor frame configuration section as shown right.



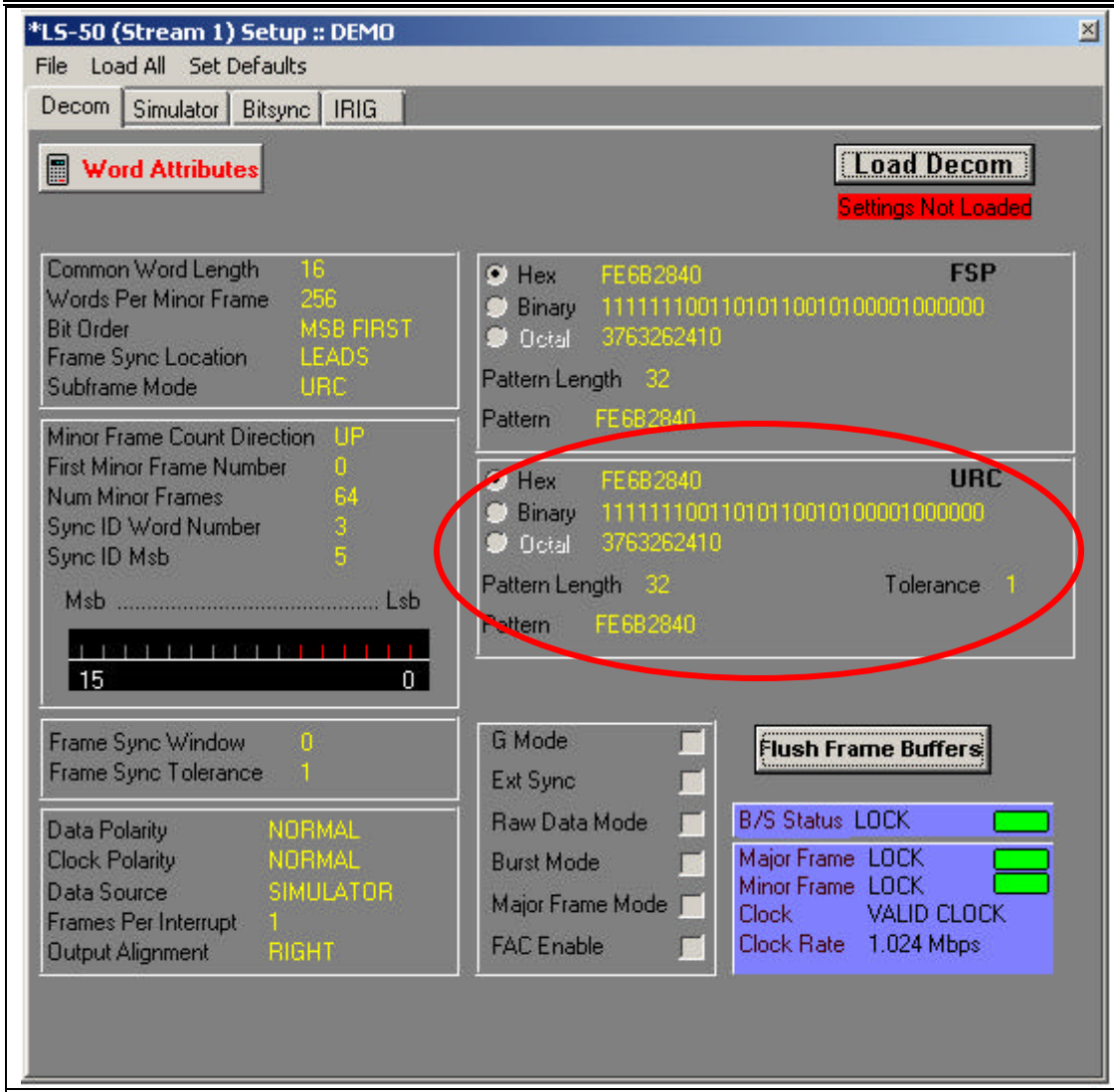
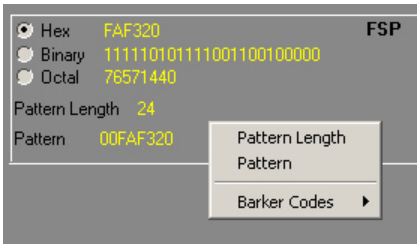



**Figure 3-5 Unique Recycle Code Variation of the Decom Setup Tab**

### 3.1.1.3 Frame Synchronization Pattern



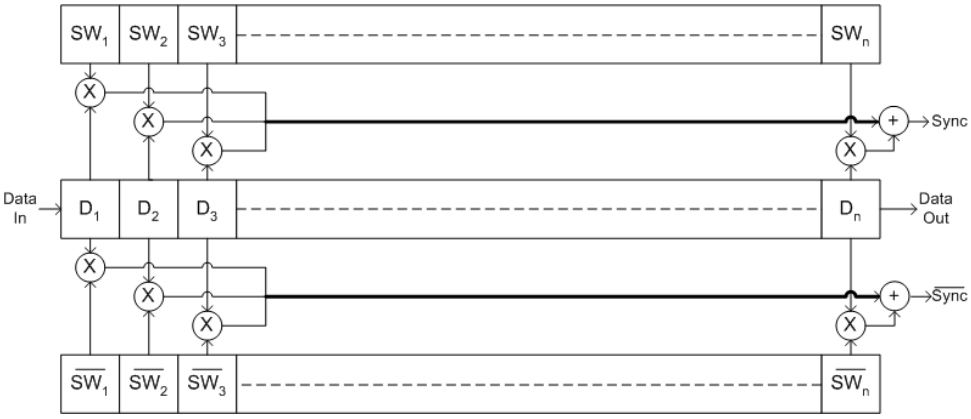

The frame synchronization pattern parameters include: the actual Pattern and the Pattern Length. The user may enter the actual pattern in a variety of different format representations including Hexadecimal (HEX), Binary and Octal. If the synchronization pattern is to contain "don't care bits," then the pattern must be entered in binary As mentioned previously, the frame synchronization pattern is a unique binary bit pattern used to indicate the beginning of a telemetry minor frame. To achieve this, a frame synchronizer is employed with correlator & state machine circuitry that recognizes unique bit patterns indicating the beginning of minor

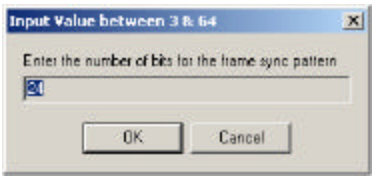
frame data. The frame synchronizer typically "searches" for patterns, "checks" for the recurrence of the pattern in the same position for several frame periods, and then "locks" on the pattern.

***Definition:***

- **Correlator** – Logic circuit (see below) used to detect the presence of a frame synchronization pattern used to identify the beginning of a minor frame.

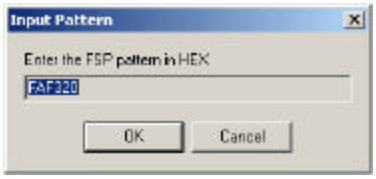
The synchronization strategy is to pass the incoming data stream into a correlator which checks each bit of the input stream against a predefined synchronization pattern. In the correlator, the data is passed through a shift register, the contents of which are bitwise compared with the predefined pattern once each bit period. When the summation output of the correlator exceeds a preset threshold, the sync pattern is declared to have been found. Optimal codes for the sync pattern are chosen because they have low correlation unless the code pattern is exactly aligned with the desired pattern.

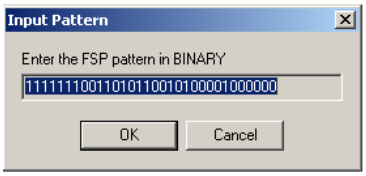


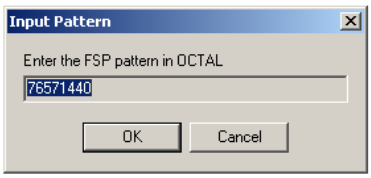




To enter the required frame synchronization pattern, the user must first invoke the **Pattern Length** command to specify the bit length of the frame sync pattern. For the LS-50, the length of the pattern may be up to 64-bits. After entering the number of bits for the frame sync pattern, the appropriate Barker code pattern will automatically be filled in on the input pattern dialog box. This feature is based on the number of bits entered for the pattern length (only for lengths of bits 7 through 32 bits will this occur). Then the user must select one of the Hexadecimal (HEX), Binary or Octal format representation radio buttons. The selected radio button will determine the appearance of the input pattern dialog box when the **Pattern** command is invoked. Note

that if the pattern length is NOT an even multiple of eight (8), then the "Octal" radio button will be grayed out. Also, if the pattern length is not an even multiple of four (4), then the "HEX" radio button will be grayed out.



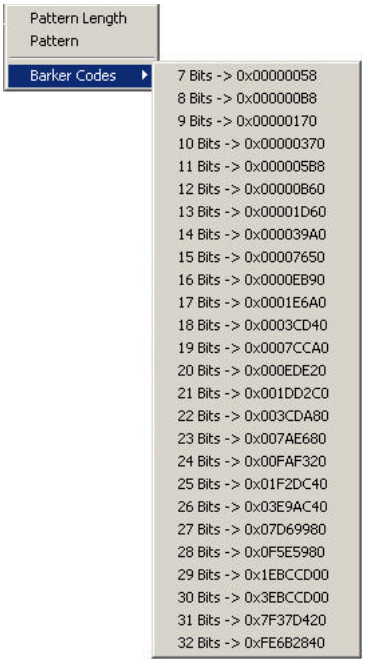




If the user wishes to use a pattern other than the one automatically selected based on the pattern length, then the pattern command should be invoked and a different pattern should be entered.

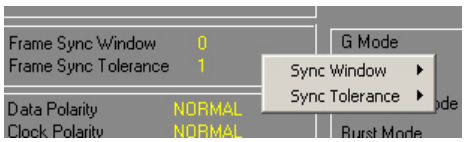
| | |
|---|---|
|  | Note – Per the IRIG-106, it is recommended that for optimal results, the frame synchronization pattern should be at least 16-bits in length[3]. *(24 or 32 bits would be much better).* In the LS-50, the pattern may be up to 64-bits in length |

As previously mentioned, optimal codes for the sync pattern should be chosen because they have low correlation properties unless the code pattern is exactly aligned with the desired pattern. To aid the user in selecting the appropriate pattern, invoke the **Barker Codes** command for a convenient list of some possible sync patterns. Note that choosing a pattern form the popup list does not "enter" the pattern – that still must be done via the **Pattern** command.




### 3.1.1.4 Frame Sync Sensitivity Parameters

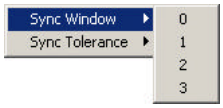
The frame synchronization sensitivity parameters include: the Sync Window and Sync Tolerance commands. Both of these commands relate to how well the frame synchronization process functions in a noisy, real world environment.




[3] J. L. Maury, Jr. and J. Styles, "Development of Optimum Frame Synchronization Codes for Goddard Space Flight Center PCM Telemetry Standards," in Proceedings of the National Telemetering Conference, June 1964.

**Statistical Measures** – The primary performance measure used in association with the frame synchronizer is; 1) the probability of falsely locking onto a random data pattern and believing it to be the real sync pattern, and 2) the probability of missing a valid sync pattern in the data stream due to an unacceptable number of bit errors.

- The probability of a false lock is only a function of the length of the chosen sync pattern, and NOT a function of the channel bit-error rate.
- The probability of missing a valid pattern is a function of both channel bit-error rate, and pattern length.




The frame synchronizer in the LS-50 typically "searches" for patterns, "checks" for the recurrence of the pattern in the *same position* for several frame periods, and then "locks" on the pattern. Because of certain peculiarities in the demodulation and bit synchronization processes for noisy channels, sometimes the recovered sync pattern may be shifted, or offset in time by one or more bit time periods. If these "bit-slips" in the recovered sync pattern are not allowed and accounted for, then the synchronization state machine will loose sync because the pattern is NOT in the exact *same position* as it was in the previous minor frame. The user specifies the number of bit-slips allowed by invoking the **Sync Window** command and entering a value of up to 3 bits. Note, in a noisy signal environment, setting the window to Zero (0) would likely result in the LS-50 NEVER acquiring or maintaining frame synchronization.


The user may specify the number of bits in the acquired sync pattern that may be different from the ideal pattern and still achieve & maintain synchronization by invoking the **Sync Tolerance** command. The user may specify that the received pattern must contain no bit errors, and would thus set the tolerance to Zero (0). In a noisy signal environment, such a setting would likely result in the LS-50 NEVER acquiring or maintaining frame synchronization. For the noisy, real world environment, the user may set the bit error tolerance from 1 to 16 bits. Some guidance on what to set the Sync Tolerance value to can be found below.

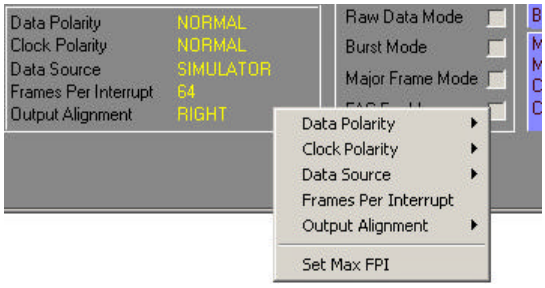
***A "Geek" Technical Tidbit:***
The probability of missing a valid sync pattern in a noisy environment….

The probability of missing a sync pattern in a data stream is directly related to the number of bit errors encountered in the channel. If the correlator allows for a number "k" or fewer bit errors (sync tolerance value) to occur in a sync pattern of length "N" bits, then the probability "P" of missing a sync pattern in a channel with a bit-error-rate of "B" is given by:

$$P = \sum_{j=k+1}^{N} \left( \frac{N!}{j!(N-j)!} \right) B^{j} (1-B)^{N-j}$$

### 3.1.1.5 Data Source Configuration

The Data Source Configuration parameters include: the Data Polarity, Clock Polarity, Data Source, Frames Per Interrupt, and Output Alignment.

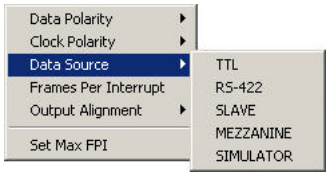




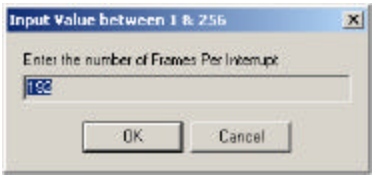
In the telemetry field, certain data transmission & demodulation schemes have inherent ambiguities that may result in the data at the decommutator input being inverted. By invoking the **Data Polarity** command, the LS-50 decommutator can be programmed by the user to accept patterns of either data polarity. The "AUTO" mode automatically inverts the incoming data if there is no frame lock and an inverted pattern is detected. This mode should probably be defaulted to unless the sync strategy is set to Frame Alternating Complement (FAC). To manually invert the incoming data, irrespective of the frame sync status, one selects the "INVERT" mode. The "NORMAL" mode leaves the polarity sense of the incoming data unchanged.

The LS-50 decommutator essentially has two basic signal input types; Clock, and Data. By using the **Clock Polarity** mode, the user may select either polarity sense of the input clock. In essence, the clock polarity mode allows the user to select either the rising or falling edge of the clock to latch incoming data into the

decommutator. For the rising edge, select "NORMAL." For the falling edge, select "INVERT."




The LS-50 decommutator has five sets of data and clock inputs, and the user may select from these by invoking the **Data Source** command. The inputs that may be selected include: TTL, RS-422, Slave, MEZZANINE, and SIMULATOR. For a single-ended clock/data input, select "TTL." For a differential clock/data input, select RS-422. For applications involving an onboard LS-40 bit synchronizer, select "MEZZANINE." For applications involving embedded asynchronous streams and a second on-board LS-50 decommutator, select "SLAVE." For development and testing applications, select "SIMULATOR." This will allow the decommutator to be driven by a known & controlled source of data. For more detailed information on the nature of the TTL, RS-422, and Slave input clock/data signals, see paragraph 2.4 on page 7.



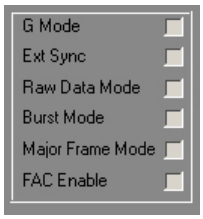

The LS-50 decommutator can be used with extremely large frame formats (16,383 words per minor frame) and contains dual ping-pong data output buffers, each with 128K bytes of memory. The output of the decommutator is a stream of words from the input, with a header prefixed to the beginning of each minor frame. This data is grouped into "blocks" of one or more minor frames and written to the on-board buffer memory. Two such ping-pong buffers are provided. Normally while the decommutator writes to one ping-pong buffer, the other is accessible for use. When a block's worth of data has been written, an interrupt is generated and the two buffers are logically switched so that fresh data becomes available. The user may control the number of minor frames that make up the ping-pong buffer by invoking the **Frames Per Interrupt** command. For optimal results, the user should set the frames per interrupt value to some multiple of the minor frames per major frame size. The LS-50 can support up to 256 frames per interrupt, depending on the frame size. Note, for fast streams, the user should maximize the number of frames per interrupt to reduce the load on the CPU. If the user is unsure what to set the frames per interrupt value to, the **Set Max FPI** (Frames Per Interrupt) command may be invoked to set the maximum number of frames per interrupt based on the minor frame size and the amount of memory on the card.

| | |
|---|---|
|  | ***Note:*** The number of minor frames per interrupt cannot exceed:<br>a. 256<br>b. ((words-per-minor frame + 5) * 2 * frames per interrupt) cannot exceed 131,072 bytes.<br>*a.* (words-per-minor frame + 5) cannot exceed 16,383 words. |


To select left-justified or right-justified output data from the decommutator, the user may invoke the **Output Alignment** command. Note: The output alignment should always be set to Right Aligned, with the possible exception of connecting the LS-50 to a LS-71 DAC[4]. In general, if left alignment is selected, then the processing overhead of LDPS will be increased, because part of the 'normalization' process involves the right alignment of all the data prior to sending it off to the client or processing tasks such as audio or video, etc.

| | |
|---|---|
| | ***Recommendation:***<br>A good rule of thumb - If the minor frame rate is 50 Hz or less, then set the FPI to 1. If it is more, then set it to the number of minor frames per major frame, if it will fit. Otherwise the user will have to experiment with FPI numbers between 1 and the minors-per-major (ideally a multiple of minors-per-major). |

### 3.1.1.6 Decom Mode Check Boxs




The LS-50 decommutator setup tab has a number of mode selection check boxes that include: G Mode, External Sync, Raw Data Mode, Burst Mode, Major Frame Mode, and FAC Enable.

Normally the decommutator output stops when it loses minor frame lock. If **G Mode** is checked, the decommutator will continue to processes incoming bits into "frames" and output them. If it detects a sync pattern while in this state, it will abort the frame it is assembling, and start a new buffer. Basically, the G Mode tells the decom to try to lock onto the frame sync pattern, but even if it cannot, it collects the buffer of data and generates an interrupt even if there is no frame lock.

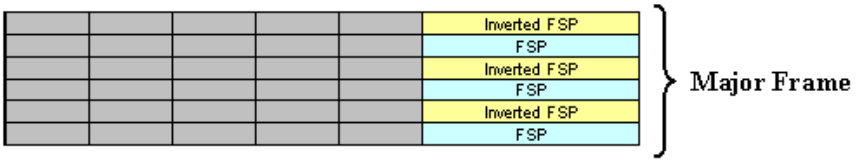
To support fixed length frames that arrive at irregular intervals, the user may check the **Burst Mode** box. Check this box if the incoming data consists of fixed-length frames separated by zero or more fill bits. The data in the frames will be output and the fill bits will be discarded.

[4] If the bits-per-word is greater than 14, then left alignment may come into play as a possible requirement, depending on what resolution the DAC output is using. If the data is right aligned and bits-per-word is 16, then the two LSBs on the DAC output will be lost. If the data is left aligned, then the two MSBs on the DAC output will be lost.

***Definition:***

- **FAC** – A less commonly used subframe synchronization method that is a variant of the FCC mode is called Frame Alternating Complement (FAC).

In this method, the frame synchronization pattern is alternated with the complement of the frame synchronization pattern.




The **Ext Sync** mode instructs the decommutator to establish the "lock" condition based upon an external sync pulse signal only. This mode bypasses the internal frame synchronizer (correlator/state machine) in favor of an external signal provided by the user.

The **Raw Data Mode** instructs the decommutator to ignore the frame lock state (i.e., don't look for a frame sync pattern) and just ingest the correct number of bits and generate an interrupt. This mode is used to record 100% of the input bits, regardless of lock state.

When selected, the **Major Frame Mode** will generate an interrupt only when a complete major frame of data has been gathered and the decom is in major and minor frame lock. Note, in this mode the frames per interrupt is fixed to the number of minor frames.

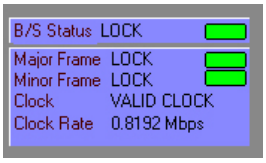
The **FAC Enable** mode is used to enable the Frame Alternating Complement subframe synchronization method. As discussed above, the FAC mode is a variant of the FCC subframe synchronization method.

#### 3.1.1.7 Flush Frame Buffers Button

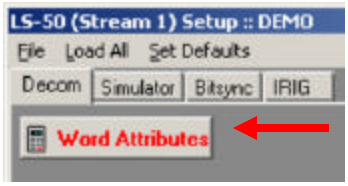
When this button is pressed, data is flushed out of the decommutator's buffer. The result is an interrupt, regardless if the buffer is filled or not.

#### 3.1.1.8 Decom Status Displays

The LS-50 decom setup tab has a window display showing the status of some of the LS-50's functional states. These states include: bit synchronizer signal lock, major and minor frame lock, a valid clock indication, as well as the clock rate in Mbps. This status display is updated at a ten-hertz rate and is common to all LS-50 function setup tabs.



### 3.1.1.9 Decommutator Word Attributes




The **Word Attributes** button directly below the Decom tab allows the user to make individual exceptions to the definitions established in the Major Frame Configuration section of the Decom tab (see paragraph 3.1.1.1 on page 21). The word attributes include: word length, bit order, and master/slave status. The word attributes dialog box is shown in Figure 3-6 below. To modify the word attributes of a particular word in the minor frame, navigate using the scroll bar at the bottom of the window and select a word by clicking on the middle of the column. Right clicking will invoke the attributes menu as shown in the figure below (red oval). To select a contiguous group of words, select the first word, then shift-click on the last word to select the group. To select a noncontiguous set of words, select the first word, and then control-click on each subsequent word until all words are selected. After the words are selected, right click to invoke the attributes menu.

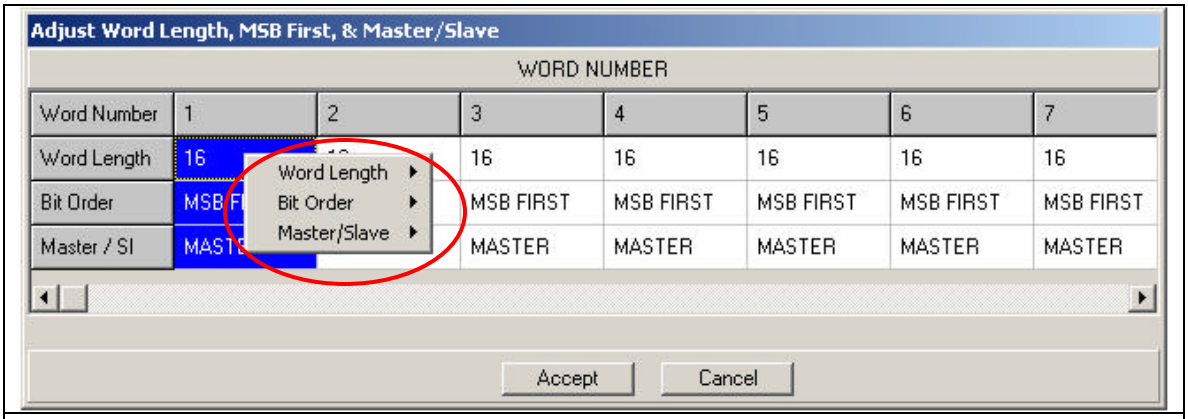


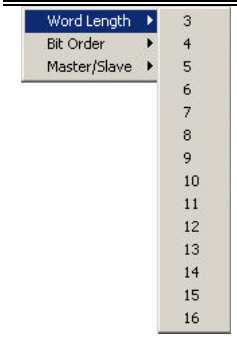

**Figure 3-6 LS-50 Decom – Word Attributes Setup**

***Definitions:***

- **Subcommutated** – A parameter sent at a rate less than or equal to the minor frame rate, with each parameter appearing at a fixed subframe location.
- **Subframe** – Corresponds to a column within a major frame.
- **Super-Subcommutated** – A subframe parameter that appears more than once per minor frame.

| FSP | SFID | SubCom1 | SuperSubCom1 | Prime1 | SuperSubCom1 |
|---|---|---|---|---|---|
| FSP | SFID | SubCom2 | SuperSubCom2 | Prime1 | SuperSubCom2 |
| FSP | SFID | SubCom1 | SuperSubCom1 | Prime1 | SuperSubCom1 |
| FSP | SFID | SubCom2 | SuperSubCom2 | Prime1 | SuperSubCom2 |
| FSP | SFID | SubCom1 | SuperSubCom1 | Prime1 | SuperSubCom1 |
| FSP | SFID | SubCom2 | SuperSubCom2 | Prime1 | SuperSubCom2 |

**Major Frame**



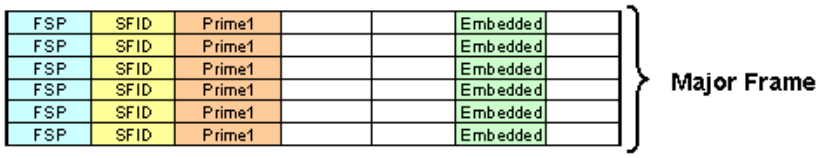

The **Word Length** command may be used to set the length of selected words from 3 to 16 bits in length. The user might invoke this command because not all words in a minor frame need be of the same length. For example, the common words in a minor frame could be 8-bits in length. However, several of the words might be 14 or 16 bits in length and would be individually specified using this command.

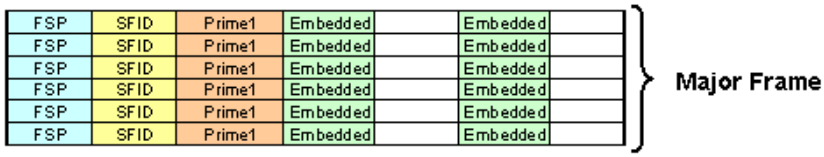

By invoking the **Bit Order** command, the user specifies for the selected words of the minor frame whether the Most Significant Bit (MSB) is first, as read from left to right, or the Least Significant Bit (LSB) is first, again, read from left to right. The user might invoke this command because not all words in a minor frame need have the same bit order. For example, the common words in a minor frame could have LSB first bit order. However, several of the words might be MSB first and would be individually specified using this command.


