

DN 6000347

**TECHNICAL MANUAL
MODEL PCI 650**

• FRAME SYNCHRONIZATION VERIFICATION UNIT •

Acroamatics, Inc.

70 South Kellogg Avenue
Goleta, CA 93117-3476

April 1, 2000

PROPRIETARY NOTICE

Information contained in this document is disclosed in confidence and may not be duplicated in full or in part by any person without prior written approval by Acroamatics, Inc., except as provided by separate contractual agreement. Its sole purpose is to provide the user with adequately detailed documentation in order to install, operate, maintain, and order spare parts for the equipment supplied. The use of this document for any other purpose is expressly prohibited.

HOTLink™ is a registered trademark of Cypress Semiconductor Corp.

ACROAMATICS DOCUMENT HISTORY

The following table indicates major changes made to *Technical Manual for the Model PCI 650 Eight Stream PCM Decommutator*, Acroamatics Document Number 6000347, released on January 1, 2000, and contains a record of all revisions made since that date.

DN6000347 CHANGE HISTORY			
Rev	Date	Action	Name
	1-1-00	Original Issue	DJM
	02-08-08	Changed all DCOM refs to FSVU	DJM

**TECHNICAL MANUAL
MODEL PCI 650
• FRAME SYNCHRONIZATION VERIFICATION UNIT •**

TABLE OF CONTENTS

SECTION 1	INTRODUCTION	1-1
1.1	DESCRIPTION	1-1
1.2	PCM SYNCHRONIZATION	1-1
1.3	FRAME SYNCHRONIZATION MODES	1-2
1.4	FRAME SYNCHRONIZATION STRATEGY	1-3
1.5	BIT SLIP CORRECTION	1-5
1.6	DOCUMENT CONVENTIONS	1-5
SECTION 2	INSTALLATION	2-1
2.1	GENERAL	2-1
2.2	UNPACKING	2-1
2.3	FACTORY RETURN	2-1
2.4	INSTALLING	2-1
2.5	CONNECTORS	2-2
SECTION 3	OPERATION	3-1
3.1	INTRODUCTION TO PLX	3-1
3.2	PCI SYSTEM CONFIGURATION REGISTERS	3-2
3.2.1	Device Configuration Register Description	3-2
3.2.2	Mandatory Device Configuration Registers	3-2
3.2.2.1	Vendor ID Register	3-2
3.2.2.2	Device ID Register	3-2
3.2.2.3	Command Register	3-2
3.2.2.4	Status Register	3-5
3.2.2.5	Revision ID Register	3-6
3.2.2.6	Class Code Register	3-6
3.2.2.7	Header Type Register	3-6
3.3	OPTIONAL CONFIGURATION REGISTERS	3-6
3.3.1	Cache Line Size Register	3-6
3.3.2	Latency Timer: Timeslice Register	3-7
3.3.3	BIST Register	3-7
3.3.4	Base Address Registers	3-7
3.3.5	Memory Base Address Register	3-8
3.3.6	I/O Base Address Register	3-9
3.3.7	Determining Block Size and Assigning Address Range	3-9
3.3.8	Expansion ROM Base Address Register	3-10
3.3.9	CardBus CIS Pointer	3-10
3.3.10	Subsystem Vendor ID and Subsystem ID Registers	3-10
3.3.11	Interrupt Pin Register	3-11
3.3.12	Interrupt Line Register	3-11
3.3.13	Min_Gnt Register: Timeslice Request	3-11
3.3.14	Max_Lat Register: Priority-Level Request	3-11
3.4	PLX9054 CONTROL REGISTER ACCESS	3-12
3.4.1	Local Configuration Registers	3-12

3.4.2 Runtime Registers	3-12
3.4.3 DMA Control Registers	3-12
3.5 PCI FSVU CONTROL REGISTERS	3-15
3.5.1 Signature Register	3-15
3.5.2 Control Register	3-15
3.5.3 PCM Status Registers	3-15
3.5.4 Sync Strategy Register	3-17
3.5.4.1 Verify to Search Count (Bits 0-3)	3-17
3.5.4.2 Verify to Lock Count (Bits 4-7)	3-17
3.5.4.3 Lock to Search Count (Bits 8-11)	3-17
3.5.4.4 Error Tolerance Count (Bits 12-15)	3-18
3.5.4.5 Sync Pattern Length (Bits 16-22)	3-18
3.5.4.6 Bit Slip Window (Bits 24-25)	3-18
3.5.4.7 Input Polarity (Bits 26-27)	3-18
3.5.5 Sync Mode (bits 28-29)	3-18
3.5.5.1 Fixed Mode	3-18
3.5.5.2 Adaptive Mode	3-19
3.5.5.3 Burst Mode	3-19
3.5.6 Frame Sync Pattern Format Definition (Bits 30-31)	3-19
3.5.6.1 Normal	3-19
3.5.6.2 Alternate Complement	3-20
3.5.6.3 Frame Complement Sync	3-20
3.5.6.4 Sync Pattern	3-20
3.5.6.5 Sync Pattern Mask	3-20
3.5.6.6 Bits Per Frame	3-20
3.5.7 Serial Data Registers	3-20
SECTION 4 THEORY OF OPERATION	4-1
4.1 BLOCK DIAGRAM	4-1
SECTION 5 ADJUSTABLE SWITCH & JUMPER SETTINGS	5-1
5.1 DESCRIPTION	5-1
SECTION 6 DRAWINGS	6-1
6.1 INTRODUCTION TO THE DRAWINGS	6-1
6.1.1 Drawing System	6-1
6.1.2 Drawing Package Organization	6-1
6.1.3 Programmed Parts	6-2
SECTION 7 SCHEMATICS	7-1
7.1 INTRODUCTION	7-1
7.1.1 Schematic Conventions	7-1
7.1.2 Troubleshooting	7-1

TECHNICAL MANUAL
MODEL PCI 650

• FRAME SYNCHRONIZATION VERIFICATION UNIT •

SECTION 1 INTRODUCTION

1.1 DESCRIPTION

The *Model PCI 650 FSVU (Frame Synchronization Verification Unit)* contains eight PCM Decommutators that are designed for PCM stream quality verification rather than data processing. Each decommutator contains a minor frame synchronizer with a 64 bit pattern correlator, a 16 bit counter that counts the number of bits per frame, and a programmable synchronizer strategy providing Search, Verify, and Lock states. A programmable watchdog