For telemetry formats that involve embedded asynchronous frames, and the use of a second on-board LS-50 decommutator, the user may specify the location of the embedded words by invoking the **Master/Slave** command and selecting the "SLAVE" mode. Thus selected, whenever any of the embedded words are encountered by the primary decommutator, they are serially redirected out of the decom via the "slave port." The slave port is a serial output (clock & data) that drives a second on-board LS-50 decommutator. The embedded words may be prime commutated, or super-commutated within the minor frame. The default mode for all common words in the minor frame is "MASTER."

***Definition:***

- **Embedded Asynchronous Frame** – Literally, one telemetry stream embedded within the frame structure of another, where the embedded words are at fixed locations within the primary minor frame. The LS-50 can support multiple embedded asynchronous streams using either a second hardware decommutator, and/or a software decommutator (see the LDSP user's manual for more information). The embedded words may be prime commutated, or super-commutated within the minor frame as shown below. The embedded stream is said to be "asynchronous," because there is often no definable temporal relationship between the synchronization marker of the embedded stream and the synchronization marker of the primary minor frame. More specifically, the location of the sync marker and SFID of the embedded frame are often not the same from one major frame to the next. The asynchronous nature of the embedded stream also implies that there is no bit alignment between the words of the embedded stream and the words of the primary stream. For example, bit-1 (leftmost) of the frame sync pattern of the embedded stream could be located in the middle of the second embedded word in the first major frame, and reoccur again in the second to last bit of the fifth embedded word of the next major frame, and so on…



***Definitions:***

- **Commutated** – A parameter sent once per minor frame and located in the same location in each minor frame relative to the synchronization marker. (Also called "Prime" Commutated)
- **Supercommutated** – A parameter sent at a sampling rate that is an integer factor greater then the minor frame rate, with each appearance of the parameter at a fixed location relative to the synchronization marker of the minor frame. (Note, the number of appearances of a supercommutated parameter within each minor frame is NOT fixed by the IRIG-106 standard).

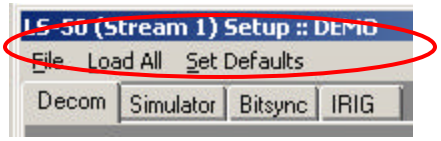
| | | | | | |
|---|---|---|---|---|---|
| FSP | SFID | Prime1 | SuperCom1 | Prime2 | SuperCom1 |
| FSP | SFID | Prime1 | SuperCom1 | Prime2 | SuperCom1 |
| FSP | SFID | Prime1 | SuperCom1 | Prime2 | SuperCom1 |
| FSP | SFID | Prime1 | SuperCom1 | Prime2 | SuperCom1 |
| FSP | SFID | Prime1 | SuperCom1 | Prime2 | SuperCom1 |
| FSP | SFID | Prime1 | SuperCom1 | Prime2 | SuperCom1 |

**Major Frame**

### 3.1.1.10 Load Decom Button


The Decom setup tab has a button control to load the setup information entered by the user. Changes made with any of the controls will not take affect until this button is pressed. The user may load all four major functions (Decom, Simulator, Bitsync, and IRIG) from the "Load All" command on the menu next to the File menu). If any changes are made to the decom setup without loading, a red text will appear below the Load button (shown above right), indicating the displayed data does not match the cards' loaded data.

### 3.1.1.11 Saving the Decommutator Setup Configuration




Below the window header of the "LS-50 (Stream 1) Setup" display shown in Figure 3-3 on page 18 are the *"File," "Load All,"* and *"Set Defaults"* commands. After the decom setup configuration is complete, save the settings by invoking the *"File ? SaveAs"* command. To download all of the configurations (decom, simulator, Bitsync, and IRIG) to the LS-50 hardware, invoke the *"Load All"* command. To recall a previously defined LS-50 setup configuration, invoke the

*"File ? Recall"* command and select the appropriate file from the file menu and then download the configuration to the LS-50 hardware by invoking the *"Load All"* command. To set the LS-50 hardware to its default state, invoke the *"Set Defaults"* command.

### 3.1.2 The LS-50 Simulator Tab

The LS-50 simulator setup tab and its associated menus and controls are shown in Figure 3-7 below. The LS-50 simulator may be used to drive the decommutator in a self-test or frame definition scenario, or it may be used independently to create PCM data streams not intended for the on-board decommutator.

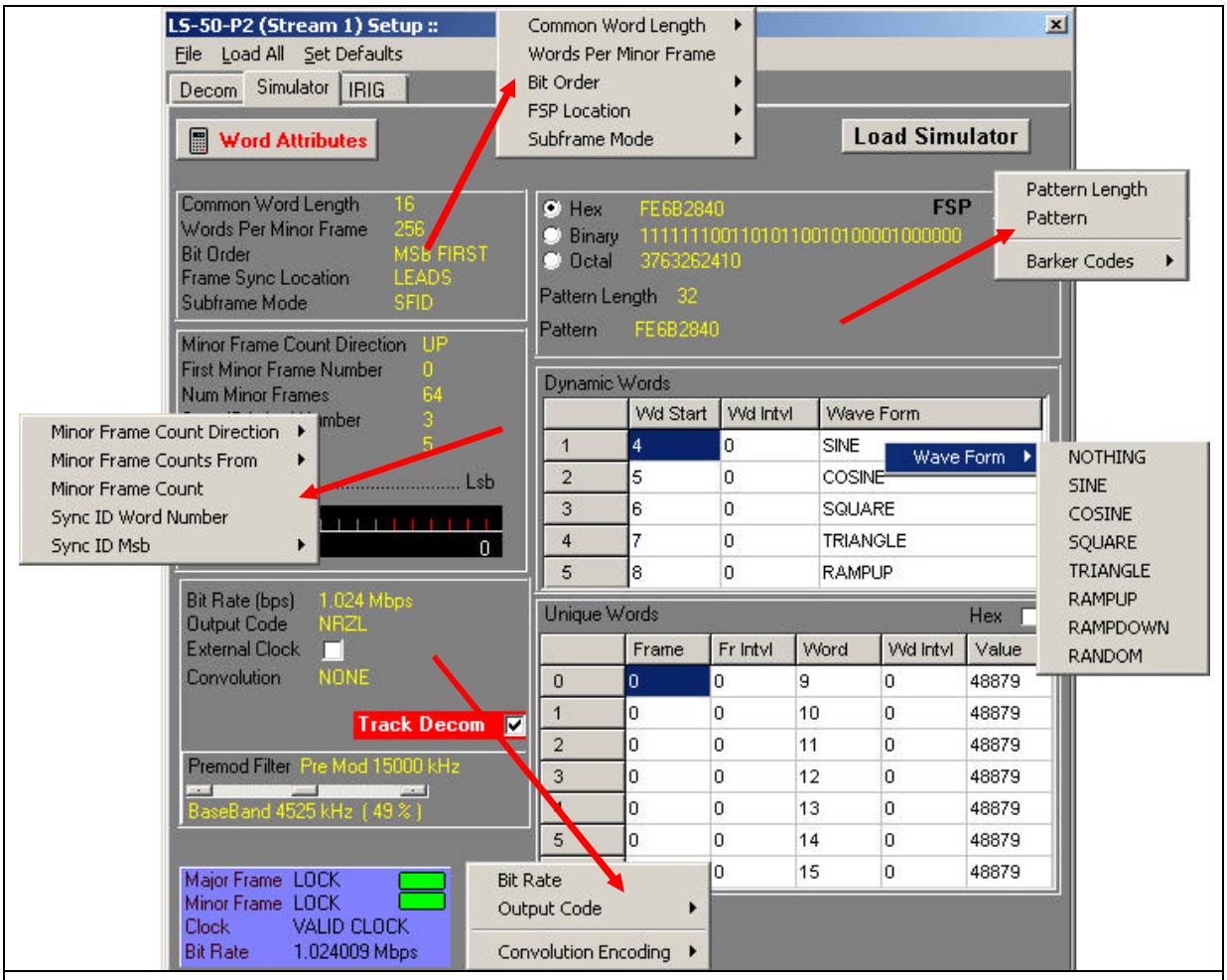


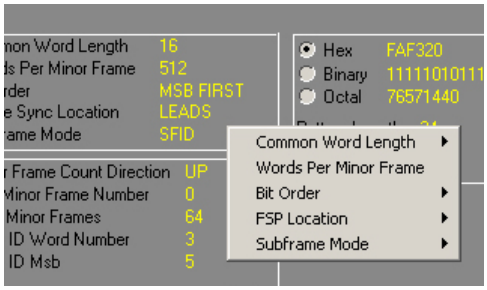

**Figure 3-7 The LS-50-P Simulator Configuration Menus**

There are up to nine groups of controls displayed for the simulator, depending on the setting of other controls.

The nine control groups of the LS-50 simulator tab include:

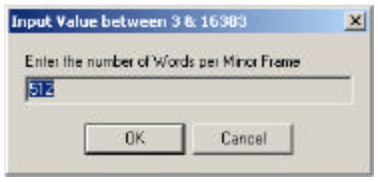
- Major Frame Configuration
- Minor Frame Configuration
- Frame Synchronization Pattern
- Clock & Output Coding Configuration
- Data Source Configuration
- Pre-modulation Filter Configuration
- Dynamic Word Configuration
- Unique Words Configuration
- Word Attributes Control

### 3.1.2.1 Major Frame Configuration




The major frame configuration consists of five controls/parameters that include: common word length, the number of words per minor frame, the bit order of the words in the frame, the frame synchronization patter location, and the subframe synchronization mode.


The **Common Word Length** may be set from 3 to 16 bits in length. The common word length defines the length in bits of the *majority* of words that make up a minor frame. Note, not all words in a minor frame need be of the same length. For example, the majority of the words in a minor frame could be 8-bits in length, and thus the common word length would be 8. However, several of the words might be 14 or 16 bits in length and would be individually specified using the Simulator Word Attributes command function described in paragraph 3.1.2.9 on page 46.



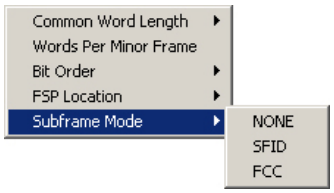

The minor frame length is defined by the user by invoking the **Words Per Minor Frame** command. Here, the user enters the number of words (of length specified by common word length) that make up a minor frame. The minor frame length on the LS-50 may be between 3 and 16,383 words.

By invoking the **Bit Order** command, the user specifies for the common words of the minor frame whether the Most Significant Bit (MSB) is first, as read from left to right, or the Least Significant Bit (LSB) is first, again, read from left to right. Note,

not all words in a minor frame need have the same bit order. For example, the majority of the words in a minor frame could have LSB first bit order. However, several of the words might be MSB first and would be individually specified using the Simulator Word Attributes command function described in paragraph 3.1.2.9 on page 46.


The user specifies the location of the FSP by invoking the **FSP Location** command, and selecting "TRAILS" or "LEADS."

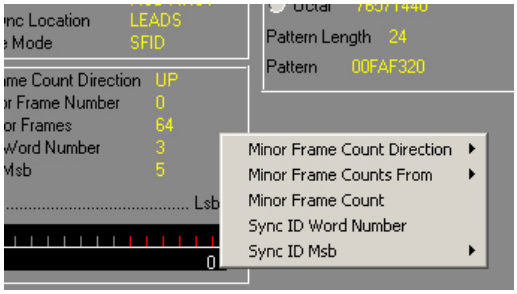
To implement a subframe synchronization scheme, telemetry designers often add one or more "special" words to each minor frame. These special words are used by the frame synchronizer state machine to establish the location of the first minor frame in the major frame. The LS-50 supports three subframe synchronization modes: SFID, FCC, and URC.



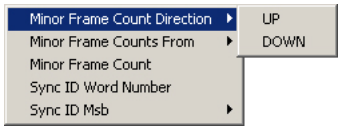

The user specifies the method of subframe synchronization by invoking the **Subframe Mode** command, and selecting "None," "SFID," or "FCC."

### 3.1.2.2 Minor Frame Configuration

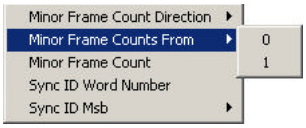
The minor frame configuration consists of five controls/parameters that include: Minor Frame Count Direction, Minor Frame Counts From, Minor Frame Count, Sync ID Word Number, and Sync ID MSB.

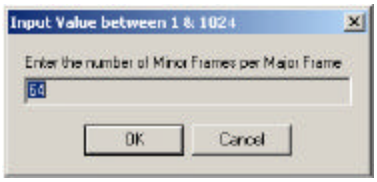



As mentioned previously, in the SFID mode, the synchronization pattern occupies one or more words in each minor frame and acts as a counter. The user may specify whether the pattern value increments or decrements from minor frame to minor frame by invoking the **Minor Frame Count Direction** command.

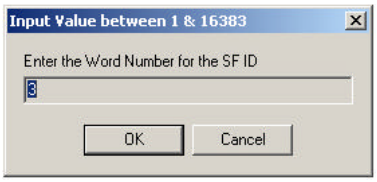



In some telemetry frame designs, the subframe counter in minor frame-0 will initially begin counting from a starting value of zero (0), while in other frame designs, the subframe counter will begin counting from a starting value of one (1). The user specifies one or the other of these two conditions by invoking the **Minor Frame Counts From** command.

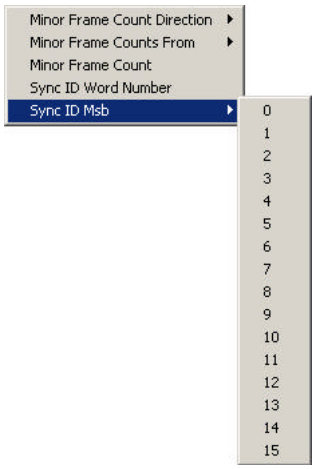




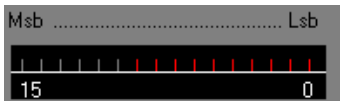

As mentioned previously, the major frame is composed of an integer number of minor frames, and the minor frame is a fixed length block of data sub-divided into an integer number of fixed-length words. By invoking the **Minor Frame Count** command, the user may specify the number of minor frames that make up the major frame. The LS-50 can support up to 1024 minor frames per major frame.




The location of the subframe identification (SFID) word(s) is arbitrary within the minor frame and may be specified by the user by invoking the **Sync ID Word Number** command. As the LS-50 can support up to 16,383 words per minor frame, the user may thus locate the SFID word anywhere within this range, provided it does not overlap or coincide with the frame synchronization pattern location.

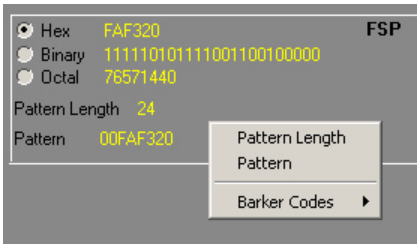



As described previously, the SFID word is used as a counter, but it is not always the case that ALL of the bits in the SFID word are used for this purpose. For example, the SFID word might be 16-bits in length, but there might only be 512 minor frames in the major frame. In this scenario, a 9-bit counter ($2^9 = 512$) would be required and the user would specify the location of the counter within the larger 16-bit word by invoking the **Sync ID Msb** command and selecting the appropriate bit position for the most significant bit of the SFID counter. The Sync ID Msb is represented graphically in the minor frame configuration section as shown right.

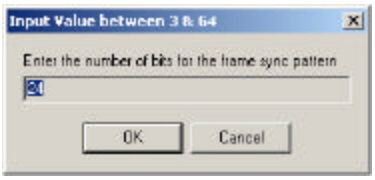



### 3.1.2.3 Frame Synchronization Pattern

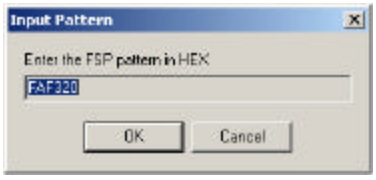
The frame synchronization pattern parameters include: the actual Pattern and the Pattern Length. The user may enter the actual pattern in a variety of different format representations including Hexadecimal (HEX), Binary and Octal. If the synchronization pattern is to contain "don't care bits," then the pattern must be entered in binary as mentioned previously, the frame synchronization pattern is a unique binary bit pattern used to indicate the beginning of a telemetry minor frame. To achieve this, a frame synchronizer is employed with correlator & state machine circuitry that recognizes unique bit patterns indicating the beginning of minor frame data. The frame synchronizer

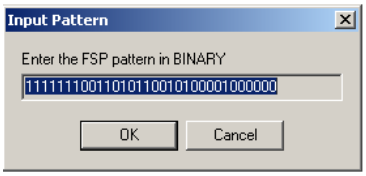


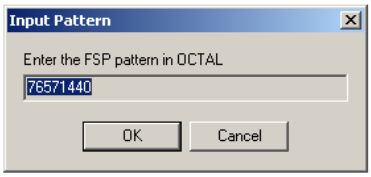
typically "searches" for patterns, "checks" for the recurrence of the pattern in the same position for several frame periods, and then "locks" on the pattern.




To enter the required frame synchronization pattern, the user must first invoke the **Pattern Length** command to specify the bit length of the frame sync pattern. For the LS-50, the length of the pattern may be up to 64-bits. After entering the number of bits for the frame sync pattern, the appropriate Barker code pattern will automatically be filled in on the input pattern dialog box, based on the number of bits (only for number of bits 7 through 32 will this occur). Then the user must select one of the Hexadecimal (HEX), Binary or Octal format representation radio buttons. The selected radio button will determine the appearance of the input pattern dialog box when the **Pattern** command is invoked. Note that if the pattern length is NOT an even multiple of eight (8), then the "Octal" radio button will be grayed out.



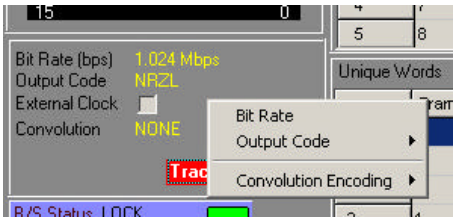



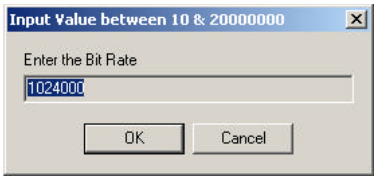

If the user wishes to use a pattern other than the one automatically selected based on the pattern length, then the Pattern command should be invoked and a different pattern should be entered.

As previously mentioned, optimal codes for the sync pattern should be chosen because they have low correlation properties unless the code pattern is exactly aligned with the desired pattern. To aid the user in selecting the appropriate pattern, invoke the **Barker Codes** command for a convenient list of some possible sync patterns. Note that choosing a pattern form the popup list does not "enter" the pattern – that still must be done via the **Pattern** command.

### 3.1.2.4 Clock & Data Output Mode Configuration

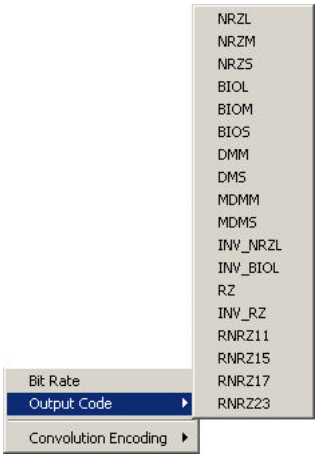
The Clock and Data Output Mode controls include: the output bit rate (bits/second), the output encoding format, and the Forward Error Correction (FEC) coding mode. Invoking **Bit Rate** allows the user to specify the output bit rate (bits/second) of the PCM encoder on the simulator. The user may enter a value between 10 bps to 20 Mbps for NRZ codes, and 10 bps to 10 Mbps for all other codes. By invoking the **Output Code** command, the user may select from a variety of possible PCM output codes, some of which are shown graphically



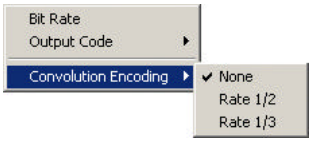



in Figure 3-8 on page 43. The PCM output codes fall into several general classes including: Non-Return to Zero (NRZ)

codes, self-clocking codes such as Bi-Phase, and Miller, and Randomized codes. NRZ codes are the most commonly used but are occasionally problematic if they are not well behaved[5]. Ill-behaved data streams may be mitigated by using a self-clocking code such as Bi-Phase or and Miller, but with the added penalty of doubling the required channel bandwidth. Randomized codes do not require twice the bandwidth to transmit, but in a worst-case scenario, their use can triple the received bit-error-rate.




Bit-error-rate issues in telemetry systems are often alleviated by using Forward Error Correction schemes such as **Convolution Encoding** of the data. A convolutional code is a type of error-correcting code often used to improve the performance of a radio or satellite link. The LS-50 can support rate ½, and rate ? convolutional codes as well as non-FEC encoded data. In general, if a convolutional code is said to be rate ½, this means that for every input data bit, the encoder will produce two output code symbol bits. For rate ?, every input data bit will produce three output code symbol bits. Thus, employing this type of FEC scheme in a telemetry system will double or triple the transmitted channel data rate. (*There is no free lunch in telemetry engineering!*)



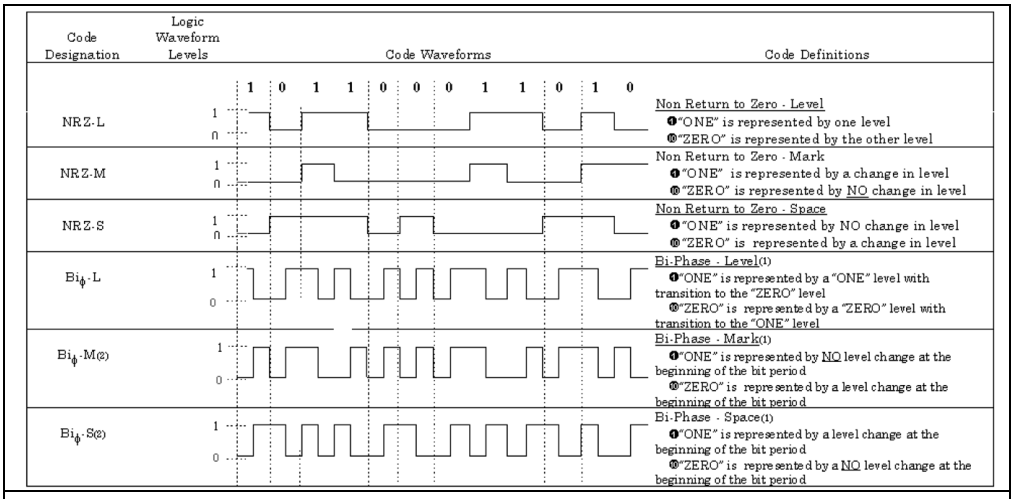
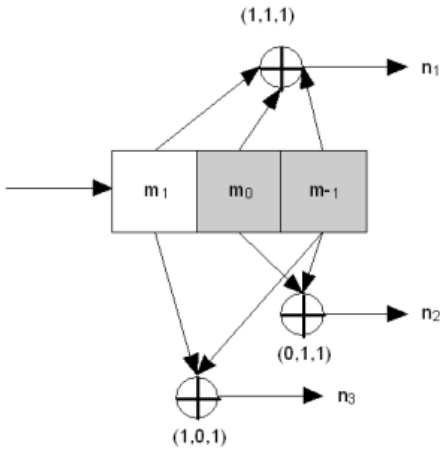



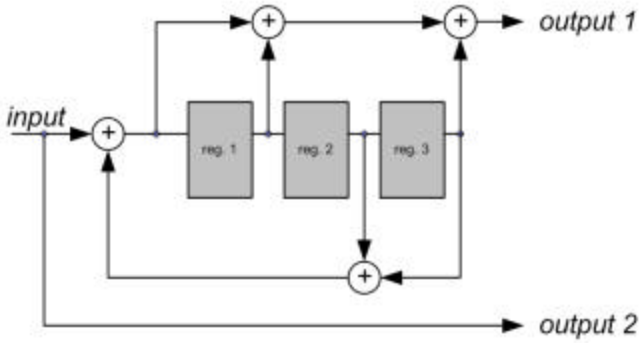

**Figure 3-8 PCM Code Definitions**

[5] An NRZ data stream is said to be "ill-behaved" if its spectrum has strong DC components caused by long strings of ones or zeros. Bit synchronizers have great difficulty locking onto ill-behaved signals.

Several algorithms exist for decoding convolutional codes. For relatively small constraint length values, the Viterbi algorithm is universally used as it provides maximum likelihood performance and is highly parallelizable. Viterbi decoders are thus easy to implement in VLSI or FPGA hardware. An especially popular Viterbi-decoded convolutional code, used on the Voyager program has a constraint length of 7 and a rate of ½.



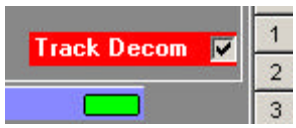

Rate 1/3 non-recursive, non-systematic convolutional encoder with constraint length 3




Rate 1/2 recursive, systematic convolutional encoder with constraint length 4

**Figure 3-9 Some Examples of Convolutional Encoder Circuits**

#### 3.1.2.4.1 Linking the Simulator and Decommutator Configurations



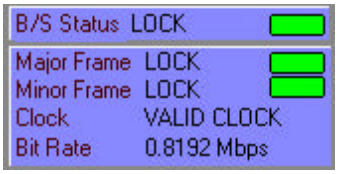

The LS-50 simulator may be used to drive the decommutator in a self-test or frame definition scenario, or it may be used independently to create PCM data streams not intended for the on-board decommutator. When they are used together, the user may click the **Track Decom** checkbox. This convenience will link the major and minor frame configurations entered for the decommutator with the simulator. When unchecked, the major and minor frame configurations of the simulator may be entered independently of the decommutator.

#### 3.1.2.5 Pre-modulation Filter Configuration


The baseband output of the LS-50 has a programmable pre-modulation filter that is useful in shaping the waveform of the signal prior to modulation by an external FM or multi-mode modulator. The pre-modulation filter has eight user selectable cut-off frequencies that include: 250, 500, 1000, 3000, 6000, 9000, 12000, and 15000 KHz. The pre-modulation filter also has an output amplitude slider control that allows the user to adjust the output amplitude of the filter from 0 to 100% (at 100%, the output is 2 Vpp).

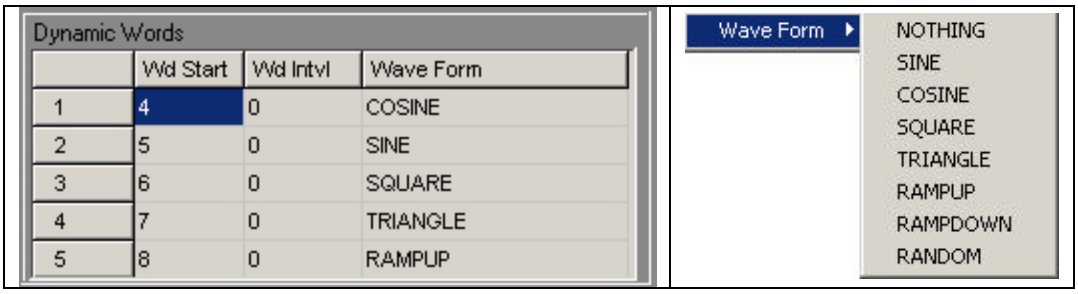
### 3.1.2.6 Status Displays

The LS-50 Simulator setup tab has a window display showing the status of some of the LS-50's functional states. These states include: bit synchronizer signal lock, major and minor frame lock, a valid clock indication, as well as the clock rate in Mbps. This status display is updated at a ten-hertz rate and is common to all LS-50 function setup tabs.




### 3.1.2.7 Dynamic Words Setup

The PCM simulator in the LS-50 may be programmed to generate dynamic data for up to five (5) words in every minor frame at the same location. For each **Dynamic Word**, the user may select from one of seven mathematical functions as shown below right.




To configure a Dynamic Word, highlight the value in the "Wd Start" cell and enter the word number. To disable a dynamic word, set the "Wd Start" cell value to "–1." Commutation of the dynamic word is set via the "Wd Intvl" cell value. If the dynamic word is to be Prime commutated, then set the "Wd Intvl" cell value to Zero. If the dynamic word is to be Super-commutated, then set the "Wd Intvl" cell value to the required increment value. To define the mathematical function that will determine the value of the dynamic word, place the cursor in the "Wave Form" cell and right-click to review the menu of functions shown above right. Select the function from the list.

### 3.1.2.8 Unique Words Setup

The PCM simulator in the LS-50 may be programmed to generate static data for up to seven (7) words in every minor frame at the same location(s). For each **Unique Word**, the user may select the minor frame number, the frame interval, the word number within the minor frame, the word interval, and finally the word value. To disable a unique word, set the "Frame" cell value to "–1" and the "Word" cell value to "–1." To display the word value in Hexadecimal, click the "Hex" checkbox. In the upper right of the Unique Words display.

Unique Words Hex

| | Frame | Fr Intvl | Word | Wd Intvl | Value |
|---|---|---|---|---|---|
| 0 | 1 | 0 | -1 | 0 | 0 |
| 1 | 1 | 0 | -1 | 0 | 0 |
| 2 | 1 | 0 | -1 | 0 | 0 |
| 3 | 1 | 0 | -1 | 0 | 0 |
| 4 | 1 | 0 | -1 | 0 | 0 |
| 5 | 1 | 0 | -1 | 0 | 0 |
| 6 | 1 | 0 | -1 | 0 | 0 |

A wide assortment of word commutation is possible using the minor frame, frame interval, word number, and word interval values. Prime, super-commutated, subcommutated, super-subcommutated, etc. are all possible. Note: in general for both Dynamic and Unique words, they cannot be the same as the frame sync pattern or the SFID word.

### 3.1.2.9 Simulator Word Attributes


The **Word Attributes** button directly below the Simulator tab allows the user to make individual exceptions to the definitions established in the Major Frame Configuration section of the Simulator tab (see paragraph 3.1.2.1 on page 39). The word attributes include: word length, and word value. The word attributes dialog box is shown in Figure 3-10 below. To modify the word attributes of a particular word in the minor frame, navigate using the scroll bar at the bottom of the window and select a word by clicking on the middle of the column. Right clicking will invoke the attributes menu as shown in the figure below (red oval). To display the word values in Hex format, click the "View Hex" check box as shown below (yellow oval). To select a contiguous group of words, select the first word, then shift-click on the last word to select the group. To select a noncontiguous set of words, select the first word, and then control-click on each subsequent word until all words are selected. After the words are selected, right click to invoke the attributes menu.

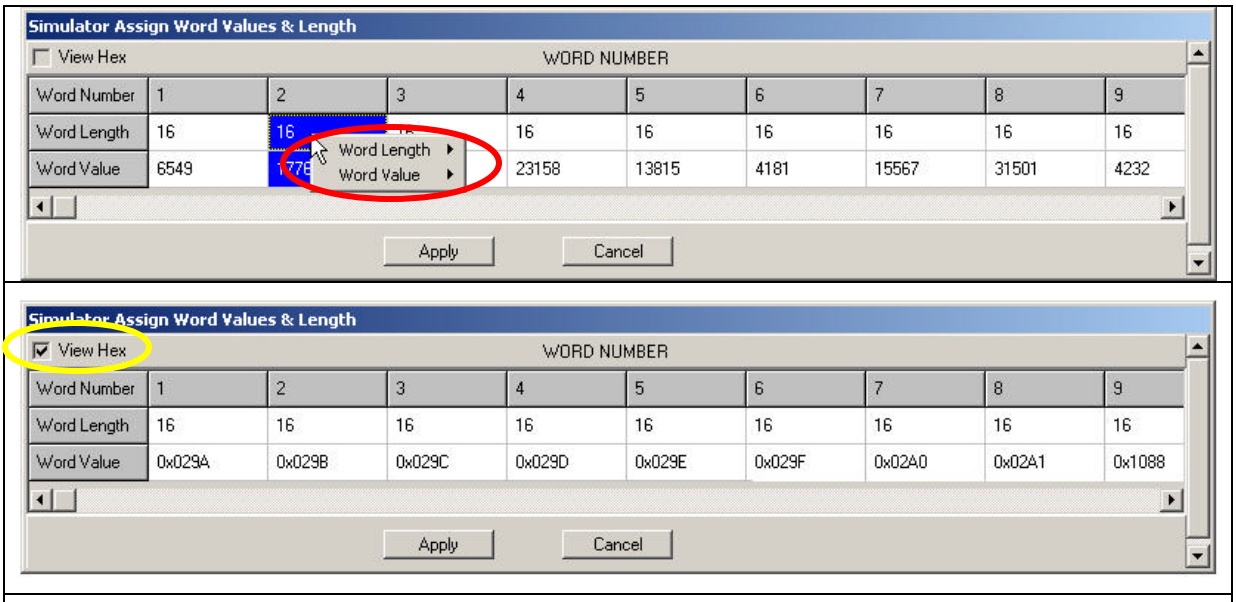
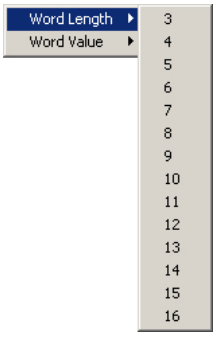
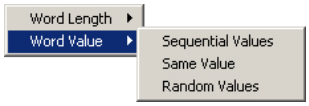


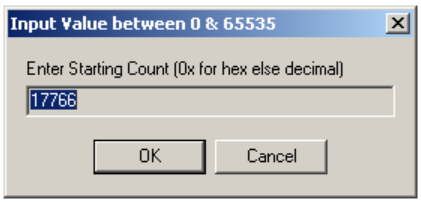

**Figure 3-10 LS-50 Simulator – Word Attributes Setup**

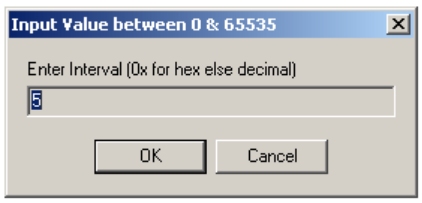



The **Word Length** command may be used to set the length of selected words from 3 to 16 bits in length. The user might invoke this command because not all words in a minor frame need be of the same length. For example, the common words in a minor frame could be 8-bits in length. However, several of the words might be 14 or 16 bits in length and would be individually specified using this command.

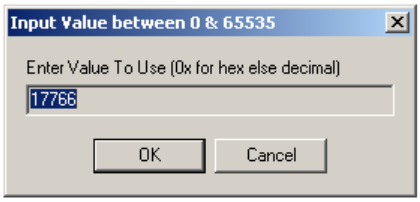



The **Word Value** command may be used to set the numerical value of individual words, or groups of words in either decimal or Hexadecimal format. By invoking *"Sequential Values,"* the user my specify an initial value and in increment value for a sequences of words. The word sequence may be contiguous or irregular. To select a contiguous group of words, select the first word, then shift-click on the last word to select the group. To select a noncontiguous set of words, select the first word, and then control-click on each subsequent word until all words are selected.



By invoking *"Same Value,"* the user may specify a common value for an individual word, or for a sequence of words. The word sequence may be contiguous or irregular. By invoking *"Random Value,"* the user may populate an individual word, or a sequence of words with random numerical values. As with the other two word value modes, the word sequence for the random values may be contiguous or irregular.

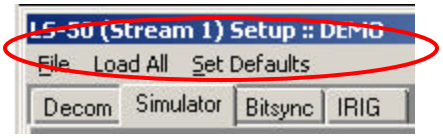



### 3.1.2.10 Load Simulator Button

The Simulator setup tab has a button control to load the setup information entered by the user. Changes made with any of the controls will not take affect until this button is pressed. The user may load all four major functions (Decom, Simulator, Bitsync, and IRIG) from the "Load All" command on the menu next to the File menu). If any changes are made to the Simulator setup without loading, a red text will appear below the Load button (shown below right), indicating the displayed data does not match the cards' loaded data.


### 3.1.2.11 Saving the Simulator Setup Configuration

Below the window header of the "LS-50 (Stream 1) Setup" display shown in Figure 3-3 on page 18 are the *"File," "Load All,"* and *"Set Defaults"* commands. After the simulator setup configuration is complete, save the settings by invoking the *"File ? SaveAs"* command. To download all of the configurations (decom, simulator, Bitsync, and IRIG) to the LS-50 hardware, invoke the *"Load All"* command. To recall a previously defined LS-50 setup configuration, invoke the *"File ? Recall"* command and select the appropriate file from the file menu and then download the configuration to the LS-50 hardware by invoking the *"Load All"* command. To set the LS-50 hardware to its default state, invoke the *"Set Defaults"* command.



***IRIG-106 "Factoid"***

- The IRIG-106 standard defines two variations of the basic telemetry frame structure. These variations are referred to as **Class-I**, and **Class-II** and are summarized below.

| **Parameter** | **Class-I** | **Class-II** |
|---|---|---|
| Bits/Minor Frame | <8192 Bits | <16,384 Bits |
| Words/Minor Frame | ≤512 Words | >512 Words |
| Minor Frame Length | Fixed | Variable |
| Fragmented Words | Not Allowed | Up to 8 |
| Format Changes | Not Allowed | Allowed |
| Asynchronous Formats | Not Allowed | Allowed |
| Bit Rates | >10 bps | >5 Mbps |
| Independent Subframe | Not Allowed | Allowed |
| SuperCom Spacing | Uniform in Minor Frame | "Anything Goes" |
| Data Format | Unsigned Binary, Complemented Binary | Others Allowed |
| Word Length | 4 to 16 Bits | 16 to 64 Bits |

### 3.1.3 The LS-50 Bit Synchronizer Tab

The LS-40-DB Bit Synchronizer setup tab and its associated menus and controls are shown in Figure 3-11 below. The View Extended Functions check box is described in detail in paragraph 3.1.3.9 on page 54. The Lumistar LS-40-DB Bit Synchronizer daughterboard provides optimal reconstruction of a serial PCM data stream that has been corrupted by noise, phase jitter, amplitude modulation, or base line variations.

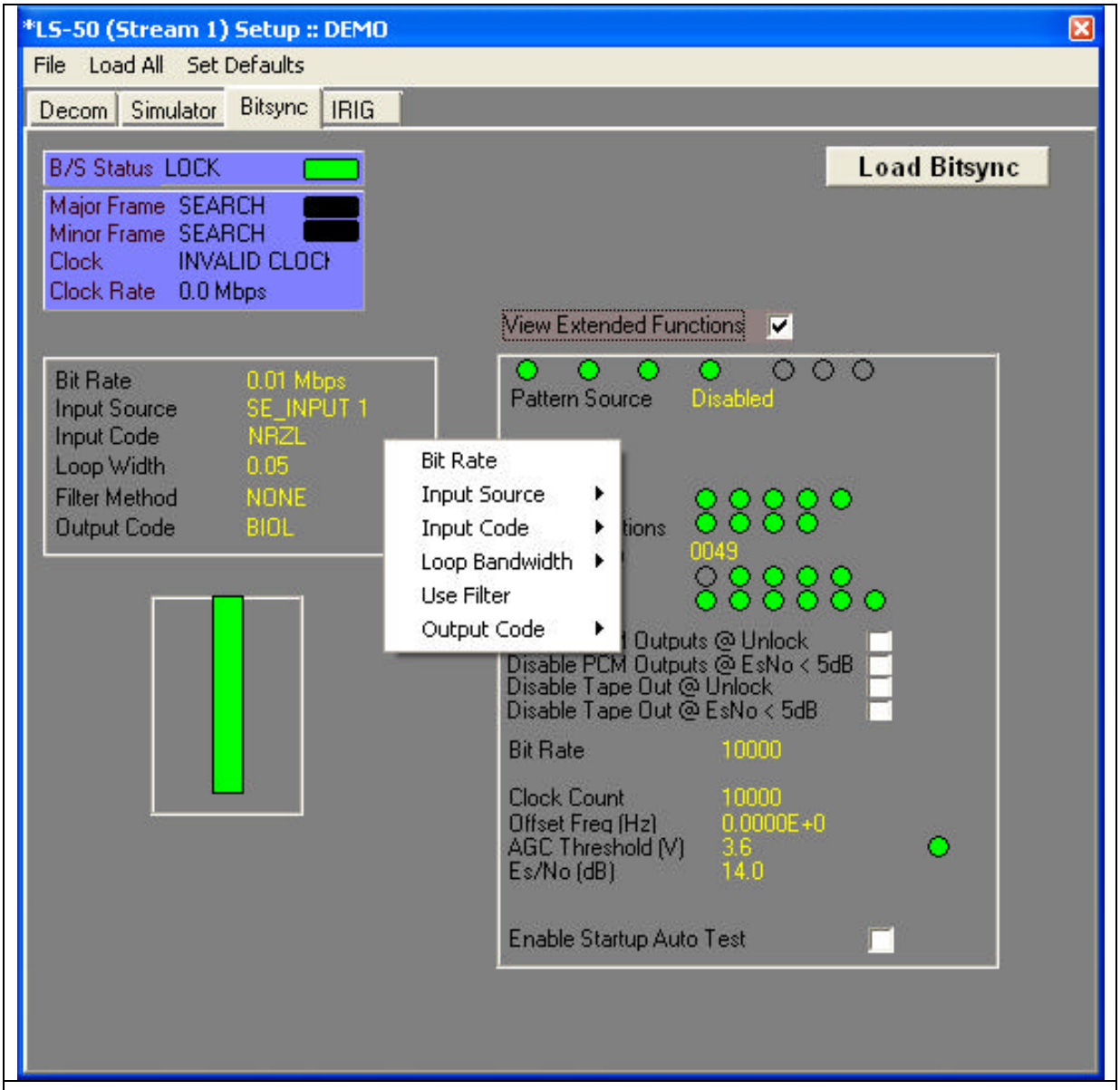
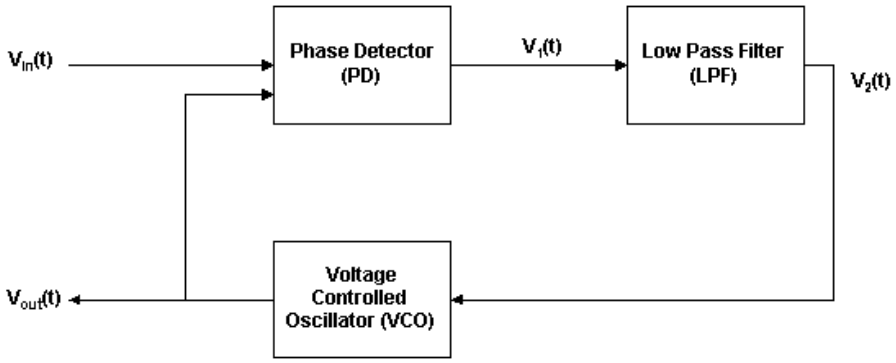



**Figure 3-11 The LS-50 Bit Synchronizer Configuration Menus**

***A "Geek" Technical Tidbit:***

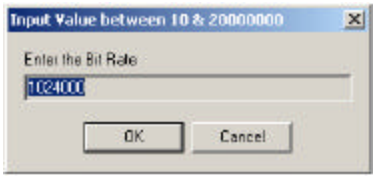
At the heart of any modern bit synchronizer is a phase-lock-loop (PLL) circuit. The implementation may be analog, digital, or some combination.



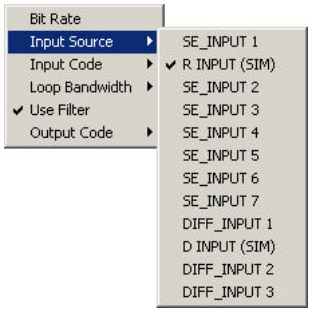

A phase lock loop consists of three basic components: a Phase Detector (PD), a Low Pass Filter (LPD) and a Voltage Controlled Oscillator (VCO). The phase detector produces an output signal, $V_1(t)$ that is a function of the phase difference between the input signal $V_{in}(t)$ and the VCO output signal $V_{out}(t)$. The low pass filter is used to remove the AC component of the signal coming from the phase detector output [$V_1(t)$]. The filtered output signal, $V_2(t)$ is the control signal that is used to change the frequency of the VCO output. The VCO is a special type of oscillator that produces a periodic waveform, the frequency of which may be varied about some free-running frequency, $f_0$, according to the value of the applied input voltage $V_2(t)$. The frequency of $f_0$ is the frequency of the VCO output when the applied input voltage $V_2(t)$ is zero.

When used in a bit synchronizer, the PLL configuration may be designed so that it acts as a narrowband tracking filter when the LPF is a narrowband filter. The frequency of the VCO will become that of one of the line components of the input spectrum. As such, the VCO output signal will equal the average frequency of this input signal component. Once the VCO has acquired this frequency component, the frequency of the VCO will track this signal component if it changes slightly. If the bandwidth of the LPF is wider, then the VCO can track the instantaneous frequency of the input signal as it changes. In either case, when the PPL tracks the changes in the input signal, the PPL is said to be "locked" If the applied input signal to the PLL has an initial frequency of $f_0$, then the PLL will acquire lock and the VCO will track the input signal frequency over some specific range, provided that the input frequency changes slowly. The loop will remain locked only over some finite range of frequency shift. This range is called the lock range. The lock range depends on the overall dc gain of the loop, including the dc gain of the LPF used. If the input signal has an initial frequency that is not equal to $f_0$, the loop may not lock, even though the input frequency is within the lock range. The frequency range over which the input signal will cause the loop to lock is called the capture range of the loop.

### 3.1.3.1 Input Bit Rate




The LS-40-DB20 Bit Synchronizer can operate over an input range of 100 bits per second to 20 Mbps for all NRZ codes, or from 100 bits per second to 10 Mbps for the Bi-Phase and Miller codes. The LS-40-DB10 is limited to 10 Mbps for NRZ codes and 5 Mbps for the Bi-Phase and Miller codes. By invoking the **Input Bit Rate** command, the user may enter the required input data rate in bits per second.




### 3.1.3.2 Input Source

The LS-40-DB Bit Synchronizer can support up to twelve (12) separate input signals. The inputs include both single-ended (SE) and differential (D/Diff) with 50? , 75?, or 1K? (Jumper Select) input impedance. The input signal amplitude supported ranges from 0.1 V pp to 10 V pp. To select the appropriate input, invoke the **Input Source** command and select the specific input from the drop-down list.

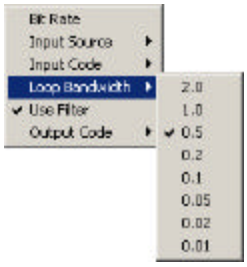
### 3.1.3.3 Input Code

The LS-40-DB Bit Synchronizer supports the PCM input code types specified in Table 3-1 below. Both normal and inverted variants are available. To select the appropriate input code, invoke the **Input Code** command and select the specific input code from the drop-down list.

**Table 3-1 LS-40-DB Supported PCM Input Codes (normal or inverted)**

| | |
|---|---|
| NRZ codes | NRZ-L, NRZ-M, NRZ-S |
| RZ codes | RZ |
| Split phase codes | Bi-Phase-L, Bi-Phase-M, Bi-Phase-S |
| Miller codes | DM-M, DM-S, $M^2$-M, $M^2$-S |
| Randomized codes | RNRZ-L, RNRZ-M, RNRZ-S |
| Randomization sequence | $2^{11}$-1, $2^{15}$-1, $2^{17}$-1, $2^{23}$-1 (normal or inverted) |

### 3.1.3.4 Loop Bandwidth




The Loop-Bandwidth of the PLL circuit in the LS-40-DB may be programmed by the user from 0.01% to 2% depending on the bit rate of the input signal. As described in the "Technical Tidbit" above, The Acquisition Range (0.04% to 8%, depending on the Loop-Bandwidth selected) and the Tracking Range (0.1% to 20%, again depending on the Loop-Bandwidth selected) are both heavily dependent on the loop bandwidth of the PLL. To select the appropriate loop bandwidth, invoke the **Loop Bandwidth** command and select the specific value from the drop-down list

### 3.1.3.5 Use Filter

The user may enable additional data filtering, prior to the actual phase lock loop of the bit synchronizer by invoking the **Use Filter** command. The additional filter uses a "Raised-Root Cosine" topology and is used to improve the performance metric of the bit synchronizer.

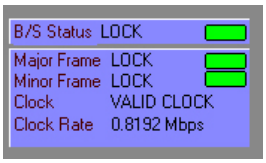
### 3.1.3.6 Output Code (for Tape Output)

The LS-40-DB Bit Synchronizer supports the PCM output code types specified in Table 3-2 below for the Tape Output. Both normal and inverted variants are available. To select the appropriate output code, invoke the **Output Code** command and select the specific output from the drop-down list.

**Table 3-2 LS-40-DB Supported PCM Output Codes for the Tape Output**

| | |
|---|---|
| NRZ codes | NRZ-L, NRZ-M, NRZ-S, INV_NRZL |
| RZ codes | RZ, INV_RZ |
| Split phase codes | Bi-Phase-L, Bi-Phase-M, Bi-Phase-S, INV_BIOL |
| Miller codes | DM-M, DM-S, $M^2$-M, $M^2$-S |
| Randomized codes | RNRZ-L, RNRZ-M, RNRZ-S |
| Randomization sequence | $2^{11}$-1, $2^{15}$-1, $2^{17}$-1, $2^{23}$-1 |

### 3.1.3.7 Bit Sync Status Display



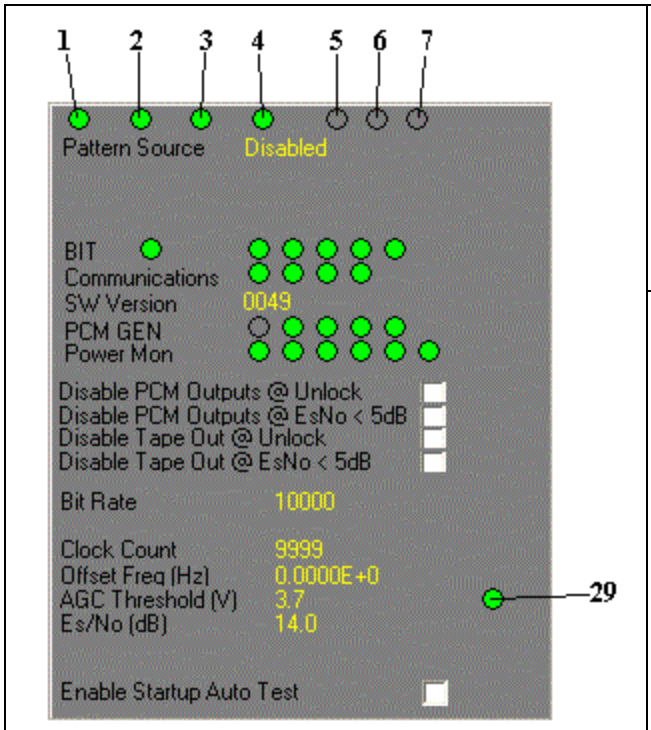

The LS-50 Bit Sync setup tab has a window display showing the status of some of the LS-50's functional states. These states include: bit synchronizer signal lock, major and minor frame lock, a valid clock indication, as well as the clock rate in Mbps. This status display is updated at a ten-hertz rate and is common to all LS-50 function setup tabs.

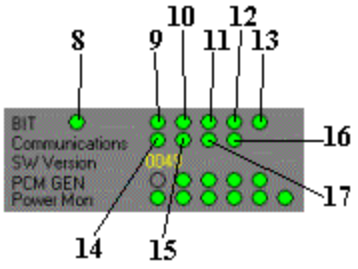
### 3.1.3.8 Load Bit Sync Button

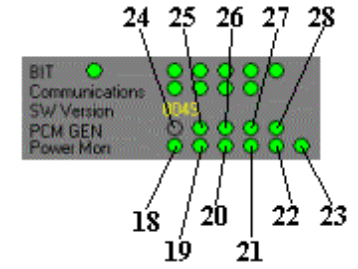

The Bit Synchronizer setup tab has a button control to load the setup information entered by the user. Changes made with any of the controls will not take affect until this button is pressed. The

user may load all four major functions (Decom, Simulator, Bitsync, and IRIG) from the "Load All" command on the menu next to the File menu). If any changes are made to the Bit Synchronizer setup without loading, a red text will appear below the Load button (shown below right), indicating the displayed data does not match the cards' loaded data.

### 3.1.3.9 View Extended Functions



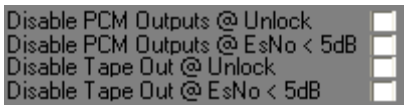




| | | | |
|---|---|---|---|
| 1 | Bit Sync Status | 16 | Protocol Error |
| 2 | Input Signal Above Threshold | 17 | Syntax Error |
| 3 | PLL Lock | 18 | +3.3 VDC |
| 4 | Input Quality Above Threshold | 19 | -5 VDC |
| 5 | PRN Correlator Locked | 20 | +5 VDC |
| 6 | PRN Correlator History | 21 | -12 VDC |
| 7 | Error Count Overflow | 22 | +12 VDC |
| 8 | Overall Health Flag | 23 | VCC (+5 VDC) |
| 9 | Synchronization Flag | 24 | Link Analysis Active |
| 10 | Power Source Error | 25 | Signal Within Range |
| 11 | Serial Error | 26 | Es/No Within Range |
| 12 | Hardware Error | 27 | Offset Frequency |
| 13 | Power Up BIT Error | 28 | Symbol Tracker |
| 14 | Overflow | 29 | Overall Health Flag |
| 15 | Semantic Error | | |

**Figure 3-12 Bit Synchronizer Extended Functions Display**

#### 3.1.3.9.1 Pattern Source

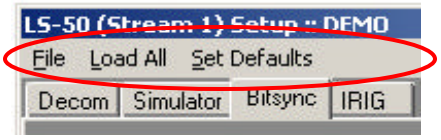

When the BERT function is enabled (see paragraph 3.1.5 on page 61), the user may select the source of the PN pattern by invoking the **Pattern Source** command. Place the cursor in the extended functions display (see Figure 3-12 upper left on page 54) and right click, then select Internal, External, or Disabled.




#### 3.1.3.9.2 Disable Output Checkboxes

The extended functions feature allows the user to automatically disable the PCM and/or Tape outputs of the bit synchronizer during certain signal conditions. The user may select to disable the PCM output whenever the bit synchronizer is out of lock, and/or when the system $E_s/N_0$ level drops below 5 dB. The tape output of the bit synchronizer may be similarly controlled.

### 3.1.3.10 Saving the Bit Synchronizer Setup Configuration




Below the window header of the "LS-50 (Stream 1) Setup" display shown in Figure 3-3 on page 18 are the *"File," "Load All,"* and *"Set Defaults"* commands. After the bit synchronizer setup configuration is complete, save the settings by invoking the *"File ? SaveAs"* command. To download all of the configurations (decom, simulator, Bitsync, and IRIG) to the LS-50 hardware, invoke the *"Load All"* command. To recall a previously defined LS-50 setup configuration, invoke the *"File ? Recall"* command and select the appropriate file from the file menu and then download the configuration to the LS-50 hardware by invoking the *"Load All"* command. To set the LS-50 hardware to its default state, invoke the *"Set Defaults"* command.

### 3.1.4 The LS-50 IRIG Time Code Tab

The LS-50 IRIG Time Code configuration setup tab and its associated menus and controls are shown in Figure 3-13 below. The IRIG time code functions include both a reader and generator that can operate with IRIG A, B, or G time code formats. The time code generator creates and outputs time information in accordance with the IRIG 200 time code standards. The time code reader is typically used to insert time information into the PCM minor frame block of data.

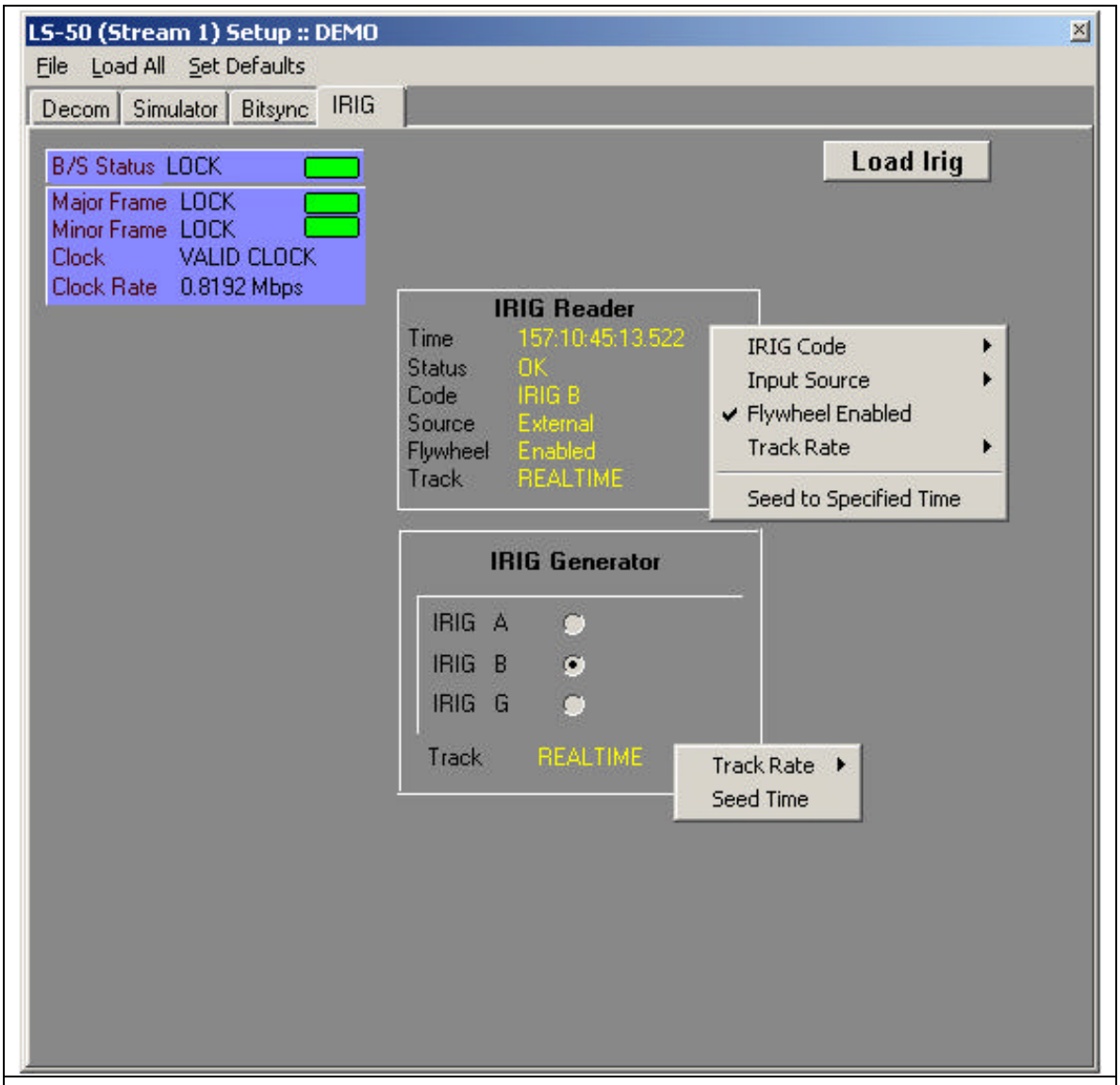



**Figure 3-13 The LS-50 IRIG Time Code Reader/Generator Configuration Menus**

***IRIG-200 "Factoid"***

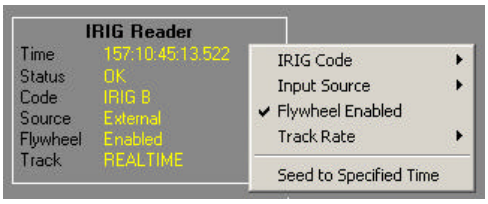
- IRIG time code formats are used on military test ranges and come in several different formats for differing resolutions. Within the IRIG formats there are two different classes: Class-I (IRIG A through H frame formats), and Class-II (MIL-STD-1553 time format). The timing information within the frame can be either days, hrs, minutes and seconds (in BCD format), or in straight binary seconds format. The basic lengths and rates of the time-code frames as defined in IRIG Standard 200 are shown below:

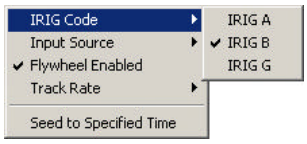
| Format | Bit Rate | Frame Rate | Bits/Frame | Carrier Freq. |
|---|---|---|---|---|
| A | 1,000 bps | 10 f/sec. | 78 bitss | 10 KHz |
| B | 100 bps | 1 f/sec. | 74 bits | 1 KHz |
| D | 1 bps | 1 f/hr. | 25 bits | 100 Hz or 1 KHz |
| E | 10 bps | 6 f/min. | 71 bits | 100 Hz or 1 KHz |
| G | 10,000 bps | 100 f/sec. | 74 bits | 100 KHz |
| H | 1 bps | 1 f/min. | 32 bits | 100 Hz or 1 KHz |

*Note: the LS-50 supports IRIG A,B and G formats.*

### 3.1.4.1 IRIG Time Code Reader Menu

The IRIG time code reader configuration consists of five controls/parameters that include: IRIG Code, Input Source, Flywheel Mode, Tracking Rate, and Seed to Specific Time. Each is discussed in the following paragraphs.

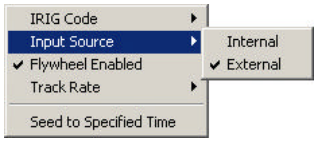



#### 3.1.4.1.1 IRIG Code

The IRIG functionality in the LS-50 supports three Class-I IRIG frame formats including "A," "B," and "G." To select the appropriate code format, the user invokes the **IRIG Code** command and selects from the drop-down list.

#### 3.1.4.1.2 Input Source




The time source for the IRIG time code reader may be either internal, or external. The user selects the input source by invoking the **Input Source** command. The "Internal" mode derives time information from the LDPS application (see paragraph 3.1.4.1.5 below). The "External" mode connects the reader input to an external time source signal (see connector pin-15, Figure 2-3 on page 13).

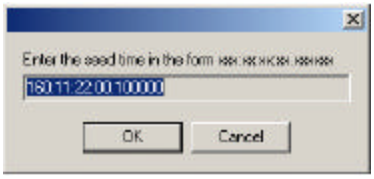
#### 3.1.4.1.3 "Flywheel" Mode

To enable the time code reader to continue to operate, or "flywheel" during dropout periods of the carrier signal, the user must select the **Flywheel Enabled** mode. While in this mode, the IRIG time reader will flywheel if the time carrier was lost for at least one cycle in the last time frame. If the carrier is lost altogether, the reader will continue to flywheel indefinitely, with an accompanying loss of timing accuracy.

#### 3.1.4.1.4 Track Rate


The IRIG time code reader can operate at several different input carrier frequencies. These include the standard carrier frequency (see the table in the IRIG-200 "Factoid"), and frequencies that are half the standard frequency (half speed) and twice the standard frequency (double speed). The **Track Rate** feature is useful when the source of the incoming time code is coming from a tape recorder playing at either half speed, or double speed. Playing the tape at a different speed will change the carrier frequency of the time code signals recorded on the tape.

#### 3.1.4.1.5 Seed to Specific Time Value




The time code reader can function in the absence of an input carrier. If no carrier is present, the system time from the CPU is used instead. In this scenario, the user may specify an arbitrary initial time, or "Seed" value by invoking the **Set Seed to Specific Time** command. Here the user enters the time in days, hrs, minutes and seconds format as shown right (days:hrs:min:sec).

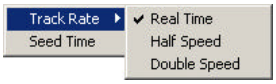
### 3.1.4.2 IRIG Time Code Generator Menus


The IRIG time code generator configuration consists of two controls/parameters that include: Tracking Rate, and Seed Time. Each is discussed in the following paragraphs.

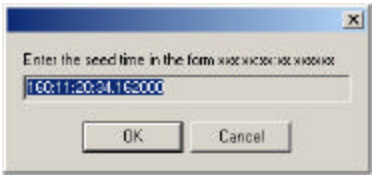
#### 3.1.4.2.1 IRIG Code

The IRIG functionality in the LS-50 supports three class-I IRIG frame formats including "A," "B," and "G." To select the appropriate code format, the user clicks one of the **IRIG Code** radio buttons to make the selection.

#### 3.1.4.2.2 Track Rate



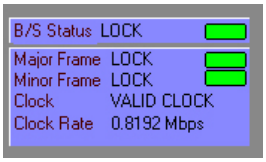

The IRIG time code generator can operate at several different output carrier frequencies. These include the standard carrier frequency (see the table in the IRIG-200 "Factoid"), and frequencies that are half the standard frequency (half speed) and twice the standard frequency (double speed). The **Track Rate** feature is useful when simulating the outgoing time code coming from a tape recorder playing at either half speed, or double speed.




#### 3.1.4.2.3 Set Seed to Specific Time Value

The initial time, or "Seed" information within the IRIG frame can be set by the user by invoking the **Seed Time** command. Here the user enters the time in days, hrs, minutes and seconds format as shown left (days:hrs:min:sec).

### 3.1.4.3 Bit Sync Status Display

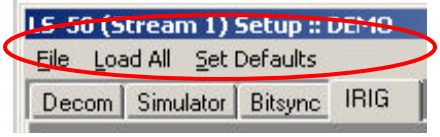



The LS-50 Bit IRIG setup tab has a window display showing the status of some of the LS-50's functional states. These states include: bit synchronizer signal lock, major and minor frame lock, a valid clock indication, as well as the clock rate in Mbps. This status display is updated at a ten-hertz rate and is common to all LS-50 function setup tabs.

### 3.1.4.4 Load IRIG Button


The IRIG setup tab has a button control to load the setup information entered by the user. Changes made with any of the controls will not take affect until this button is pressed. The user may load all four major functions (Decom, Simulator, Bitsync, and IRIG) from the "Load All" command on the menu next to the File menu). If any changes are made to the IRIG setup without loading, a red text will appear below the Load button (shown below right), indicating the displayed data does not match the cards' loaded data.

### 3.1.4.5 Saving the IRIG Time Code Setup Configuration




Below the window header of the "LS-50 (Stream 1) Setup" display shown in Figure 3-3 on page 18 are the *"File," "Load All,"* and *"Set Defaults"* commands. After the IRIG time code setup configuration is complete, save the settings by invoking the *"File ? SaveAs"* command. To download all of the configurations (decom, simulator, Bitsync, and IRIG) to the LS-50 hardware, invoke the *"Load All"* command. To recall a previously defined LS-50 setup configuration, invoke the *"File ? Recall"* command and select the appropriate file from the file menu and then download the

configuration to the LS-50 hardware by invoking the *"Load All"* command. To set the LS-50 hardware to its default state, invoke the *"Set Defaults"* command.

### 3.1.5 LS-50 Bit Error Rate Test (BERT) Function

From the "Ls50P2 Ver X.YY (Decom)" display shown below in Figure 3-14, click *"Bert"* to invoke the BERT functionality for the LS-50. Note this feature can only be accessed if the actual P2 hardware is installed in the system. It cannot be simulated, and thus the BERT menu item will not appear if the P2 board is not installed. Also note that in a dual LS-50 configuration of the P2 board (LS-55-DD), the BERT function is only supported in the first LS-50.

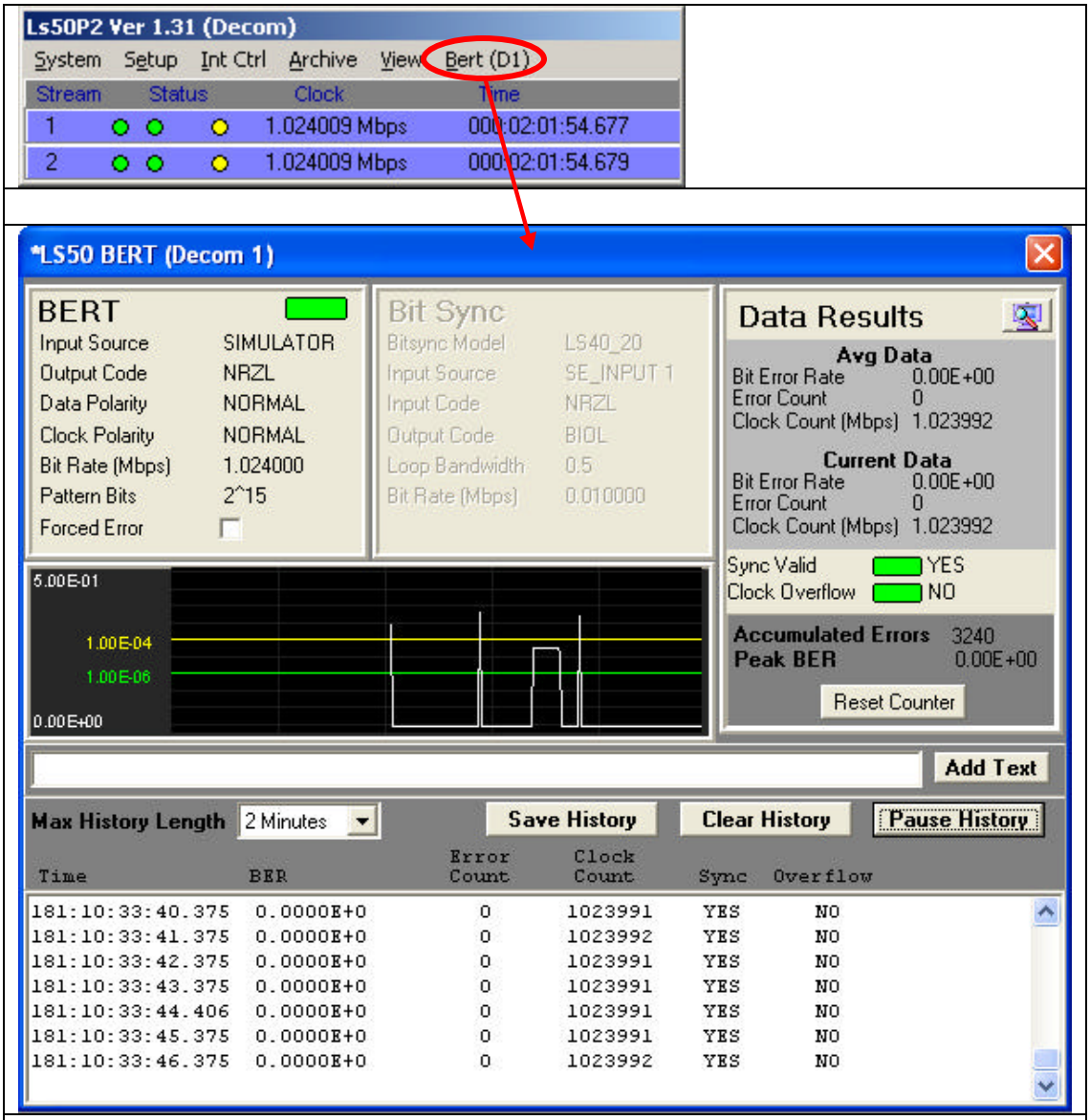


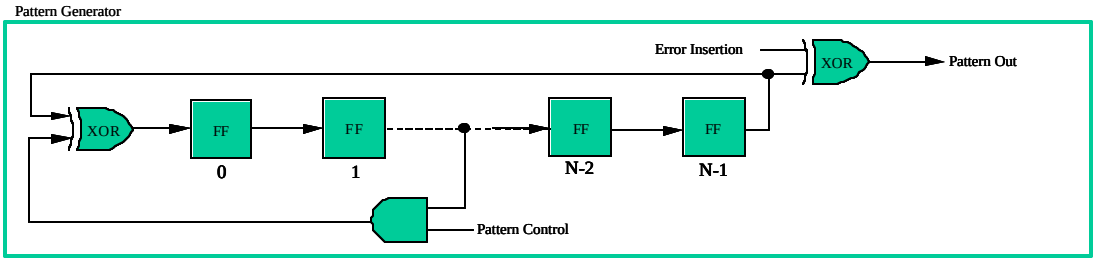

**Figure 3-14 Configuration Menus/Controls for the LS-50-P BERT Functionality**

The BERT configuration display shown in the figure above has several distinct regions that include: the BERT configuration, an optional bit sync configuration (if the selected BERT input source is Mezzanine), a Data Results Display, and a History Display. Each of these regions will be discussed starting in paragraph 3.1.5.1 on page 63. Before that however, some background information on the BERT functionality is presented.

The BERT is an instrument that generates a special digital test signal. This signal is sent through the system and the BERT counts the number of bit errors in the recovered signal and provides the user with a Bit Error Rate, or BER. The BER measurement is one of the fundamental parameters that characterize the overall performance of the telemetry system and of many of its components.

***A "Geek" Technical Tidbit:***

The basic performance measure of any digital transmission system, of which a telemetry system is an example, is the probability that any transmitted bit will be received in error. These bit errors when they occur can be introduced in many places along the path the signal flows through. Errors introduced into the transmission are often random in nature and are strongly affected by system parameters such as *signal level*, *noise level*, and *timing jitter*.




The actual digital test signal generated by the BERT employs a Pseudorandom Noise (PN) sequence to simulate traffic and to examine the transmission system for pattern-dependent tendencies or critical timing effects. An example of such a PN generator is shown above. Selecting the proper PN sequence that will be appropriate for the particular system being tested is important. Some of the key properties of the selected PN sequence that are of importance include: 1) The length of the PN Sequence. 2) The Linear Feedback Shift Register configuration used to implement the PN generator (this defines the binary run properties of the sequence). 3) Spectral line spacing of the sequence (which depends on the bit rate of the sequence). Although there are many, two PN sequence patterns have been standardized by the CCITT[6] for testing digital transmission systems. They are based on 15-stage and 23-stage Linear Feedback Shift Register configurations.

[6] CCITT Rec. 0151, Yellow Book, Vol. 4 Fascicle IV.4 Recommendation 0.151.

As mentioned earlier, errors introduced into the transmission of a digital signal are often random in nature and are strongly affected by system parameters such as signal level, noise level and noise bandwidth, timing jitter, and data rate. The BER is actually a probability and is related to another system parameter - $E_b/N_0$ (pronounced ebbno). $E_b/N_0$ is the ratio of the energy-per-bit and the noise-power-per-unit-bandwidth of the digital transmission. The $E_b/N_0$ as a quantity is a theoretical convenience rather than the direct output of a test measurement device. The parameters that do in effect define the $E_b/N_0$, and that can be directly measured by the user are the received carrier power (C), and the received noise power (N). These measured parameters, in addition to the noise bandwidth (W) of the system component being tested and the data rate ($R_b$) of the signal define the system $E_b/N_0$ in the following relationship:

$$\frac{Eb}{No} = \left(\frac{C}{N}\right)\left(\frac{W}{Rb}\right)$$

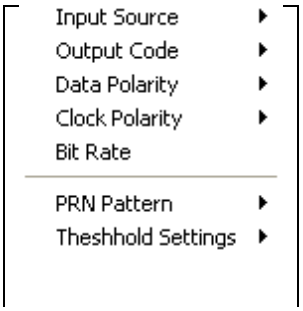
With the system $E_b/N_0$ defined in terms of measurable quantities, we can now define the BER probability. For example, the BER probability of a digital signal employing bipolar signaling expressed in terms of $E_b/N_0$ has the following relationship:

$$Pe = Q\left(\sqrt{\frac{2Eb}{No}}\right)$$

Where $E_b$ is the average energy of a modulated bit, and $N_0$ is the noise power spectral density (noise in 1-Hz bandwidth). The value Q(X) is called the Gaussian Integral Function and is usually calculated numerically. Note, the quantity "X" will vary mathematically for each type of modulation and signal encoding used in the system.

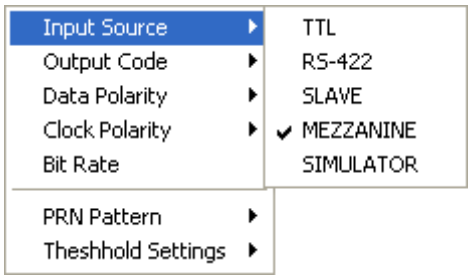
### 3.1.5.1 BERT Configuration Setup Menu

The BERT configuration pane consists of seven (7) controls/parameters that include: Input Source, Output Code, Data Polarity, Clock Polarity, Bit Rate, PRN Pattern, and Threshold Settings.

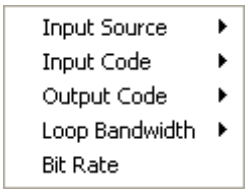



#### 3.1.5.1.1 Input Source

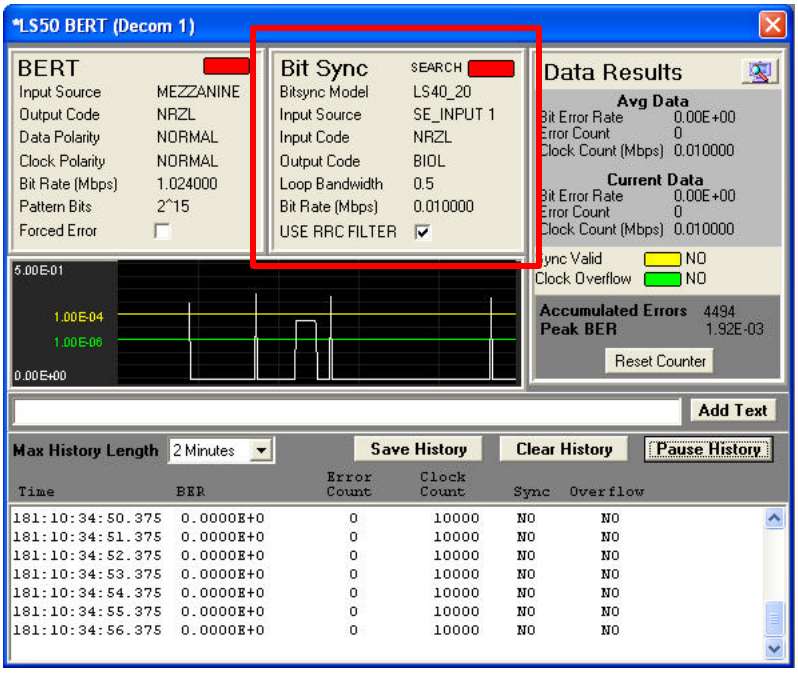
The user may select from one of five input sources by invoking the **Input Source** command and selecting the appropriate input type. The input source may include: TTL, or RS-422 differential inputs, the input from the Slave Port on the decommutator, the Mezzanine bit sync daughtercard (LS-40-DB), or the LS-50's onboard PCM simulator.




If the selected input source is the Mezzanine (LS-40-DB), then the "Bit Sync" configuration pane will appear next to the BERT configuration pain as shown below (red rectangle). Setup of the LS-40-DB is identical to that described in paragraph 3.1.3 on page 50, with the caveat that the configuration established here only applies when the BERT mode is invoked. In other words, the bit sync configuration in BERT mode can be different from the configuration during normal operation.




**Bit Sync Config**




### 3.1.5.1.2 BERT Output Code

The BERT supports the PCM output code types specified in Table 3-2 on page 53. Both normal and inverted variants are available. To select the appropriate output code, invoke the **Output Code** command and select the specific output from the drop-down list.

### 3.1.5.1.3 BERT Data Polarity

In the telemetry field, certain data transmission & demodulation schemes have inherent ambiguities that may result in the data at the decommutator input being inverted. This may be simulated by the BERT by invoking the **Data Polarity** command and selecting either "NORMAL" or "INV" (inverted).

#### 3.1.5.1.4 BERT Clock Polarity

The BERT essentially has two basic output signals: Clock, and Data. By using the **Clock Polarity** mode, the user may select either polarity sense of the output clock. In essence, the clock polarity mode allows the user to select either the rising or falling edge of the clock to coincide with the output data. For the rising edge, select "NORMAL." For the falling edge, select "INV."

#### 3.1.5.1.5 BERT Bit Rate

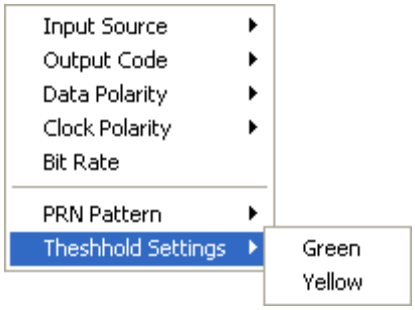
Invoking **Bit Rate** allows the user to specify the output bit rate (bits/second) of the BERT. The user may enter a value between 10 bps to 20 Mbps for NRZ codes, and 10 bps to 10 Mbps for all other codes.

#### 3.1.5.1.6 BERT PRN Pattern

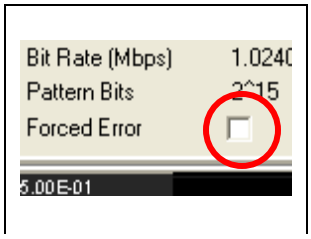
Selecting the proper PN sequence that will be appropriate for the particular system being tested is important. Some of the key properties of the selected PN sequence that are of importance include: the length of the PN Sequence, the type of Linear Feedback Shift Register configuration used to implement the PN generator (this defines the binary run properties of the sequence), and the spectral line spacing of the sequence (which depends on the bit rate of the sequence). The user may select from one of seven (7) PN sequences by invoking the **PRN Pattern** command. Available pattern lengths include: $2^{11}$-1, $2^{15}$-1, $2^{17}$-1, $2^{19}$-1, $2^{21}$-1, $2^{23}$-1, and $2^{25}$-1.

#### 3.1.5.1.7 BERT Threshold Settings

The strip cart recorder pane shown in Figure 3-15 on page 66 has two error threshold lines that may be manipulated by the user. Invoke the **Threshold Settings** command, and select either Green or Yellow. Enter the threshold value in scientific notation (X.xxE±Y) in the resulting dialog box.




#### 3.1.5.1.8 Forced Error Checkbox




To introduce bit errors at a know rate, the user may click the **Forced Error** checkbox. This will inject a single bit error that will repeat once every $2^{n}$-1 bits[7], where "**n**" is the length of the PN pattern selected by the user (see paragraph 3.1.5.1.6). Use this feature to calibrate a test scenario that is in an unknown and un-quantified state.

[7] Selecting a $2^{11}$-1 pattern, for example, will result in a BER of 4.9 x $10^{-4}$. The $2^{15}$-1 pattern will produce a BER of 3.1 x $10^{-5}$.

### 3.1.5.2 BER Strip Chart Configuration

The BERT Strip Chart configuration pane consists of four (4) controls/parameters that include: Max Stripchart Value, Min Stripchart Value, Stripchart Linearity, and Stripchart Y Min Location. To invoke the configuration menu (shown right), place the cursor in the display shown below, and right click.

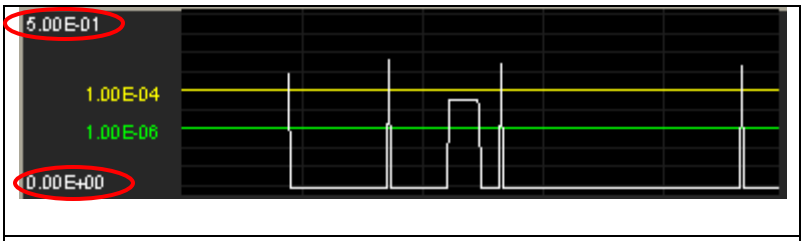
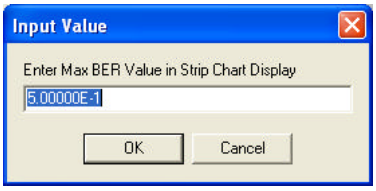




**Figure 3-15 BER Strip Chart Recorder Display**

#### 3.1.5.2.1 Min and Max Strip Chart Values

To specify the extreme values for the strip chart (red ovals in Figure 3-15) the user must invoke both the **Max Stripchart Value** and **Min Stripchart Value** command and enter the value in scientific notation (X.xxE±Y) in the resulting dialog boxs.

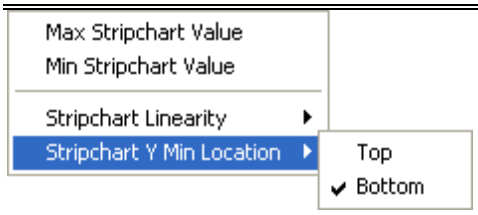




#### 3.1.5.2.2 Strip Chart Linearity

The BER strip chart can display data in either linear, or logarithmic (Logbase10) format. Invoke the **Stripchart Linearity** command and select either "Linear" or "Log10."


#### 3.1.5.2.3 Strip Chart Y Min Location

The vertical location of the minimum value specified in paragraph 3.1.5.2.1 may be placed either at the top or bottom of the strip chart display by invoking the **Stripchart Y Min Location** command.




### 3.1.5.3 Data Results Display

The BER data results are displayed as shown in Figure 3-16 below. Both long-term and instantaneous values for bit error rate, error count, and clock are displayed. Status indicators for "Sync Valid" and "Clock Overflow" are also provided. Total errors counted and the peak BER value encountered are displayed. Both of these values continue to update until the **Reset Counter** button is clicked, at which time both values will return to zero.

#### 3.1.5.3.1 BER Average Period

The average values for bit error rate, error count, and clock are calculated during a time interval defined by the user by invoking the **BER Average Period** command and selecting an interval from 1 to 60 seconds in length. To invoke the command, place the cursor at the bottom of the display shown in Figure 3-16 below and right click (resulting menu shown right).

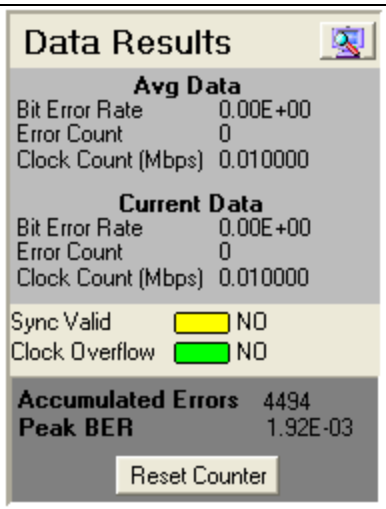




**Figure 3-16 BER Data Results Display**

### 3.1.5.4 History Display

At the bottom of the BERT configuration and status display is the BER history recording as shown in Figure 3-17 below. The history is listed chronologically and has a user defined length from 2 minutes to 24 hours. The history may be annotated by entering text in the text box and clicking the **Add Text** button. To save the history, click the **Save History** button and enter a file name and location in the resulting dialog box. At any time, the history may be suspended by clicking the **Pause History** button. To clear the history and begin again, click the **Clear History** button.

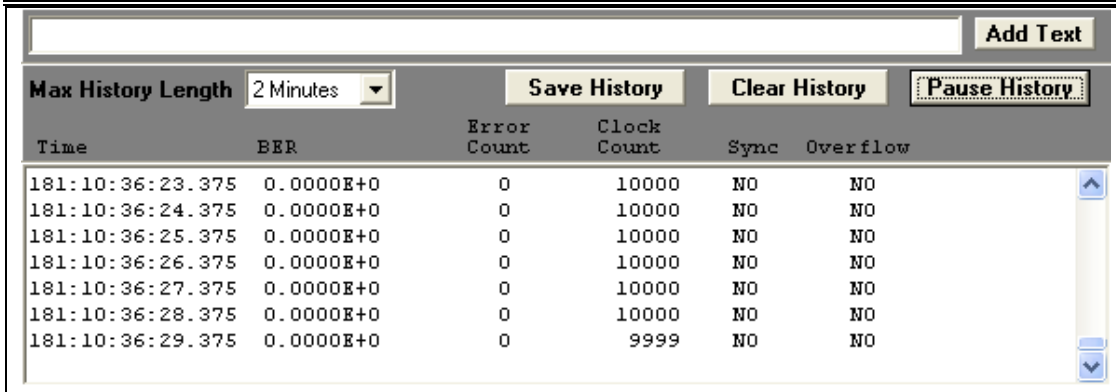
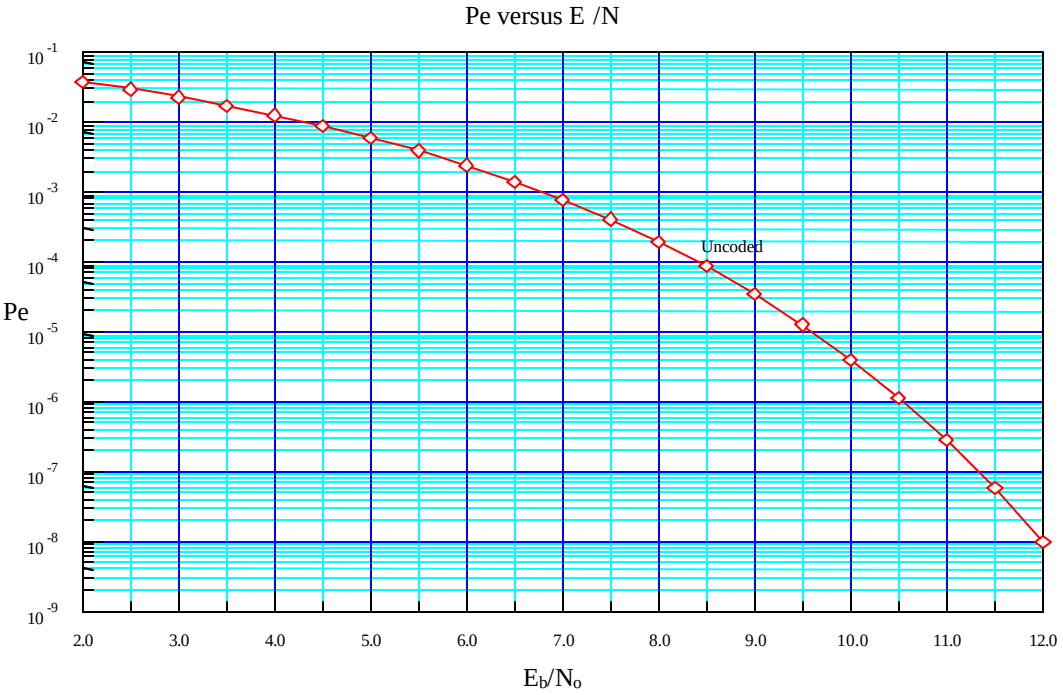



**Figure 3-17 BER History Display**

### *A "Geek" Technical Tidbit:*

It is often helpful to visualize the BER probability function graphically by using a double log plot of $P_e$ versus $E_b/N_0$. This type of plot is often referred to as a "waterfall curve." Such a plot is shown in the figure below.




It is important to understand that this plot represents the theoretical relationship between the BER probability and $E_b/N_0$. If one were to characterize the actual measured BER performance for various values of $E_b/N_0$ for the system, a slightly different set of data points would be obtained. For the actual system, for any given value of $P_e$, the resulting value of $E_b/N_0$ will always be slightly higher in value than the theoretical. The overall performance of the system is thus compared to the best-case theoretical performance and is expressed in terms of the difference, or deviation from theory. As $E_b/N_0$ is a dimensionless quantity and is expressed in terms of dB, the performance of the system is often expressed as, *"so many dB from theory"*.

### 3.1.6 The LS-50 Standalone Application

The Lumistar P2 hardware platform board, configured as a LS-50 Multi-function PCM Decommutator card is supplied with a standalone, Microsoft Windows setup and control application that duplicates many of the functions in LDPS. The standalone application (Ls50P2.exe) may not be invoked if LDPS is already running, or via versa. The standalone applications' window, shown below is almost identical to the LDPS configuration and setup window for the LS-50 shown in Figure 3-3 on page 18.

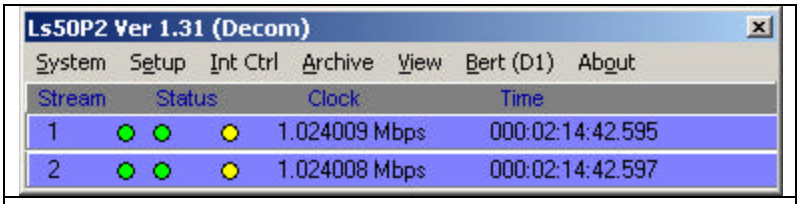
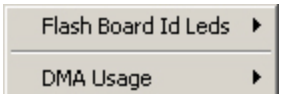



**Figure 3-18 LS-50-P Standalone Application Window**

The standalone LS-50 application has seven (7) commands that include: System, Setup, Int Control, Archive, View, Bert, and About.

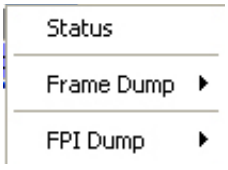



The system command has a menu of two commands. The **Flash Board Leds** command flashes the board LEDs so that multiple LS-50 cards installed in the same chassis may be identified from each other. The **DMA Usage** command is mainly a troubleshooting tool for PCs with DMA problems. By using it, one can elect not to use DMA transfer of buffered data.

The **Setup** command is identical to that described in paragraph 3.1 on page 18.


The interrupt control command has a menu of two commands. The **Start All Decoms** command starts the processing of interrupts, while the **Stop All Decoms** command stops the interrupt processing.


The Archive menu item has three commands. The **Open Raw Archive** command opens an archive file on the host computer. The **Close Raw Archive** closes the archive file currently open on the host computer. The **Archive Raw Data** command acts like a toggle that starts and pauses the archive recording.




The view command has a menu of three commands. The **Status** command displays status data for all the LS-50 decommutators installed in the system. The **Frame Dump** command displays an entire frame of data at a 20-hertz rate. The **FPI Dump** command displays an entire buffer of data at a 20-hertz rate.

Invoking the status command produces the display window shown in Figure 3-19 below. The individual status elements are described as follows:

- Overflow Count Krnl – Count of times the kernel driver missed an interrupt. If this counter is incrementing, then the system likely has interrupt conflicts.
- Overflow Count Dll – A count of times the DLL missed an interrupt from the kernel. If this is incrementing, the CPU may be stressed to hard. Try increasing the frames per interrupt setting.
- Major Frame – Lock state of the major frame.
- Minor Frame – Lock state of the minor frame
- Time – This is the time value the LS-50 is using to insert into the minor frame headers.
- Frame Count – The count of minor frames received since setup.
- Missed – The count of minor frames missed since setup. Handy to see the quality of data received.

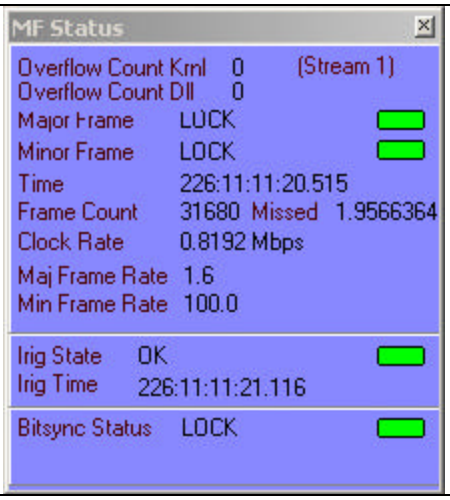



**Figure 3-19 Major Frame Status Display**

- Clock Rate – The data rate the LS-50 is configured for.
- Maj Frame Rate – The calculated rate the major frame should update, based on the decommutator setup.
- Min Frame Rate – The calculated rate the minor frame should update, based on the decommutator setup. This is the rate used to determine what the frames per interrupt setting should be for optimizing performance. For example, if the minor frame rate is 100 hertz and the frames per interrupt is set to 10, then the CPU will only interrupt at 10 hertz.
- Irig State – The state of the IRIG portion of the LS-50 (Flywheel, Error)
- Irig Time – The time decoded by the IRIG portion of the LS-50. This is only sampled at a 20 hertz rate. (The time on the decommutator portion of the card is used for time).
- Bitsync Status – The state of the bit sync portion of the LS-50
- Confidence Lvl – The confidence level metrics for the bit sync portion of the LS-50 (if equipped).

The **Frame Dump** command (see Figure 3-20 on page 72) displays an entire frame of data at a 20-hertz rate. This is only available in standalone operation. The same display is available while running the LDPS server via the **View Serial Data** menu on the server. The top part of the window in Figure 3-20 gives the decommutator setup info (abbreviated version) and some status info from the card. The **Drdy Counter** is the number of interrupts received since the decommutator was setup. The **Frames Missed**

counter is the number of minor frames missed (due to dropped lock) since the decommutator was setup. Below the status window and to the right is the major frame data. To the left of the major frame data is a selection window where the user can view selected words from the major frame, while scrolling the major frame window around. The displayed radix may be changed via the menu functions at the top of the window. The user can make a **Hardcopy** of the screen. This will create a ".BMP" (or JPG if the option is selected for JPG) in the hardcopy directory. The **Snap File** menu option will write the entire major frame of data (a snap shot of it) in ASCII format to the hardcopy directory.

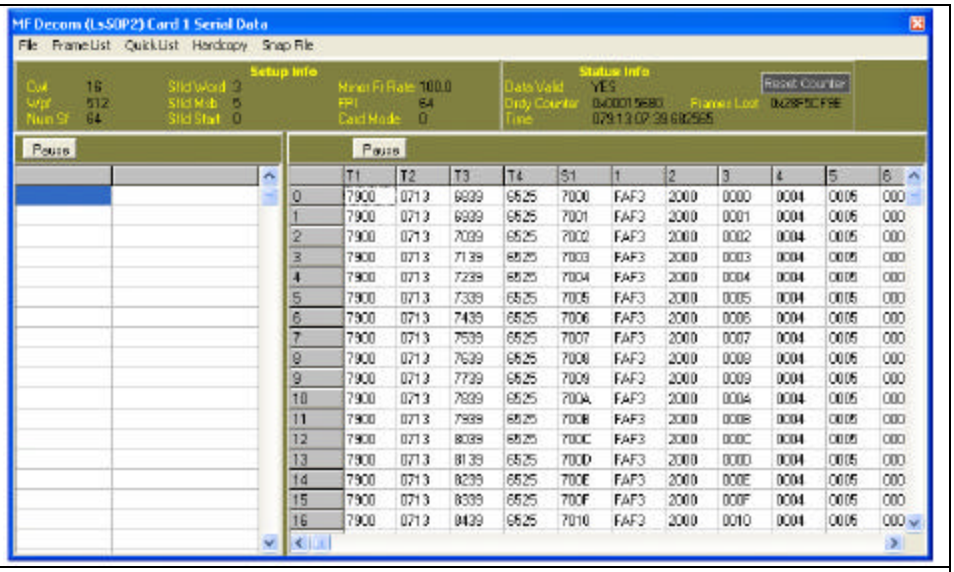



**Figure 3-20 Frame Dump Display Window**

The **FPI Dump** command produces a display like the one shown in Figure 3-21 on page 73. This display shows the content of the buffer after an interrupt. This display is handy for troubleshooting purposes.

To the right of the View command in Figure 3-18 is the **Bert** command. This is used to place the LS-50 into BER[8] mode as described in paragraph 3.1.5 on page 61. This is a modal window so the user won't be able to do anything else except interact with this window until it is closed. When the Bert window is closed, the decommutator will revert back to normal mode. Note: this feature can only be accessed if the actual LS-50 hardware is installed in the system. It cannot be simulated, and thus the BERT menu item will not appear if no board is installed.

[8] Note: Even if there are multiple LS-50 cards installed in a system, only a single card can be in BER mode at any one time. Also note that only the first LS-50 card may be in BER mode, any others are not allowed.

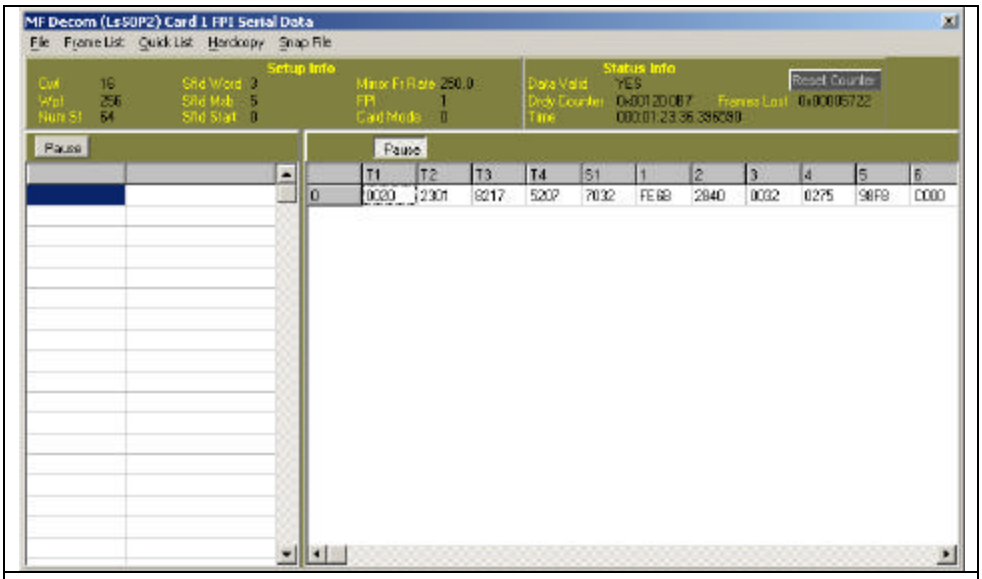



**Figure 3-21 FPI Frame Dump Display Window**

# 4 Programming Information

## 4.1 General

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

The P2 platform hardware is controlled by an array of eight-bit registers, each identified by a register number.

This chapter concludes with narratives intended to convey general guidance in converting a telemetry format definition into a download pattern for the board. Most of this setup can be done in any convenient manner. In those few cases where things are order-dependent, they will be noted.

## 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-Endean 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.

The following discussion applies to systems using *MS-DOS* or Microsoft *Windows* 3/95/98 where PCI configuration space is accessed by BIOS calls. Other environments will have system-specific ways to get this information. Consult the operating system documentation to find out how. To locate a P2 platform board in the system, perform the following steps:

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 gets that large.

2. To locate PCI9056 chips, set machine registers:

```
AX = 0xB102
CX = 0x9056
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 0x0500B00B (LS50) or 0x0700B00B (LS70), the handle points to a Lumistar P2 platform board and other configuration registers may be accessed to obtain base addresses. Otherwise skip to step 7. Set registers as shown below. Register numbers are:

Register 0x10 – PLX9056 Runtime Registers Memory Address.
Register 0x14 – PLX9056 Runtime Registers I/O Address.
Register 0x18 – Buffer Memory Address.
Register 0x1C – I/O Register Address.
Register 0x3C – (ISA-equivalent) IRQ Number.

**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. 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, then the address is in memory space. If the bit was a one, then the address is in I/O space. Reload AX, BX, and SI and repeat the call to obtain the necessary addresses. The PLX9056 runtime registers may be accessed via memory or I/O operations. Skip out when they all 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 P2 platform board may be configured to place the buffer memory in protected memory space ("flat mode") or in real space ("page mode.") In flat mode, the buffer memory occupies 128 Kbytes of contiguous address space and the Bankswitch register is ignored. 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 P2 platform board occupies 128 bytes of I/O space. Ch 0 uses the first 64 bytes. If Ch1 is configured, then it will return an identifier string from the second identifier register, 0x40 bytes away, and its memory will appear 128 Kbytes (flat) or 16 Kbytes (page) above the board base memory address.

### 4.3 Register Summaries

The P2 platform board's registers appear at the I/O address obtained by adding the hexadecimal 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. Also note there are several sets of indirect addresses associated with register accesses. Bits defined with a "–" dash are meaningless. Register assignments for Ch 1 start 0x40 bytes higher in I/O space.

All register numbers (#) are hexadecimal.

The tables that follow are memory aids for the programmer. Many bit names have been shortened for typographical purposes and have different (longer) mnemonics elsewhere in this narrative.

### 4.4 General Registers

This narrative is meant as general guidance in converting a telemetry format definition into a download pattern for the P2 platform board. Most of this setup can be done in any convenient manner. In those few cases where things are order-dependent, they will be noted.

The Board ID and Identifier registers are basic to the P2 platform board and not to any particular section.

| Table 4-1 General Write Register Summary | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Register** | **#** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Board ID | 20 | – | – | – | – | – | LED3 | LED2 | LED1 |

| Table 4-2 General Read Register Summary | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Register** | **#** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Ch 0 Identifier | 00 | 0 | "LS50" or "LS70SIM" | | | | | | |
| Ch 1 Identifier | 40 | 0 | "LS50" or "LS70SIM" or "NOTHING" | | | | | | |

### 4.4.1 Board ID Register

This register setting has no effect on the operation of the board. It controls only the state of front plate indicators (1..3). On desktop PC implementations, if there are multiple instances of the same PCI device, there is no way to tell which is which. Use this register as needed.

### 4.4.2 Identifier Register

When read repeatedly, this register returns a null-bounded ASCII string. For Lumistar decommutators it returns the string "**LS50**" to identify the board. If Ch 1 is configured, then it will return its own identifier at what would be register 0x40. Otherwise reads from register 0x40 will return "**NOTHING**"

## 4.5 LS-50 Decommutator Registers

**Table 4-3 LS-50 Decom Write Register Summary**

| Register | # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Source Control | 00 | CkPol | SOURCE | | | Force | Rev | CRC | CCIT |
| FSP Write (RS=0) | 01 | – | – | – | – | – | – | Mask | !FSP |
| URC Write (RS=1) | 01 | – | – | – | – | – | – | Mask | URC |
| FSP Threshold (RS=0) | 02 | – | Threshold Value | | | | | | |
| URC Threshold (RS=1) | 02 | – | – | Threshold Value | | | | | |
| Polarity Control | 03 | Polarity Xtol | | FAC | Trail | FSP Tolerance | | | |
| Fmt Mem Lo (RUN=0) | 04 | LSBF | MASK | Sfwd | Lcwd | WL (Word Length-1) | | | |
| Mezzanine PCM Decoder (Run=1, RS=0) | | PCM Output Code | | | | PCM Input Code | | | |
| General PCM Decoder (Run=1, RS=1) | 04 | – | – | – | – | PCM Input Code | | | |
| Fmt Mem Hi (RUN=0) | 05 | – | – | – | – | Spare (tbd) | | CRC | PASS |
| BERT Pattern (RUN=1) | 05 | – | – | – | – | REV | PATTERN | | |
| Fmt Mem Addr Lo | 06 | Address [7..0] (LSB is RS bit) | | | | | | | |
| Fmt Mem Addr Hi | 07 | Address [15..8] | | | | | | | |
| Control (CFG0 = 0) | 08 | RUN | Wobl | Wind | RA | Burst | Gmod | VFL | 2T15 |
| Control (CFG0 = 1) | 08 | RUN | WINDOW | | RA | Burst | Gmod | VFL | 2T15 |
| SfSync Position (RS=0) | 09 | SFW | | | | SFB | | | |
| SfSync Control (RS=1) | 09 | Maj Fr Mode | | Slsbf | SFUP | LastFr [9..8] | | 1stFr [9..8] | |
| First Frame (RS=0) | 0A | First Minor Frame Number [7..0] | | | | | | | |
| Last Frame (RS=1) | 0A | Last Minor Frame Number [7..0] | | | | | | | |
| Buffer Block Count | 0B | Minor Frames/Block (MAJOR=0) | | | | | | | |
| Bankswitch (pagemode) | 0C | – | – | – | – | – | PAGE | | |
| Buffer Control | 0D | IENB | AD13 | Major | Frnch | NOEL | – | CLRS | CLRD |

| Table 4-4 LS-50 Decom Read Register Summary | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Register** | **#** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Error Count Lo | 01 | Error Counter [7..0] | | | | | | | |
| Error Count Mid | 02 | Error Counter [15..8] | | | | | | | |
| Error Count Hi | 03 | OOS | Woos | Ecovf | – | Error Counter [19..16] | | | |
| Fmt Mem Lo (RUN=0) | 04 | LSBF | MASK | Sfwd | Lcwd | WL (Word Length-1) | | | |
| Clk Count Stat (RUN=1) | 04 | Update | Ovflo | – | – | – | – | – | – |
| Fmt Mem Hi (RUN=0) | 05 | – | – | – | – | Spare (tbd) | | CRC | PASS |
| Clk Count Lo (RUN=1) | 05 | Clock Counter [7..0] | | | | | | | |
| Clk Count Mid (RUN=1) | 06 | Clock Counter [15..8] | | | | | | | |
| Clk Count Hi (RUN=1) | 07 | Clock Counter [23..16] | | | | | | | |
| Status | 08 | Intrpt | POL | Xstat | Dead | Mlok | MSrc | Lock | Srch |
| Header | 09 | SLIP | Lock | Mlok | Extpin | Crcerr | CFG2 | CFG1 | CFG0 |
| Buffer Size Lo | 0A | Buffer Size [7..0] | | | | | | | |
| Buffer Size Hi | 0B | Buffer Size [15..8] | | | | | | | |
| Bankswitch | 0C | 1 | 1 | 1 | CFG4 | CFG3 | PAGE | | |
| Buffer Control | 0D | IENB | AD13 | Major | Frnch | NOEL | DMA | SIRQ | DIRQ |

### 4.5.1 The Control Register

The Control register has mode bits that affect various parts of the LS-50.

**Table 4-5 Control Register**

| Bit | Mnemonic | Description |
|---|---|---|
| 0 | 2T15 | Selects the pattern length for the BER synchronizer. See paragraph 4.11 on page 111. |
| 1 | VFL | Allows a new frame to start whenever a minor frame sync pattern is detected. Setting this bit is recommended only if frames vary in length and the longest expected frame is longer than the shortest expected time between sync patterns. If the time between patterns is longer than the longest frame, you should use BURST instead. It's okay to set BURST and VFL at once, though. |
| 2 | GMODE | Normally the decommutator output stops when it loses minor frame lock. If this bit is set, the decommutator will continue to block incoming bits into "frames" and output them. If it detects a sync pattern while in this state, it will abort the "frame" it is on and start a new one. To be meaningful, the FRNCH bit in the Buffer Control register must also be set. |
| 3 | BURST | Set this bit if the incoming data consists of fixed-length frames separated by zero or more fill bits. The data in the frames will be output and the fill bits discarded. Do not set GMODE or FRNCH along with BURST.<br>**Note:** The CRC checker is reset at the start of each minor frame if BURST is set. |
| 4 | RA | For words less than 16 bits, the decommutator parallel output and buffer memory data is left aligned with trailing zero fill to expedite number system conversions. Set this bit to yield right-aligned data with leading zero fill (certain daughtercards that use the decommutator parallel output may not function properly if RA is set.) |
| 5 | WINDOW (CFG0 = 0) | If set, the decommutator will set the SLIP status and slide over to align with an incoming frame that is one bit too short or one bit too long for the format definition. |
| 6 | WOBBLE (CFG0 = 0) | If the format definition has a major frame structure using SFID mode that is more than two minor frames long, set, WOBBLE to speed up major frame synchronization. |
| 6..5 | WINDOW (CFG0 = 1) | Allows the decommutator to set the SLIP status and slide over to align with an incoming frame that is too short or too long for the format definition. Values:<br>00 ("1-Bit") Frames must be the right length.<br>01 ("3-Bit") Frame length may be one bit off.<br>10 ("5-Bit") Frame length may be zero to two bits off.<br>11 ("7-Bit") Frame length may be zero to three bits off. |
| 7 | RUN | Set to run data and access the clock counter. Cleared to access the format memory. |

### 4.5.2 Selecting the Input Source

The LS-50 has five sets of data and clock inputs. The SRC field in the Source Control register determines the selection. In most system environments this is more a configuration than a format parameter. For one-channel boards the "Alternate" clock/data

input is shared with the simulator external clock and Slave data input pins. For two-channel boards it is the slave clock/data output from the other channel

**Table 4-6 Source Control Register**

| Bit | Mnemonic | Description |
|---|---|---|
| 0 | CCITT | Set for formats including a CCITT CRC checkword. |
| 1 | CRCEN | Set for formats including a CRC-16 or CCITT CRC checkword. |
| 2 | REVCRC | Set for reversed CRC's. |
| 3 | FORCE | Set for pseudotelemetric applications where the data stream does not include frame sync patterns, rather the first bit of the frame is defined by a pulse on the FORCE input line. Meaningful for sources 000 and 001. |
| 6..4 | SRC | Clock/Data input source selected from following:<br>000 – Primary TTL Clock/Data Input<br>001 – RS-422 Clock/Data Input<br>010 – Mezzanine Clock/Data Input (from LS-40 or LS-38.)<br>011 – Tertiary (from embedded format master) clock/data input<br>100 – On-board simulator clock/data input<br>101 – Reserved<br>110 – Alternate clock/data input<br>111 – On-board simulator clock/data input |
| 7 | CLKPOL | Set for 180-degree input clock. |

### 4.5.3 PCM Code Control

The LS-50 incorporates into its input path two PCM decoders and one PCM encoder. These code-changers are all controlled by four-bit values (see Table 4-7) that are not the same values used to control the simulator output code. If a Bi-Phase, Miller, or RZ code is selected, then the input clock is treated as a twice-rate clock.

The "Mezzanine" decoder is connected in series with the Mezzanine input. This decoder drives the LS-50 input when the mezzanine source is selected, and also the "Mezzanine" encoder. If the board is configured to host an LS-38, then the mezzanine encoder output preempts the simulator baseband output for Ch 1. Further, this output is fixed to yield square-sided data with amplitude of approximately 2V p-p unloaded.

The General PCM decoder is driven by the decomutator source select and affects any selected input source.

| Table 4-7 Decommutator PCM Codes | | | |
|---|---|---|---|
| **Value** | **PCM Code** | **Value** | **PCM Code** |
| 0000 | NRZ-L | 1000 | $M^2$ |
| 0001 | NRZ-M | 1001 | $M^2$-S |
| 0010 | NRZ-S | 1010 | Inverted NRZ-L |
| 0011 | Bi-Phase-L | 1011 | Inverted Bi-Phase-L |
| 0100 | Bi-Phase-M | 1100 | RZ |
| 0101 | Bi-Phase-S | 1101 | Inverted RZ |
| 0110 | DM-M | 1110 | RNRZ11 |
| 0111 | DM-S | 1111 | RNRZ15 |

### 4.5.4 The Frame Sync Pattern

PCM formats generally consist of strings of bits divided into words. A known group of these words is called a minor frame, whose boundaries are located by a frame sync pattern at one end or the other. Sync patterns are themselves strings of bits, usually carefully chosen to be easily recognizable by hardware. These patterns are often documented as numbers. Different patterns are used depending on the sync budget and perspectives of the entities that designed the format, but certain strings are used more often than any others. Also, in most PCM formats, all or most of the words are the same length, and the sync pattern is usually chosen to be a multiple of that length. Hence, one will probably see a number from one of 0xEB90 or 0xFE6B2840 (8- or 16-bit words,) 0xEDE20 (10-bit words,) or 0xFAF320 (8-, 12-, or 16-bit words) but the LS-50 can be programmed to use any pattern so long as it can be contained in 64 consecutive binary digits. Sometimes, too, the pattern may include "don't care" digits that are not part of the pattern, or may be offset from the frame boundary. The ARINC 573 Flight Data Recorder format, for example, starts its sync pattern two bits after the actual frame boundary, and uses those first two bits as a SFID count. Because these numbers are chosen for robust detection, the user may allow a "tolerance," meaning that any one or more bits can be wrong and still have the pattern be recognized.

Suggestion: Substitution of "digits" for "bits" in places is deliberate. Each digit ends up with three possible values. Treat the pattern as a string.

The LS-50 always presumes minor frames start with "Word 1." Word 1 may be defined as coinciding with the beginning of sync pattern (leading sync,) or as starting immediately after the end of sync pattern (trailing sync.)

The pattern actually written to the LS-50 must be extended to exactly 64 digits in length. To extend the pattern for leading sync, enough "don't care" digits must be appended after the last sync bit to make exactly 64 digits. For trailing sync, "don't care" digits must be prefixed before the first sync digit to make 64 digits. The LS-50 Low Address register

must be set to 0x00 to access the Frame Sync Pattern and Tolerance registers. Starting with the first bit, write all 64 digits to the Frame Sync Pattern Register in sequence, translating by:

```
Zero: 0x03
One: 0x02
Don't Care: 0x00
```

While sending the pattern out, count the number of digits that are not "don't care." Subtract the tolerance value (the result must be greater than zero or the format definition is nonsense) and write the result to the Frame Sync Threshold register.

Some modes of transmission have inherent ambiguities that may result in the data at the decommutator input being inverted. Hence the LS-50 can be programmed to accept patterns of either data polarity. If an inverted polarity pattern is detected, it automatically inverts the data. This is called "Automatic Polarity" and should be selected as the default unless the frame format has Frame Alternating Complement (FAC). This value is among the fields in the Polarity Control register.

**Table 4-8 Polarity Control Register**

| Bit | Mnemonic | Description |
|---|---|---|
| 0..3 | TOLERANCE | Maximum number of errors allowed in a valid frame sync pattern. |
| 4 | TRAIL | Set for trailing sync. Also set when the FORCE input is used. |
| 5 | FAC | Set for FAC or Frame Code Complement (FCC) formats. Causes true and inverted frame sync patterns to be treated equally. |
| 7..6 | POLARITY | Data polarity control selected from the following:<br>00 – Inverted.<br>10 – Automatic.<br>11 – True. |

### 4.5.5 The Decommutator Format Memory

The LS-50 uses a memory-intensive approach with a number of format parameters. The format memory holds an attribute word for each word in the minor frame, and holds the word length and a number of flags associated with that word. To access the format memory, the Control register RUN bit must be cleared. Then to access the attribute word for format word number $k$, cleave $(k-1)$ into bytes and write them to the low and high halves of the Format Memory Address register.

The LSB of the address register is also used as an indirect address bit where register numbers are overloaded (noted as encountered herein.) This function is independent of RUN. The rest of the address register is relevant only if RUN is clear.

It is advised for the user to perform two discrete single-byte accesses whether reading or writing for immunity to Big/Little-Endian issues on non-PC architectures.

Once the address has been written, one can access that location through the read/write Format Memory registers

When setting up a format with *n* words per minor frame, load the first n locations of the memory. The attributes for word 1 are written to location zero, the attributes for word 2 go to location 1, … for word n (with the LCWD bit set) to location *n–1*. Finally, another copy of the attributes for word 1 must be written to location *n*. Each attribute word is formatted as shown in the table below.

**Table 4-9 Decommutator Attribute Word**

| Bit | Mnemonic | Description |
|---|---|---|
| 0..3 | WL | The word length in bits, less 1. |
| 4 | LCWD | Set to identify last word in the minor frame. |
| 5 | SFWD | For SFID and URC formats, set to identify the word during which major frame correlation is to take place. |
| 6 | MASK | Setting this bit causes the word to be suppressed, i. e., not to appear at the output. |
| 7 | LSBF | Set for LSB-first word assembly. Clear for MSB-first. |
| 8 | PASS | In the decommutator processing the outer format of a data stream with an embedded asynchronous format, set this bit to identify the words belonging to the embedded format. |
| 9 | CRC | Set to identify word where a CRC checkword begins. |
| 11..15 | | Not used. |

### 4.5.6 Major Frame Synchronization

Many telemetry formats define structures consisting of groups of consecutively numbered minor frames. Such a structure is called a major frame. The content of the minor frames differs from one to the next so one needs to know which is which. The LS-50 has a ten-bit frame counter to identify consecutively numbered frames that appear in the frame header at the output. Such formats include ways to synchronize this counter to the larger structure.

The straightforward technique is simply not to have a major frame structure. If there is no major frame structure, the SFWD bit is not set for any location. The major frame lock status has no meaning and should be ignored.

The most common major frame synchronization technique is called SubFrame IDentification (SFID.) In this method, a word (or part of a word) is reserved in a fixed location in the minor frame. That field has a count that increments (or decrements) from

one frame to the next, starting at a known value and ending at some other known value and immediately restarting again.

More rarely encountered major frame synchronization technique is Frame Code Complement (FCC.) In this method there is no defined count field in the data. The first frame in each major frame has its frame sync pattern *inverted* with respect to the others. This technique has the advantage that no overhead bits are needed for major frame synchronization, with the corresponding disadvantages that the decommutator can correlate to the major frame structure only once per major frame, and a data polarity ambiguity is introduced by the inverted sync pattern.

Most rarely used of the major frame synchronization technique is Unique Recycling Code (URC.) This method uses a field within the minor frame similar to a SFID, but instead of an incrementing count, the field has a known value that is intended to appear only once per major frame. This technique manages to combine some of the disadvantages of both of the other techniques.

Setting the LS-50 to synchronize to a major frame includes loading several registers and (usually) setting the SFWD attribute bit (see Table 4-9 on page 84) in the proper format memory location.

Caveat: The major frame synchronizer may not work properly if a SFID or URC field ends on the minor frame boundary.

#### 4.5.6.1 SFID Correlation

If the frame format contains a SFID count, then the SFWD bit must be set in the format memory location that corresponds to the word where the count field ends (Usually the same word where it begins; the LS-50 allows the count to cross a word boundary, but in practice this almost never happens.)

Write the eight LSBs of the SFID count start value to the First Frame register (Write 0x00 to the address register first)

Write the eight LSBs of the SFID count ending value to the Last Frame register (Write 0x01 to the address register first)

The user must calculate two values for the SFID Position Register. The SFW is the length of the SFID count. This is one less than the number of bits needed to contain the largest value the SFID count. For example, if the count spans the range [0..63] the SFW value will be 5. The SFB value locates the count field in the SFWD word. This value is calculated by one of the methods described below.

| | |
|---|---|
|  | Read the following carefully. Experience shows this to be an area most prone to error in setup development. |

If the SFID word is transmitted MSB-first, then SFB is 15 less the number of bits separating the LSB of the SFID count and the LSB of the SFWD word, i. e., 15 in the usual case where the count is right-aligned.

If the SFID word is transmitted LSB-first, then SFB is 15 less the number of bits separating the MSB of the SFID count and the MSB of the SFWD word.

Shift SFW four bits to the left, add SFB, and write the result to the SFID position register (Write 0x00 to the address register first)

Calculate and write the Major Frame Sync Control register value as shown in Table 4-10 on page 87 (Write 0x01 to the address register first)

#### 4.5.6.2 FCC Correlation

For FCC correlation, the SFWD bit is not set anywhere in the format memory. The starting and ending frame count values are set as for SFID mode. Set the Major Frame Sync Control register value as shown in Table 4-10 on page 87 (Write 0x01 to the address register first) Also set the FAC bit in the Polarity Control register (Table 4-8 on page 83).

#### 4.5.6.3 URC Correlation

A URC format will have a URC pattern value associated with it. Like a frame sync pattern, a URC pattern consists of a string of one, zero, and "don't care" digits, and is loaded much the same way as a *trailing* frame sync pattern is loaded. Enough "don't care" digits are prefixed onto the front to make at least 32 digits. The LS-50 Low Address register must be set to 0x01 to access the URC Sync Pattern and Tolerance registers. Starting with the first bit, write all 32 digits to the URC Sync Pattern Register in sequence, translating by:

```
Zero: 0x02
One: 0x03
Don't Care: 0x00
```

While sending the pattern out, count the number of digits that are not "don't care." Subtract the tolerance value (the result must be greater than zero or the format definition is nonsense) and write the result to the URC Threshold register.

Set the SFWD format memory bit for the location where the URC pattern ends.

Set the first and last frame values as for SFID mode. Calculate and set an SFB value (SFW is meaningless) using the same sort of calculation as for SFID, except the bit reference is to the last (youngest) bit of the URC pattern, whether the word is LSB- or MSB-first, and set the Major Frame Sync Control register value as shown in Table 4-10 below (Write 0x01 to the address register first)

**Table 4-10 Major Frame Sync Control Register**

| Bit | Mnemonic | Description |
|---|---|---|
| 1..0 | | Bits [9..8] of the first frame value. |
| 3..2 | | Bits [9..8] of the last frame value. |
| 4 | SFUP | Set if the frame-count increments from one minor frame to the next. Clear if it decrements. |
| 5 | SLSBF | Set if a SFID count is present and is transmitted LSB-first. |
| 7..6 | SFMODE | Major frame synchronizer mode: 00 = SFID, 01 = FCC, 10 = URC. |

### 4.5.7 The Decommutator Output

The decommutator output is a stream of words formed from the input data, with a header prefixed to the beginning of each minor frame. This data is grouped into "blocks" of one or more minor frames and written to an on-board buffer memory. Two such memories are provided on the LS-50. Normally while the decommutator writes to one memory, the other is accessible for use. When a block's worth of data has been written, an interrupt is generated and the two memories are logically switched such that the fresh data becomes available. The user may directly access this memory through the system bus, or by using one of the PLX9056 DMA controllers to move the data into specific buffers in user system memory.

The header preceding each minor frame consists of four words of BCD timestamp and one word of decommutator status information, as shown in Table 4-11below.

The LS-50 has several registers associated with the buffer memory.

**Table 4-11 Frame Header**

| Wd | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 10's Days | | | | 1's Days | | | | 0 | 0 | Fly | Err | 100's Days | | | |
| 1 | 10's Minutes | | | | 1's Minutes | | | | 10's Hours | | | | 1's Hours | | | |
| 2 | 100's ms | | | | 10's ms | | | | 10's Seconds | | | | 1's Seconds | | | |
| 3 | 10's µs | | | | 1's µs | | | | 1's ms | | | | 100's µs | | | |
| 4 | Slip | Lock | MLock | Ext | Crc | 0 | Minor Frame Number | | | | | | | | | |

**Mnemonics in the Frame Header Table**

| | |
|---|---|
| FLYWHEEL | The IRIG time reader is in flywheel mode and the time carrier was lost for at least one cycle in the last time frame. |
| ERROR | Seconds in the last IRIG time frame dis agreed with the internal seconds count in the time reader. |
| SLIP | WINDOW is non-zero in the decommutator and the preceding frame was too long or too short. |
| LOCK | Decommutator minor frame Lock state. |
| MLOCK | Decommutator major frame Lock state. |
| EXTPIN | The instantaneous state of the Status input signal associated with Source 000 (regardless of what input source is selected. |
| CRCERR | The most recent CRC check failed. |

The Buffer Control and status register (Table 4-12 on page 89) sets the operating mode of the buffer memory and manages interrupts. The fields in this register are in two groups. The five MSBs are control bits. Reads from the register return the value written. The LSBs are interrupt flags. Interrupts come from three different sources. When the decommutator system interrupts, at least one of these bits is set. These bits deliberately have the counterintuitive behavior that writing ones to them clears them. As part of the user's interrupt acknowledge ritual, read this register and then immediately rewrite the value read back to it to release the interrupt.

When the memory is in page mode, only 16Kbytes of the memory are directly accessible. The three LSBs of the Bankswitch register select a page within the 128Kbyte memory. When the memory is in flat mode, this register is ignored.

The Buffer Block Count register is used to set the number of minor frames (1..256) that are gathered in a data block. Set this register to the desired number of frames, less 1. The register value is ignored if the buffer control register MAJOR bit is set, however. In any case, the user must ensure the defined data block fits into 64K words. If the buffer memory controller runs off the end of the memory it wraps around and starts to overwrite data at the beginning.

The Buffer Size register is primarily meant for variable frame length (VFL) applications. At each buffer turn, it is set to the number of words in the present buffer.

### 4.5.8 Status

The LS-50 returns status in several registers. The main status return is the Status register (Table 4-13 on page 90).

The instantaneous state of the status signals that will be written as the next frame status word in buffer memory can be read from the Header register (Table 4-14 on page 90). These bits can change asynchronously and this register is primarily for maintenance purposes.

The Bankswitch register can also be read. The three Bankswitch bits return the value written. The five MSBs are named CFG[7..3]. These bits are for version and option information. For the P2 platform they are generally "canned."

**Table 4-12 Buffer Control and Status Register**

| Bit | Mnemonic | Description |
|---|---|---|
| 0 | DINT<br>CLRDINT | When read as a 1, the decommutator has generated an end-of-block interrupt since the last interrupt acknowledge. Writing a 1 to this bit clears it. Writing a zero has no effect. |
| 1 | SINT<br>CLRSINT | When read as a 1, the simulator has generated an end-of-frame interrupt since the last interrupt acknowledge. Writing a 1 to this bit clears it. Writing a zero has no effect. |
| 2 | DMAINT | When read as a 1, the PCI9056 DMA controller has generated an end-of-transfer interrupt since the last interrupt acknowledge. This bit is read-only. You must clear the interrupt condition at the DMA controller. Note that this bit will be set only on the Ch 0 decommutator. If Ch 1 is using the other PCI9056 DMA channel, this bit will still be set in this register for the Ch 0. |
| 3 | NOEL | When set, data is blocked according to the block count, but if a major frame boundary occurs, the current buffer is terminated and a new one started. We suggest you set this bit only if the major frame length is a multiple of the block count, or the data blocks come out different lengths, some of which will have frames of stale data at the end. |
| 4 | FRNCH | Set to allow interrupts when the decommutator is not locked. You must set this bit to access the "frames" that can occur if GMODE is set in the decommutator control register. Do not set this if BURST is set unless VFL is also set. |
| 5 | MAJOR | When set, the block count is ignored and data blocks are aligned with major frame boundaries. |
| 6 | AD13 | This bit is primarily for maintenance purposes. When set, the decommutator itself and the system bus are connected to the same memory. It is possible to use this bit to access a block that was partially filled when the test vehicle smashed into something hard and the decommutator lost lock. |
| 7 | IENB | Decommutator end-of-block interrupts are allowed only if set. |

| Table 4-13 Status Register | | |
|---|---|---|
| **Bit** | **Mnemonic** | **Description** |
| 0..1 | | Minor Frame Correlator Status: 00 – Verify<br>01 – Search<br>10 – Lock |
| 2..3 | | Major Frame Correlator Status: 00 – Verify<br>01 – Search<br>10 – Lock |
| 4 | DEAD | Set if the input clock stops or drops below a rate of (**TBD**) bps. |
| 5 | XSTAT | Returns the signal level of the Status input if one is associated with the selected input source. Otherwise meaningless. |
| 6 | POL | Set if the data at the decommutator input is being inverted, either automatically or under program control. |
| 7 | INTRPT | Set if a data block has ended, whether decommutator interrupts are enabled or not. Reading the Buffer Control register clears this bit. |

| Table 4-14 Header Register | | |
|---|---|---|
| **Bit** | **Mnemonic** | **Description** |
| 0 | CFG0 | Set for hardware revisions that recognize wider sync windows. |
| 2..1 | | Meaningless. |
| 3 | CRCERR | Set if most recent CRC check failed. |
| 4 | EXTPIN | Similar to XSTAT except this bit always returns the state of the status line for Source 0, regardless of the selected input source. |
| 5 | SLOCK | Major Frame Lock state. |
| 6 | LOCK | Minor Frame Lock state. |
| 7 | SLIP | WINDOW is set in the decommutator and the preceding frame was too long or too short. |

The LS-50 monitors the incoming bit rate by counting clocks at the selected input and registering the count every second. If RUN is set, the user may determine the incoming rate by polling UPD (bit 7 of the Clock Count Status register.) When it comes on, read the three Clock Count registers and concatenate their values.

Reading the MSBs of the Clock Count clears the UPD and OVF bits.

The OVF flag (bit 6 of the Clock Count Status register) can be treated as a 25$^{th}$ bit, allowing a range up to 33 MHz.

| | |
|---|---|
|  | Editorial: If the LS-50 is in minor frame lock, then the clock count value is not very interesting, because whatever it reads is probably the right value. Nevertheless, if the LS-50 is *not* in lock, then the clock count may be cogent. Better to display both MF lock status and clock count. |

## 4.6 The IRIG Time Reader

An IRIG time reader is included with the LS-50. This reader can either be synchronized with an IRIG time carrier, or be seeded with local time and used as a free-running clock. The time reader is primarily used to provide timestamps for incoming data, but can also read the time directly back.

Operating modes for the IRIG reader are set in the IRIG reader Control register (Table 4-17 on page 92).

**Table 4-15 IRIG Reader Write Register Summary**

| Register | # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Control | 18 | – | – | Arrow | | – | Flywhl | MODE | |
| RTC Setting | 19 | "^AA*dddhhmmss*^W" | | | | | | | |
| Freeze Command | 1A | – | – | – | – | – | – | – | – |

**Table 4-16 IRIG Reader Read Register Summary**

| Register | # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| BCD Time Return at last Freeze Command (except BUSY) | 18 | 10's µs | | | | 1's µs | | | |
| | 19 | 1's ms | | | | 100's µs | | | |
| | 1A | 100's ms | | | | 10's ms | | | |
| | 1B | 0 | 10's Seconds | | | 1's Seconds | | | |
| | 1C | 0 | 1's Minutes | | | 1's Minutes | | | |
| | 1D | 0 | 0 | 10's Hours | | 1's Hours | | | |
| | 1E | 10's Days | | | | 1's Days | | | |
| | 1F | BUSY | 0 | FLY | ERR | 100's Days | | | |

### 4.6.1 Setting the Real Time Clock

The real time clock free runs at the rate controlled by the ARROW value. To set the time, put the reader in Real Time Clock mode and convert date and time of year to an ASCII string:

*"^AAdddhhmmss^W"*

where ^A is 0x01, ddd is a zero-extended day number [001..366], hh is a zero-extended hour number [00..23], etc., and ^W is 0x17. Write the characters of this string in sequence to the RTC Setting register. After each write, poll and wait for the BUSY flag (bit 7 of register 0x1F) to clear (it only takes a few hundred nanoseconds) before

continuing. While loading the time, the reader output is held, fractional seconds are cleared, and the respective time digits appear as they are loaded. When the last character is sent, the clock starts to run.

**Table 4-17 IRIG Reader Control Register**

| Bit | Mnemonic | Description |
|---|---|---|
| 1..0 | IRIGMODE | Reader mode and carrier select:<br>00 – Real time clock. Any incoming time carrier is ignored.<br>01 – IRIG B.<br>10 – IRIG A.<br>11 – IRIG G. |
| 2 | FLYWHEEL | Set to allow reader time to flywheel during time carrier dropouts. Must be cleared if the time carrier is not running at the selected rate. |
| 3 | | Unused. |
| 5..4 | ARROW | Specifies the length of the arrow of time in RTC or carrier flywheel:<br>00 – Real time<br>01 – Time at half rate.<br>10 – Time at twice rate. |
| 7..6 | | Unused. |

### 4.6.2 Reading Time

The main purpose of the time reader is to provide timestamps for data. However, one may read time directly from the reader into the system without disturbing the timestamp operation. To read time, first capture it by writing (anything) to the Freeze Command register. Then read the BCD time of year in microseconds by reading registers as shown in Table 4-16 on page 91.

## 4.7 The LS-50 PCM Simulator

The PCM simulator on the LS-50 can be used to generate a test data stream. This simulator primarily outputs a *static* data stream and is not intended for such purposes as archival playback or uplink command generation.

When setting up a PCM simulator for a given format, the same issues of sync pattern and format "shape" arise as they did with the decommutator. The decommutator doesn't always need to provide data content, but a simulator does. The simulator needs to output *something* for every position in the format.

A simulator also needs to provide something else a decommutator gets from its environment, a data rate clock. That is why, after describing the Command and Mode registers that control the simulator, the following paragraphs start at the end of things before jumping back to the beginning.

Another reason for starting at the output of the simulator is the need to write a lot of memory to it. Unlike the decommutator, the simulator has no RUN bit; memory accesses are sequenced under control of the simulator clock.

**Table 4-18 LS-50 Simulator Write Register Summary**

| Register | # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Command | 10 | MREQ | Mread | IACK | IENB | RStrt | XCLK | PAGE | – |
| Bankswitch | 11 | 0 | 0 | 0 | 0 | REGS | MBF | PAGE | MB1 |
| Low Address | 12 | Mailbox/Exchange Address [7..0] | | | | | | | |
| High Address | 13 | Mailbox/Exchange Address [15..8] | | | | | | | |
| Data Memory (Bs=00x1) | 14 | Word Attributes/Right-Aligned Data [7..0] | | | | | | | |
| | 15 | Word Attributes/Right-Aligned Data [15..8] | | | | | | | |
| Frame Attr Memory (Bs=01x0) | 14 | EOSF | UW6 | UW5 | UW4 | UW3 | UW2 | UW1 | UW0 |
| Frame Start (Bs=1000, Address=0) | 14 | First Minor Frame Number [7..0] | | | | | | | |
| | 15 | 1 | FCC | FAC | SFUP | – | – | 1stFr [9..8] | |
| Mode (Bs=1000, Address=1) | 14 | 0 | ClkDiv | | CWS | LSBF | – | CRC | CCIT |
| | 15 | WIDE | Uplink | – | BCRC | Event | BERT | ERR | 2T15 |
| NCO Setup (Bs=1000, Address=2) | 14 | NCO Control Bits | | | | | | | |
| Encoder Control (Bs=1000, Address=3) | 14 | QUIET | Slave | RNRZ Control | | PCM Code | | | |
| | 15 | – | – | – | DIFF | INV | Swap | 1/3 | RATE |
| External Register (Bs=1000, Address=4) | 14 | Reserved for future use | | | | | | | |
| BERT Pattern (Bs=1000, Address=17) | 14 | – | – | – | – | REV | PATTERN | | |
| RS-232 Data | 21 | (ASCII Character) | | | | | | | |
| RF Control | 22 | XDAT | RFEN | SW | Xmod | PMF | | | |
| RS-232 Baud Rate | 23 | 8 LSBs of –96 / (BaudRate / 1200) | | | | | | | |
| RF Command | 24 | EEPROMCMD | | LO | – | DAC | DEV | Addr[9..8] | |
| RF EEPROM Address | 25 | EEPROM Addr [7..0] | | | | | | | |
| Low RF Data | 26 | EEPROM or DAC Data [7..0] | | | | | | | |
| High RF Data | 27 | EEPROM or DAC Data [15..8] | | | | | | | |

| Table 4-19 LS-50 Simulator Read Register Summary | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Register** | **#** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Command | 10 | MREQ | Mread | Intrpt | IENB | 0 | XCLK | PAGE | Pgif |
| Not defined | 11 | – | – | – | – | – | – | – | – |
| Not defined | 12 | – | – | – | – | – | – | – | – |
| Not defined | 13 | – | – | – | – | – | – | – | – |
| Memory Mailbox (Bs = 0xxx) | 14 | Defined Same As Write | | | | | | | |
| | 15 | | | | | | | | |
| RS-232 Data | 21 | (ASCII Character) – top of a 64-byte FIFO | | | | | | | |
| RF Control | 22 | Defined Same As Write | | | | | | | |
| RS-232 Status | 23 | – | – | – | – | – | – | XBE | RBF |
| RF Status | 24 | – | EEOP | LO | – | DAC | – | – | – |
| Not Used | 25 | – | – | – | – | – | – | – | – |
| Low EEPROM Return | 26 | EEPROM Data [7..0] | | | | | | | |
| High EEPROM Return | 27 | EEPROM Data [15..8] | | | | | | | |

### 4.7.1 Simulator Command Register and Mode Registers

The Simulator Command register has a variety of bits that require quick access, so this is the one simulator register that is directly accessed. The other operational registers are indirect addressed in an effort to fit the simulator into a limited amount of I/O space. The Command register is laid out as shown in Table 4-20 on page 95. Note the register is read/write but some of the bits have subtly different but related meanings for write and read operations – purposely so to allow using read-modify-write-type accesses sensible under a variety of conditions. The Mode register (Table 4-21 on page 96) and Frame Start Register (Table 4-22 on page 96) are used to set static operating modes for the simulator. These registers are indirect addressed, so access to them is slower, but their contents are not likely to change except when a complete simulator setup is performed.

To access the Mode register, write 0x08 to the simulator Bankswitch register (setting only the REGS bit), and write 0x01 to the Low Address register.

To access the Frame Start register, write 0x08 to the simulator Bankswitch register (setting only the REGS bit), and write 0x00 to the Low Address register.

| Table 4-20 LS-50 Simulator Command Register | | |
|---|---|---|
| **Bit** | **Mnemonic** | **Description** |
| 0 | PAGE IN EFFECT | Has no meaning when written.<br>When read, returns the state of the simulator internal PAGE flag. This flag is copied from the PAGE bit (bit 1 of this register) on each minor frame boundary and determines which page of the simulator memory is to be used during the next frame. |
| 1 | PAGE | The simulator memory is divided into two equivalent pages. This bit specifies which page to use. If UPLINK is not set, PAGE can be used to synchronously switch the simulator between two formats. If the bit is unchanged, the same page is used over and over again unless...<br>If UPLINK is set, the format defined in page 0 is output repeatedly until PAGE is set. The *next* minor frame is in the format defined in page 1 and is output *once* and PAGE is cleared by the simulator.<br>When read, returns the last value written. |
| 2 | XCLK | When set the simulator clock generator is ignored and the simulator clock is to be supplied from the simulator external clock input. Set this bit if external clocking is desired. If this simulator is the slave of a simulator-pair generating an asynchronous embedded format, the simulator external clock must be connected to the slave clock output of the master simulator and this bit must be set to run data.<br>Conversely, this bit should be cleared during a simulator setup to ensure the simulator clock is being allowed to run.<br>When read, returns the last value written. |
| 3 | RESTART | Writing a one clears the simulator word and frame counters and aborts from any simulator memory access in progress. Writing a zero has no effect.<br>When read, always returns zero. |
| 4 | IENB | Setting this bit causes the simulator to generate a system interrupt each time the INTRPT bit is set.<br>When read, returns the last value written. |
| 5 | IACK INTRPT | When read, returns the state of the simulator interrupt flag. This flag is set on every minor frame boundary, whether interrupts are enabled or not.<br>Writing a one to this bit clears the interrupt flag. Writing a zero has no effect. |
| 6 | MREAD | Controls direction of transfer between the simulator memory and exchange registers for memory accesses. Set for reads, clear for writes.<br>When read, returns the last value written. |
| 7 | MREQ | Setting this bit initiates a simulator memory access. Writing a zero has no effect.<br>When read, returns a one if an access is in progress and not completed yet. |

| Table 4-21 LS-50 Simulator Mode Register | | |
|---|---|---|
| **Bit** | **Mnemonic** | **Description** |
| 0 | CCITT | When set, causes a CRC-CCITT checkword to be calculated. Otherwise CRC-16 is calculated. Has no meaning if CRCEN is clear or if no CRC location is specified in the simulator word attributes. |
| 1 | CRCEN | When set, causes a CRC checkword to be calculated. Has no meaning if no CRC location is specified in the simulator word attributes. |
| 2 | REVCRC | When set, causes a reversed CRC checkword to be calculated. Has no meaning if CRCEN is clear. |
| 3 | LSBF | When set, all simulator data is output LSB-first. |
| 4 | CWS | When cleared, simulator common output data is read from the common data area in simulator memory. This means all words not pre-empted by sync, unique, or waveform words. When set, the common data area is ignored and all common output words have the value of the simulator CWS memory location. |
| 6..5 | DIV | Selects a pre-scale ratio for the simulator clock: Choose one of:<br>00 – Divide by 1.<br>01 – Divide by 16.<br>10 – Divide by 256.<br>11 – Divide by 4096. |
| 7 | MREQ | Maintenance use only. Do not set this bit. |
| 8 | 2T15 | Specifies a 32,767-bit PRN pattern. See paragraph 4.11 on page 111. This bit works but is redundant if CFG6 is set. |
| 9 | ERR | Forces one error every PRN pattern iteration. See paragraph 4.11. |
| 10 | BERT | Pre-empts simulator output with PRN pattern. Zero for normal operation. See paragraph 4.11. |
| 11 | EVENT | 0-to-1 transition forces a single PRN pattern error. See paragraph 4.11. |
| 12 | BCRC | Normally the CRC generator is reset at the end of the checkword. Set this bit to cause the generator to be reset again at the end of the minor frame. |
| 13 | Meaningless | |
| 14 | UPLINK | When set, clears the PAGE bit if the PAGE IN EFFECT bit is set. |
| 15 | WIDE | Set this bit to cause the simulator frame strobe output to rise at the beginning of the last word in the minor frame. If not set, the frame strobe rises with the beginning of the last bit. The strobe always falls on the frame boundary. |

| Table 4-22 LS-50 Simulator Frame Start Register | | |
|---|---|---|
| **Bit** | **Mnemonic** | **Description** |
| 9..0 | 1STFRAME | Frame number of the first minor frame in the major frame. |
| 11..10 | | Meaningless. |
| 12 | SFUP | When set, causes minor frame numbers to increment in the course of the major frame. |
| 13 | FAC | When set, words with the FSP attribute are inverted during odd-numbered minor frames. |
| 14 | FCC | When set, words with the FSP attribute are inverted during the first minor frame of each major frame. |
| 15 | WDST | Always set this bit. |

### 4.7.2 Output Formatting

Aside from a straight serial (NRZ-L) data stream and clock, the LS-50 simulator has an additional output that is encoded by one of a set of standardized schemes used for telemetry transmission. At the output is a convolutional encoder (k=7), followed by a randomizer, followed by a PCM encoder. These encoders are set up by an Encoder Control register. This register is accessed by indirect addressing through the Exchange register. To write values to it, write 0x08 to the simulator Bankswitch register (setting only the REGS bit), and write 0x03 to the Low Address register. The upper and lower halves of the Encoder Control register (Table 4-23 on page 97) can then be accessed by writing the upper and lower halves of the Exchange register.

One of the parameters associated with the Encoder Control register is the output PCM Code. There are a number of selections here. Each has an implied parameter called the Code Factor associated with it. This factor and others are used in the calculations to set up the simulator clock generator. To properly set the data rate, the value written to this register must be known.

**Table 4-23 LS-50 Simulator Encoder Control Register**

| Bit | Mnemonic | Description |
|---|---|---|
| 3..0 | PCM CODE | PCM Output code. Choose one of the following:<br>0000 – NRZ-L<br>0001 – Inverted NRZ-L<br>0010 – NRZ-M<br>0011 – NRZ-S<br>0100 – RZ<br>0110 – Inverted RZ<br>1000 – Bi-Phase-L<br>1001 – Inverted Bi-Phase-L<br>1010 – Bi-Phase-M<br>1011 – Bi-Phase-S<br>1100 – DM-M<br>1101 – DM-S<br>1110 – $M^2$<br>1111 – $M^2$-S |
| 5..4 | RANDOMIZE | RNRZ Randomizer Control:<br>00 – Off<br>01 – RNRZ11<br>10 – RNRZ15 |
| 6 | SLAVEN | LS-50: When set, if Unique Word 6 is to appear at the output, it is preempted. The simulator slave output clock runs during this word and data from a slave simulator is inserted. This feature is for use in the master of a simulator-pair to create a simulated stream with an asynchronous embedded format. Meaningless for LS-70 configuration. |
| 7 | QUIET | Forces the PCM and baseband outputs to DC. |
| 8 | RATE | Enables the convolutional encoder. Causes output to be rate-1/2 encoded unless 1/3 is also set. |
| 9 | 1/3 | If RATE is set, causes output to be rate-1/3 encoded. |
| 10 | SWAP | Swaps the G1 and G2 symbol when set. |
| 11 | INVERT | Inverts the G1 symbol when set. |
| 12 | DIFF | Enables differential encoding when set. |
| 15..13 | | Meaningless. |

### 4.7.3 The Clock Generator

The simulator uses a Numerically-Controlled Oscillator (NCO) to generate its output clock. Exercise the following algorithm to get the NCO operating.

1. If the logical AND of 0x0C and the value (chosen according to the output code) written to the Encoder Control register is not zero, then multiply the desired output bit rate by 2. Otherwise multiply by 1.

2. If the RATE bit in the Encoder Control register is set, then multiply the rate by 2, but if the 1/3 bit is also set, then multiply the rate by 3.

3. Clamp the upper bound of the rate at 35,000,000. Neither the NCO nor the simulator is certified to be reliable beyond that point.

4. If the bit-rate result is 262,144 or greater, then the DIV field in the Mode register (Table 4-21 on page 96) should be 00. Otherwise choose a DIV field and multiply the rate by the "by" factor to get larger than 262,144 if possible.

5. Multiply the bit-rate by 35.791394. Truncate the result to an integer.

6. Write 0x08 to the simulator Bankswitch register (setting only the REGS bit), and write 0x02 to the Low Address register.

7. Write 4 to the low exchange register. Then write 2. Then write 0.

8. Repeat 32 times:
   - The value "x" is 0x80 if the LSB of the rate is 1, or 0x0 if it is zero. Write x to the low exchange register. Then write x+4. Then shift the rate one bit to the right, discarding the LSB.
   - (End repeat)

9. Set the rate value to 1. Then repeat 8 times:
   1. The value "x" is 0x80 if the LSB of the rate is 1, or 0x0 if it is zero. Write x to the low exchange register. Then write x+4. Then shift the rate one bit to the right, discarding the LSB.
   2. (End repeat)

10. Write 2 to the low exchange register. Then write 0.

### 4.7.4 Communicating With Simulator Memory

The LS-50 simulator uses two separate memories during its operation. Each minor frame word location has an attribute word associated with it, and a data value to be output. The attribute table and common value table are each 16K (16-bit) words long. There are also

tables of unique, sync, and waveform values. These items are together in one 64K word memory. Additionally, there are two pages of this memory, making 128K words total.

Juxtaposed with this memory, there are two pages of simulator frame attributes in a separate memory with each page being 1K (8-bit) words in length.

To access memory, the simulator clock must be running. If performing a large number of accesses, write to the Command register clearing the XCLK bit and also the Mode register, setting the DIV field to 00. Set the simulator clock generator to some convenient rate.

The authors personal preference is $2^{20}$, or 1048576 Hz.

Specify a word location by writing to the Low and High Address registers. Select which memory to access by writing to the Bankswitch register (Table 4-24 below). Set **only one** of the MB1, MBF, or REGS bits.

**Table 4-24 LS-50 Simulator Bankswitch Register**

| Bit | Mnemonic | Description |
|---|---|---|
| 0 | MB1 | Set to access data/word attribute memory. |
| 1 | MB0 | Memory PAGE associated with memory accesses. Meaningless if neither MB1 nor MBF are set. |
| 2 | MBF | Set to access frame attribute memory. |
| 3 | REGS | Set to write indirectly addressed simulator registers instead of memory. |
| 7..4 | | Reserved. Do not set any of these bits. |

If performing a memory-write, write the data to the Low and High Exchange registers and then write to the Command register, clearing MREAD and setting MREQ. Repeatedly poll the Command register, waiting for MREQ to go away, which will take one to three clock times.

If performing a memory read, write to the Command register, setting MREAD and MREQ. Repeatedly poll the Command register, waiting for MREQ to go away, which will take one to three clock times. The returned data can then be read from the Exchange registers.

### 4.7.5 The Simulator Memory Map

The data/word attribute memory is mapped as shown in Table 4-25 on page 100. Remember there are actually two such memories, selected by an additional PAGE address bit. When accessing memory, use the MB0 bit in the Bankswitch register to

select the correct page. When the simulator uses the memory operationally, it will use its PAGE IN EFFECT bit.

**Table 4-25 LS-50 Simulator Memory Map**

| Range | Definition |
|---|---|
| 0x0000—0x3FFF | Common data values, lookup by word number mod 16,384 (CWS=0.) |
| 0x4000—0x4006 | Unique word values, lookup by unique word number. |
| 0x4007 | CWS data value (CWS=1.) |
| 0x4008—0x43FF | Not used. |
| 0x4400—0x47FF | Frame Sync Pattern data, lookup by word number modulo 1,024. |
| 0x4800—0x4BFF | SFID data, lookup by minor frame number. |
| 0x4C00—0x4FFF | Waveform 1 data, lookup by minor frame number. |
| 0x5000—0x53FF | Waveform 2 data, lookup by minor frame number. |
| 0x5400—0x57FF | Waveform 3 data, lookup by minor frame number. |
| 0x5800—0x5BFF | Waveform 4 data, lookup by minor frame number. |
| 0x5C00—0x5FFF | Waveform 5 data, lookup by minor frame number. |
| 0x6000—0x7FFF | Not used. |
| 0x8000—0xBFFF | Word attributes, lookup by word number modulo 16,384. |
| 0xC000—0xFFFF | Not used. |

### 4.7.6 Attributes and Data

The simulator Frame Attribute Memory is loaded with frame attributes. The Data and Word Attribute Memory holds both output data and word attributes.

Each minor frame word location has an associated attribute word. The attribute words are stored in data memory (See Table 4-25 on page 100 for location.) The attribute word is formatted per Table 4-26 on page 101.

Each minor frame in the major frame has an attribute word in the Frame Attribute memory, as shown in Table 4-27 on page 101.

Data in the data areas is always right aligned, regardless of word length or bit ordering. However, for LSB-first data, the frame sync words need to be bit-reversed to get the pattern to output properly. Also note the simulator allocates and integral number of words to the frame sync pattern, and an entire word to the SFID count, regardless of how many bits they actually use.

The simulator design provides support for formats using FCC or SFID major frame correlation. There is no provision for URC formats *per se*. To simulate a URC, one will need to pre-empt enough unique words to put the URC pattern in the first minor frame.

| Table 4-26 LS-50 Simulator Word Attributes | | |
|---|---|---|
| **Bit** | **Mnemonic** | **Description** |
| 5..0 | WSPL*n* | If FSPL*n* (*n*=0..5) is set for the current frame, substitute contents of unique word *n* location for whatever other data would be output here. |
| 6 | WSPL6 | If FSPL6 bit is set for the current frame, substitute the contents of unique word 6 location for whatever other data would be output here. **… BUT …**<br>If SLAVEN is set in the Encoder Control register (Table 4-23 on page 97), do not substitute the unique word. Instead, allow the slave clock output to run during this word and insert whatever appears at the slave data input. |
| 7 | CRCW | Output the CRC checkword, starting with the first bit of this word. The checkword output lasts for 16 bit periods, during which any other data otherwise defined for output is discarded. |
| 11..8 | WL | The word length in bits, less 1. |
| 14..12 | I | Data source for this word, unless overridden by CRC, slave, or unique word:<br>000 – Common data.<br>001 – Frame Sync Pattern data<br>010 – SFID data<br>011 – Waveform 1 data<br>100..111 – Waveform (2..5) data |
| 15 | EOF | Set to identify last word in minor frame. |

| Table 4-27 LS-50 Simulator Frame Attributes | | |
|---|---|---|
| **Bit** | **Mnemonic** | **Description** |
| 6..0 | FSPL*n* | If WSPL*n* set for the current word, substitute contents of unique word *n* location for whatever other data would be output here. |
| 7 | EOSF | Set to identify last minor frame in the major frame. |

### 4.7.7 Baseband and RF Control

History has combined two functions that were more closely related in past versions of this hardware. The P2 hardware platform includes a serial EEPROM holding 1,024 sixteen-bit words (Ch 1 includes another such EEPROM.) The EEPROM holds configuration data for the LS-50 simulator (and RF output, if present.)

Multiple RF output options are provided by the P2 hardware platform. As of this writing, the only one implemented uses a Quasonix low-power RF transmitter module that provides both FM and SOQPSK modulated signals. Controls for the baseband output and pre-modulation filters have no effect on this RF output.

#### 4.7.7.1 EEPROM Access

A given EEPROM location is read by performing the following sequence:

1. Wait for the RF status register to become zero (usually immediate.)
2. Write the location (8 LSBs) to the RF EEPROM Address Register.
3. Shift the location right 8 bits, then add 0x40. Write to the RF Command Register. The RF Status Register EEOP bit comes on.

4. Wait (about 15ms) for the RF status register to become zero.
5. Concatenate the values of the Low and High EEPROM Return registers to get a 16-bit result.
6. The first EEPROM operation following system reset sometimes yields an improper result. Start by performing a dummy read, discarding the data.

The EEPROM data is arranged as shown in Table 4–28 below.

**Table 4-28 RF EEPROM Map**

| Address | Default | Description |
|---|---|---|
| 0 | 75 | 74 => RF output present. Other values reserved. |
| 1 | 0 | 2 = Quasonix Transmitter. Other values reserved. |
| 2 | 0 | Reserved. |
| 3 | 102 | Identifies this EEPROM map. |
| 4 | 1 | Number of RF Bands implemented. |
| 5 | 2200 | Band 0 Minimum RF Output Frequency (MHz) |
| 6 | 2394 | Band 0 Maximum RF Output Frequency (MHz) |
| 7..10 | 0 | Reserved for more band limits. |
| 11 | 10241 | Legacy Reserve |
| 12 | 250 | Legacy Reserve |
| 13..14 | 0 | Reserved |
| 15 | 8 | Number of pre-modulation filter selections available. |
| 16 | 250 | Pre-Mod Filter 0 cutoff in kHz. 65,535 if square-sided. |
| 17 | 500 | Pre-Mod Filter 1 cutoff in kHz. 0 if filter not present. |
| 18..31 | | etc. More Pre-Mod Filter selections. |
| 32 | 10 | Maximum RF output level in dBm. |
| 33 | -5 | RF Attenuator step (dB) |
| 34 | 19 | Number of RF attenuator Steps available. |
| 35 | | RF Attenuator D/A setting for listed maximum output level. |
| 36 | | RF Attenuator D/A setting for listed maximum output + location 33. |
| 37 | | RF Attenuator D/A setting for listed maximum output + 2 * location 33. |
| 38..63 | | More RF Attenuator D/A settings |
| 64,65 | 0, 0 | Dummy Baseband Amplitude Lookup first entry |
| 66,67 | | n (mv), n counts Baseband Amplitude Lookup entry |
| 68..153 | | More Deviation Lookup entries |
| 154.. | | Reserved |

#### 4.7.7.2 Baseband Output Level

The baseband output level is adjustable from less than 200 mv to approximately 8,000 mv p-p. This "volume control," level is set by a D/A converter. To choose the setting, start with the desired (unloaded) output level in millivolts and scan even-numbered EEPROM locations starting with location 64. When the EEPROM data value is about to exceed the level setting, stop. These settings are sufficiently close together so the D/A value can be calculated by linear interpolation between the count values in the next-higher odd-numbered EEPROM locations. Consider the following example:

The desired level is 1,000mv. Scanning the EEPROM might yield:

Loc 104 = 919.
Loc 105 = 6830.
Loc 106 = 1011.
Loc 107 = 7214.

D/A Value = (7214 - 6830) ´ (1000 - 919) ¸ (1011 - 919) + 6830 = **7168**.

To set the level, split the value into bytes and write the eight LSBs to the Low RF Data Register, and the MSBs to the High RF Data Register. Then write 0x0C to the RF Command Register. The DAC bit comes on in the RF status register and persists for about 10ms.

#### 4.7.7.3 Pre-Mod Filtering

The baseband output passes through a pre-modulation filter on its way out. The effect of the filter is determined by the value of EEPROM location n+16, where n is the value of the PMF field. A value of 0 means the output is disabled. A value of 65,535 means the data output is square-sided (i. e., unfiltered.) Any other value represents the filter cutoff in kHz.

#### 4.7.7.4 External Data Input

Setting the XMOD bit disconnects the LS-50 simulator from the baseband output driver. In this scenario, the baseband output is driven from the external baseband input. To use this input, SW1-3 (and SW1-4 for Ch 1) must be turned on and no shunt is permitted on E1A-11 (E1A-13 for Ch 1.)

To modulate the Quasonix transmitter with an external data source, connect the (TTL) signal to the simulator Slave Data Input and set the XDAT bit. This data is expected to be synchronous with the simulator clock.

#### 4.7.7.5 The Quasonix Transmitter

The Quasonix transmitter is controlled by an RS-232 link. An on-board UART provides the necessary communications path. Before attempting to control the transmitter, initialize the UART by writing 0xFE to the RS-232 Baud Rate Register (57,600 baud.) Then read the RS-232 data register and discard the value.

In normal operation, the Quasonix transmitter is controlled by sending a series of commands using the RS-232 link. The transmitter responds to each command with a response string ending with a '+' character. That results in the following general protocol to send a command:

1. Read the RS-232 Data Register and discard.

2. Send a character by writing it to the RS-232 Data Register. Wait (about 180µs) for the XBE bit to come back on in the RS-232 Status Register. Repeat until all characters are sent.
3. Read the RS-232 Status Register and wait for the RBF bit to come on. Read the RS-232 Data Register. Repeat if the value returned is not '+' (0x2B.)

In the following, <sp> means 0x20 (ASCII space) and <cr> means 0x0D (carriage return.)

To set the RF output frequency, send: **FR<sp>nnnn.n<cr>**
where nnnn.n is Fc in kHz.

To set the modulation type send: **MO<sp>n<cr>**
where n is 0 for SOQPSK and 1 for FM.
Sending this command will "hang" the interface for approximately a second. This is normal.

To enable RF output send: **RF<sp>1<cr>**
and set the RFEN bit in the RF Control Register (SW should always be zero at this writing.)

To disable RF output send **RF<sp>0<cr>**
and clear RFEN.

## 4.8 The IRIG Time Generator

The IRIG Time Generator is physically a part of the PCM Simulator but is a distinct logical entity. It has its own setup registers, all accessed through a single I/O address, with the adjacent address being an indirect address register. Hence, to access a register in the generator, write the register number to the address register (register 0x16 relative to the base I/O address) and then access the data through register 0x17.

**Table 4-29 IRIG Generator Write Register Summary**

| **Register** | **#** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|---|
| Indirect Address | 16 | – | – | – | – | Adr | | | |
| **Register 17:** | **Adr** | | | | | | | | |
| BCD Seconds Preset | 00 | – | 10's Seconds | | | 1's Seconds | | | |
| BCD Minutes Preset | 01 | – | 10's Minutes | | | 1's Minutes | | | |
| BCD Hours Preset | 02 | – | – | 10's Hours | | 1's Hours | | | |
| BCD Days Preset | 03 | 10's Days | | | | 1's Days | | | |
| | 04 | – | – | – | – | – | – | 100's Days | |
| Control Functions (by index number) | 05 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 |
| | 06 | 66 | 65 | 64 | 63 | 62 | 61 | 60 | 58 |
| | 07 | 75 | 74 | 73 | 72 | 71 | 70 | 68 | 67 |
| | 08 | – | – | – | – | – | 78 | 77 | 76 |
| Seconds from Midnight Preset (IRIG A, IRIG B) | 09 | Seconds [7..0] | | | | | | | |
| | 0A | Seconds [15..8] | | | | | | | |
| Control | 0B | SET | – | Arrow | | MODE | | HOLD | Sec16 |
| Data Hold Frac Secs | 0C | $10^{th's}$ Seconds | | | | $100^{th's}$ Seconds | | | |
| BCD Data Hold Secs | 0D | DHold | 10's Seconds | | | 1's Seconds | | | |

**Table 4-30 IRIG Generator Read Register Summary**

| **Register** | **#** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|---|
| Not Defined | 16 | – | – | – | – | – | – | – | – |
| **Register 17:** | **Adr** | | | | | | | | |
| BCD Frac Seconds | 00 | .1's Seconds | | | | .01's Seconds | | | |
| BCD Seconds | 01 | 0 | 10's Seconds | | | 1's Seconds | | | |
| BCD Minutes | 02 | 0 | 10's Minutes | | | 1's Minutes | | | |
| BCD Hours | 03 | 0 | 0 | 10's Hours | | 1's Hours | | | |
| BCD Days | 04 | 10's Days | | | | 1's Days | | | |
| BCD Days | 05 | Indeterminate | | | | | | 100's Days | |

### 4.8.1 Setting Time on the IRIG Generator

Setting up the IRIG generator is a three-step process. First, write the generator Control register (Table 4-31 on page 106) setting the MODE and ARROW fields to get the time carrier running at the right frequency. Then write the start time into the preset registers (Table 4-31). There isn't much call for the control functions, but if there are values, write them at this time. Finally, write the Control register again, this time with the PRESET bit

set. This loads the time counters and resets the generator back to the beginning of the (first) time frame of that second.

One may also read the time of day back from the generator, but the data returned is unfrozen and may be subject to rollover errors. This process is mostly intended for maintenance purposes. The time is in BCD as described in Table 4-30 above.

**Table 4-31 IRIG Generator Control Register**

| Bit | Mnemonic | Description |
|---|---|---|
| 0 | SEC16 | Binary time in seconds from midnight is 17 bits long. This is the MSB to go into effect on a PRESET. |
| 1 | HOLD | When set, stops time in seconds from incrementing. |
| 3..2 | MODE | Selects a time carrier: 0x – IRIG B.<br>10 – IRIG A.<br>11 – IRIG G. |
| 5..4 | ARROW | Specifies the length of the arrow of time, i. e., carrier frequency:<br>00 – Real time<br>01 – Time at half rate.<br>10 – Time at twice rate. |
| 6 | | Unused. |
| 7 | PRESET | Resets the generator to the beginning of a time frame, clears fractional seconds, and places the time loaded into the Preset registers into effect. |

## 4.9 Interrupts

If the data rates are extremely low and operational demands are not great, one may be able to avoid using interrupts by using the polling technique described below.

### 4.9.1 Polling

The decommutator and simulator both have interrupt flags that latch set on a particular event, regardless of whether interrupts are actually enabled.

To implement a polling scheme for synchronization of an application program with the LS-50 decommutator, wait for the Status register INTRPT bit to come on (Table 4-13 on page 90). This event indicates a buffer turnover. As soon as this event occurs, read from the Buffer Status register and discard the value. This will turn off the flag. Then immediately move the data from buffer memory.

To implement a polling scheme for synchronization of an application program with the LS-50, or LS-70 simulator, wait for the Command register INTRPT bit to come on (Table 4-20 for the LS-50 or **Error! Reference source not found.** for the LS-70) to indicate a minor frame boundary. As soon as this event occurs, turn off the flag by clearing bit 7 (MREQ) of the value by reading, and then writing it back. Bit 0 returns which page of simulator memory is in use at that moment.

### 4.9.2 Using Interrupts

Polling techniques will suffice only for the least-demanding applications. Usually one will have to engage interrupts and synchronize at the interrupt level. The user will need to connect the driver or application to the PCI interrupt assigned to the board. This calls for careful setup and usage.

#### 4.9.2.1 Connecting to the System

An interrupt handler is required for the driver or application. The customary and universal rules of interrupt processing apply; save the processor state, acknowledge the interrupt, expeditiously do what time-critical things are required, restore the processor state and get back out. In the iAP86 (PC) environment, the only state saved by the interrupt itself is the program counter and processor flag register. Any CPU registers used must be saved for later restoration. Of course, the stack pointer needs to be left where one found it.

In PCI systems, a further complication is caused by the fact that as a PCI device, the physical interrupt may be shared with some other device. This means the application may get interrupts that are not intended for it. One needs to interrogate the card to find out which interrupt has been assigned (see paragraph 4.2 on page 74) and connect to the appropriate handler, usually by a system call passing the address of the handler. Before doing that, though, one may (as in MS-DOS) be required to first interrogate the other system calls to determine who currently "owns" the interrupt. On exit, the handler must restore CPU registers and end by transferring control (e. g., by a far jump) to *that* entity. Other environments may have different ways to accomplish this.

#### 4.9.2.2 Preparing to be interrupted

After connecting the handler to the system, further prepare the system and the board for interrupts. For PC environments, this means making sure the "8259" interrupt is unmasked for the selected IRQ, and experience indicates it is wise to issue a non-specific End-of-Interrupt at this time. Again for PC environments, this means writing 0x20 to I/O port 0x20 and, if the IRQ number is greater than 7, also writing 0x20 to I/O port 0xA0. Theoretically one should not need to do this, but the theory is contrary to experiment, and this procedure seems to be harmless.

The history of moving designs across architectures has left several levels of interrupt enabled. The decommutator *per se* and simulator both have INTRPT status flags. The user must clear out any pending interrupts. For the decommutator, read the Buffer Control/Status register and write the value read back to it. For the simulator, read the Command register and write the value read back to it.

If the DMA controller is to be used, clear any pending DMA interrupt. This is done by reading the DMA Command/Status registers and writing 0x08 back if the value read had bit 4 set (i. e., logical AND with 0x10). These registers are at offsets 0xA8 (Channel 0) and/or 0xA9 (Channel 1) in the PLX9056 Runtime Register space. (PLX9056 Runtime Registers appear in both memory and I/O space.)

One must enable the PLX9056 PCI interrupt by setting the Interrupt Control/Status register at PLX9056 Runtime Register offset 0x69, with the logical OR of 0x09 with the value read from that register.

Finally, one must set the IENB bit(s) for the decommutator and/or simulator in the Decommutator Control and/or Simulator Command registers. If data is running (for the simulator, data is always running if its clock is running) an interrupt will eventually occur.

#### 4.9.2.3 Being Interrupted

At interrupt time the handler will be called. Again, because of the shared nature of PCI interrupts, one must interrogate the card to find out if it is the one interrupting. Decommutator and simulator interrupts set the DINT and SINT bits in the Buffer Control/Status register. Immediately write back the value read. This will clear these bits and release the PCI interrupt *if they were interrupting* (writing zero back to these bits has no effect.) If the bit in the value read is clear, skip over the operation it calls for and continue with the rest of the handler. In practice, this may mean skipping everything.

For the simulator interrupt, one would ordinarily set a semaphore for some operation down at the task level, and possibly rewrite the simulator Command register, toggling its PAGE bit.

For the decommutator, initiate whatever sort of operation designed to move data from its buffer memory, by means of pick/choose, block move, or initiating a DMA operation.

If bit 2 of the Buffer Status is set, the interrupt was from the PLX9056 itself, usually a DMA end-of-operation. One cannot clear this bit in the Buffer Status by writing to it. There is the further complication in dual-channel environments. The PLX9056 has two DMA channels, with the implication that one is reserved for each channel, but only one interrupt pin. Both interrupts are connected to bit 2 of the Buffer Status for the Ch 0 decommutator – if the Ch 1 is also a decommutator, bit 2 of *its* Buffer Status is meaningless. Therefore, poll both of the two DMA Command/ Status registers. If bit 4 was set, the DMA interrupt for that channel is active and one needs to write 0x08 back to clear it. Usually for a DMA interrupt one would post some sort of semaphore indicating data is available in system memory.

Finally, one should issue a non-specific End-of-Interrupt before leaving the handler. This is fail-safe because one is still at interrupt level. Any other handlers daisy-chained downstream will still run.

## 4.10 DMA

The PLX PCI9056 includes two DMA controllers that permit rapid data movement. The implication is to use them to move incoming data from the buffer into system memory

after the buffers toggle. By arbitrary convention, DMA Channel 1 is assigned to Ch 1, and Channel 0 is reserved for use by Ch 1 if configured. If one plans to write for the DMA controller, obtain the PLX9056 data sheet (http://www.plxtech.com).

DMA operations require knowledge of the physical addresses in system memory, which may or may not be the same as the logical addresses used by an application. One will need to make the necessary system calls to convert logical addresses to physical addresses. If the operating system does not provide this capability, then the DMA Controller cannot be used to move data.

For all DMA applications, the PCI9056 PCI Command Register (a 16-bit register in Runtime register space at offset 0x04) should be set to 0x07. Additionally, each DMA controller has three data items in registers. These registers are in the Runtime register space:

— A 32-bit DMA Mode register at offset 0x94 (0x80 for DMA Channel 0.)
— A 16-byte Descriptor at offset 0x98 (0x84 for Channel 0.)
— An 8-bit Command register at offset 0xA9 (0xA8 for Channel 0.)

DMA operations may be run as chained or unchained. Unchained DMA, using a single descriptor, can be used if one can always move the entire active part of the data buffer to one continuous physical buffer in system memory. The memory management used by some operating systems (e. g., Windows NT) does not always permit that because it breaks all user memory buffers into segments of some arbitrary size (4096 bytes for NT) or less. The user needs to set up chained DMA operations in these systems. A chained DMA operation needs multiple descriptors (collectively called a "chaining table" albeit the actual structure is that of a singly-linked list) in memory someplace.

The PCI9056 allows the chaining table to be stored either in local memory (i. e., on-board) or PCI memory (i. e., elsewhere in the system). Some similar products from other companies used an earlier version of the PLX part that did not allow the chaining table to be in PCI memory. Those products had local memory reserved for the chaining table. Conversely, the P2 platform board has no local memory where the chaining table can reliably be stored.

### 4.10.1 DMA Descriptors

A descriptor is a structure of four 32-bit items. When descriptors are stored in memory, each must start on a paragraph boundary. This means the physical address of the first byte of the descriptor must end in 0x0.

The first item of the descriptor is the PCI physical address of the target buffer in system memory. For unchained DMA, this is the start address. For chained DMA, this is the starting physical address of the segment.

The next item of the descriptor is the local physical address of the source data. The active data buffer is at local addresses 0x00000..0x1FFFF and there is no local mapping in this area so the address in the (first) descriptor is normally always zero.

Note: Local address space 0x20000..0x23FFF also points into the buffer, but the high-order local address bits are supplied by the bankswitch register. The DMA controller has no knowledge or control over the bankswitch register so this is of little utility. HOWEVER, the other mapping (starting at local address 0) is always in effect, so the DMA controller can be used to pour out the entire buffer whether the PCI interface is in flat or page mode.

The third item in the descriptor is the transfer size in bytes. For an unchained DMA operation this is the active buffer size. For chained DMA this is the size of the current segment.

The fourth item is called a descriptor pointer. This item is split into two fields. Bits (04..31) have meaning only for chained DMA. They are the 28 MSBs of the physical address of the next PCI physical address field in the chaining table (why this is referred to as a pointer.) When this value is used as an address the four LSBs are understood to be zero regardless of their real value.

Bit 03 is a transfer direction bit and is always 1 to move data from buffers to system memory.

Bit 0 has meaning only for chained DMA. It is 1 to specify the next descriptor pointer field is a PCI physical address and must be 1 for all applications using chained DMA.

Bits (01..02) have meaning only for chained DMA. They must be 11 for the last descriptor in the chaining table, and 00 for all of its predecessors.

### 4.10.2 DMA Channel Mode Register

The DMA Mode Register specifies operating conditions for a DMA operation. Only a few values are meaningful here. The recommended basic value is 0x0143. Add 0x200 to this value to specify a chained DMA. Additionally, add 0x400 to obtain a second interrupt when the DMA operation completes.

When setting up a chained DMA operation, the first descriptor can be loaded directly into the PCI9056 descriptor register. This is not recommended for two reasons. First, because it creates the complication of a special case, and also because the earlier PCI9080 had a known bug that causes improper operation if physical PCI memory mapping could result in a mixture of chained and unchained DMA operations. If one will have *any* need for chained DMA, use chaining for *all* DMA operations. The descriptor register in the PCI9056 is loaded with PCI and local addresses that are meaningless, a byte count of all zeros, and the descriptor pointer set up to point to the first entry in the chaining table.

### 4.10.3 DMA Channel Command Register

The Command Register is used to start/stop DMA operations and to monitor their progress. This register is a set of eight isolated bits as follows:

- Bit 0 is a channel enable bit. This bit should always be written as a one except in the unlikely event of wanting to pause or abort a DMA operation in progress, which one would ordinarily never do. When read, returns the bit value written.
- Bit 1 is the DMA Start Command bit. Write a 1 after the descriptor(s) have been set up to start a DMA operation. This bit is write-only and writing a zero has no effect.
- Bit 2 is the DMA Abort Command bit. Writing a 1 (with bit 0 cleared) terminates a DMA operation in progress. Ordinarily one would never do this, though. This bit is write-only and writing a zero has no effect.
- Bit 3 is the DMA Interrupt Acknowledge bit. Write a 1 in response to a DMA completion interrupt. This is the only way to clear the DMA bit of the Buffer Control register. This bit is write-only and writing a zero has no effect.
- Bit 4 is read-only. It returns 1 whenever there is no DMA operation in progress. Use this bit to monitor the progress of a DMA operation when running without using a DMA completion interrupt.
- Bits (5..7) are undefined.

## 4.11 Bit Error Rate Measurement

The LS-50 can perform simple Bit Error Rate (BER) measurements where a test loop is driven from the simulator and monitored by the decommutator. The LS-50 is equipped with a "sidelong" Pseudo Random Noise (PRN) pattern synchronizer. This circuit constantly attempts to lock to one of seven selected PRN patterns and counts any errors detected. The PRN synchronizer runs all the time and needs at least sixteen consecutive error-free bits to achieve lock, but thereafter can maintain lock unless the short-term BER exceeds $4x10^{-1}$.

Every second, when the decommutator clock counter updates, the accumulated error count is latched and can be read from the error count registers. Reading the register clears the error counter. The error count also includes several status bits as shown in Table 4-33 below. If the status bits do not show overflow or loss of sync, the BER can be calculated by dividing the clock count value into the error count value read during the same update.

The error count value is meaningless unless a well-formed PRN pattern is being received. The simulator has the capability of generating such a pattern, controlled by bits in the simulator Pattern register (Table 4-32). For compatibility with older boards, the 2T15 bit in the Mode register (Table 4-18 on page 93) is logically OR'ed with bit 1 of the Pattern

register. Normally the decommutator and simulator Pattern register values must match each other. The 2T15 bit in the simulator Mode register must match the state of the 2T15 bit in the decommutator Control register. Setting the BERT bit replaces the normal simulator NRZL and PCM outputs with the PRN pattern, and also causes a one-bit pulse on the simulator Frame Strobe output with each iteration of the pattern. Setting the ERR bit causes one bit of each iteration of the pattern to be wrong. With this bit set, and no other link errors, the BER should be 1 divided by the pattern length, e. g., 4.885 ´ 10-4 for a 2047-bit pattern. Setting the EVENT bit, then clearing it, causes a single error to be introduced into the pattern. This event is asynchronous, i. e., the error is not a specific bit.

**Table 4-32 PRN Pattern Registers**

| Bit | Mnemonic | Description |
|---|---|---|
| 2..0 | PATTERN | PRN pattern length, defined as:<br>000: $2^{11}$-1 (2047 bits) 100: $2^{19}$-1 (524,287 bits)<br>001: Reserved 101: $2^{21}$-1 (2,097,151 bits)<br>010: $2^{15}$-1 (32767 bits) 110: $2^{23}$-1 (8,388,607 bits)<br>011: $2^{17}$-1 (131,071 bits) 111: $2^{25}$-1 (33,554,431 bits) |
| 3 | REV | "Reverse." Moves inner tap selections away from the "big" end of the shift register toward the "little" end. |
| 7..4 | | Unused. |

**Table 4-33 Error Count High Register**

| Bit | Mnemonic | Description |
|---|---|---|
| 3..0 | | Bits [19..16] of the error count. |
| 4 | | Unused. |
| 5 | ECOVF | Set if the error counter overflowed during the last sample. |
| 6 | WOOS | Set if the PRN synchronizer lost lock during the last sample. If this bit is set the BER calculation may be grossly inaccurate. |
| 7 | OOS | This bit is **not** latched. It is set if the PRN synchronizer is presently out of lock. |

## 4.12 Channel 0 Daughtercard Interface

The "!PRES" bit in the Daughtercard Status Register will return zero if a daughtercard is present. This makes the other daughtercard registers meaningful. The Data Register is used to write a series of setup bytes to the daughtercard. Write by this sequence:

- Write the [next] setup byte to the Data Register.
- Write the Control Register with the !STROBE bit cleared. BUSY sets. Immediately write the control register again with !STROBE set.
- Wait until BUSY clears. Repeat if more to send.

| Table 4-34 Daughtercard Write Register Summary | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Register** | **#** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Daughtercard Data | 0E | See LS-40 Manual or paragraph 4.12.2 on page 114 | | | | | | | |
| Daughtercard Control | 0F | – | – | Source3..2 | | !INIT | !STB | Source1..0 | |

| Table 4-35 Daughtercard Read Register Summary | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Register** | **#** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Daughtercard Data | 0E | See LS-40 Manual or paragraph 4.12.2 on page 114 | | | | | | | |
| Daughtercard Status | 0F | 1 | READ | 0 | !Pres | 1 | BUSY | LOCK | SIG |

| Table 4-36 Daughtercard Control Register | | |
|---|---|---|
| **Bit** | **Mnemonic** | **Description** |
| 1..0 | SOURCE | LS-40 Bit Synchronizer Input source LSBs. See Table 4-38 on page 114. |
| 2 | !STROBE | Clearing the bit, then setting again indicates you wrote fresh data to the Command Register. Set by system reset. |
| 3 | !INIT | Daughtercard Reset when cleared. For compatibility with other systems. This bit has no meaning for Lumistar Daughtercards. Set by system reset. |
| 5..4 | SOURCE | LS-40 Bit Synchronizer Input source MSBs. See Table 4-38 on page 114. |
| 7..6 | | Not Used. |

| Table 4-37 Daughtercard Status Register | | |
|---|---|---|
| **Bit** | **Mnemonic** | **Description** |
| 0 | SIG | Signal Present. Set to indicate the input signal amplitude is valid. |
| 1 | LOCK | Bit Synchronizer Lock status. |
| 2 | BUSY | Set in response to !STROBE. Persists until the daughtercard is ready to accept another character. |
| 3 | 1 | Always 1. |
| 4 | !PRESENT | Set if NO Daughtercard installed. |
| 5 | SQUAL | For LS-40, set for estimated $E_b/N_o$ < 5dB |
| 6 | READ | Read back Operation in progress |
| 7 | 1 | Always 1. |

Certain commands cause the daughtercard to return a string of response bytes, including all LS-40 commands of the form 0xEn. If the daughtercard has a response, it will assert READ in the status register before it releases BUSY. To gather the string of response bytes, perform the following:

- Wait until BUSY clears. If READ is clear, skip out.
- Read the Data Register and save the value.
- Write the Control Register with the !STROBE bit cleared. BUSY sets. Immediately write the control register again with !STROBE set.

- Go back and wait on BUSY again.

### 4.12.1 Plug-and-Play

If "!PRESENT" is zero, the board can be queried for exactly what daughtercard is installed. To do so, send data bytes 0xED and 0x0A in succession. The daughtercard should assert READ and return six bytes. The third byte returned may be interpreted as (other values TBD):

- 0x31 – 10 MBPS LS-40 Bit Synchronizer.
- 0x32 – 20 MBPS LS-40 Bit Synchronizer.
- 0x38 – LS-38 70 MHz Demodulator/Bit Synchronizer.

### 4.12.2 LS-40 Bit Synchronizer Module

Setup and status responses for the LS-40 are defined in the LS-40 Technical Manual. Additionally, the Input Source fields in the Daughtercard Control Register are defined in Table 4-38 below.

**Table 4-38 LS-40 Bit Synchronizer Input Source**

| MSBs | LSBs | Description |
|---|---|---|
| 00 | 00 | J1-36 |
| 00 | 01 | J1-9 PCM Simulator Baseband output |
| 00 | 10 | J1-37 |
| 00 | 11 | J1-39 |
| 01 | 00 | J1-34 |
| 01 | 01 | J1-42 |
| 01 | 10 | J1-35 |
| 01 | 11 | J1-49 |
| 10 | 00 | Differential J1-34 – J1-36 |
| 10 | 01 | Differential J1-42 – J1-9 Simulator Output |
| 10 | 10 | Differential J1-35 – J1-37 |
| 10 | 11 | Differential J1-40 – J1-39 |

### 4.12.3 LS-38 70MHz Demodulator

The LS-38 is set up by writing a fourteen-byte command packet as defined in Table 4-39 below.

**Table 4-39 LS-38 Command Packet**

| Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0x10 | | | | | | | |
| 2 | Bit Synchronizer Loop Width:<br>00000 = 2% 01010 = 0.1%<br>00001 = 1% 11000 = 0.05%<br>01000 = 0.5% 11001 = 0.02%<br>01001 = 0.2% 11010 = 0.01% | | | | | 0 | Bit Rate [25..24] | |
| 3 | Bit Rate [23..16] | | | | | | | |
| 4 | Bit Rate [15..8] | | | | | | | |
| 5 | Bit Rate [7..0] | | | | | | | |
| 6 | FM Modulation Index (0000) | | | | 0 = FM. 1 = SOQPSK | | | |
| 7 | Demodulator Loop Bandwidth:<br>0x00 = 100 Hz 0x03 = 1000 Hz 0x06 = 10000 Hz<br>0x01 = 250 Hz 0x04 = 2500 Hz 0x07 = 25000 Hz<br>0x02 = 500 Hz 0x05 = 5000 Hz 0x08 = 50000 Hz | | | | | | | |
| 8 | – | – | – | – | – | DERAND | DPOL | CLKPOL |
| 9 | – | – | – | – | – | – | – | – |
| 10 | – | – | – | – | – | – | – | – |
| 11 | – | – | – | – | – | – | – | – |
| 12 | – | – | – | – | – | – | – | – |
| 13 | – | – | – | – | – | – | – | – |
| 14 | 0x0A | | | | | | | |

Setting DERAND turns on an RNRZ15 de-randomizer. Setting DPOL or CLKPOL inverts the output data or clock respectively.

The LS-38 can be caused to send a status response by sending data bytes 0x20 and 0x0A in succession. This will cause the LS-38 to return a fifteen-byte status packet defined in Table 4-40 below.

| Table 4-40 LS-38 Status Packet | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Byte** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 1 | 0x20 | | | | | | | |
| 2 | QUAL | LOCK | SIGNAL | – | – | – | Clock Count [25..24] | |
| 3 | Clock Count [23..16] | | | | | | | |
| 4 | Clock Count [15..8] | | | | | | | |
| 5 | Clock Count [7..0] | | | | | | | |
| 6 | FM Modulation Index = 0.02n + 0.56 | | | | – | – | – | – |
| 7 | Signal Quality | | | | | | | |
| 8 | Saturation Count | | | | | | | |
| 9 | – | – | – | – | – | Positive | SigStr[9..8] | |
| 10 | SigStr [7..0] | | | | | | | |
| 11 | 0 | | | | | | | |
| 12 | 0 | | | | | | | |
| 13 | 0 | | | | | | | |
| 14 | 0 | | | | | | | |
| 15 | 0 | | | | | | | |

QUAL, LOCK, SIGNAL reflect the state of the board status LED drivers. Signal Quality is a dimensionless number in the range 0 (very poor) to 255 (very good). Saturation Count is the count of input signal over-ranges in the last (**TBD** time period). SigStr is the estimated input level in tenths of dBm

## 4.13 Channel 1 Daughter-card Interface

If the P2 platform board is configured as a dual-decommutator, a second bit synchronizer daughter-card interface is provided through the J3 connector. This interface is not the same as the Channel 0 interface, but it "rhymes." The "!PRES" bit in the Channel 1 Daughter-card Status Register will return zero if a bit synchronizer is connected to J3. This makes the other daughter-card registers meaningful.

Communication with the Channel 1 bit synchronizer is analogous to that of the Channel 0 bit synchronizer, except the communication path is serial instead of parallel. For this reason, the Channel 1 Daughter-card Status Register has no BUSY bit. Setup and status data transferred are ytbd, but will probably also "rhyme" with the data transfers defined in the present iteration of the LS-40 PCM Bit Synchronizer documentation. Tentatively, the baud rate for serial data transfers has been set to 115.2K, making the proper setting for the RS-232 Baud Rate Register 0xFF.

| Table 4-41 Ch 1 Daughter-card Write Register Summary | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Register** | **#** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| RS232 Data | 21 | (ASCII Character) | | | | | | | |
| RS232 Baud Rate | 23 | 8 LSBs of –96 / (Baud Rate / 1200) | | | | | | | |

| Table 4-42 Ch 1 Daughter-card Read Register Summary | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Register** | **#** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Daughter-card Status | 0F | 1 | READ | Squal | !Pres | 1 | – | LOCK | SIG |
| RS232 Data | 21 | (ASCII Character) – top of a 64-byte FIFO | | | | | | | |
| RS232 Status | 23 | – | – | – | – | – | – | XBE | RBF |

| Table 4-43 Ch 1 Daughter-card Status Register | | |
|---|---|---|
| **Bit** | **Mnemonic** | **Description** |
| 0 | SIG | Signal Present. Set to indicate the input signal amplitude is valid. |
| 1 | LOCK | Bit Synchronizer Lock status. |
| 2 | | Not defined. |
| 3 | 1 | Always 1. |
| 4 | !PRESENT | Set if NO Daughter-card installed. |
| 5 | SQUAL | For LS40, set for estimated $E_b/N_o$ < 5dB |
| 6 | READ | Read back Operation in progress |
| 7 | 1 | Always 1. |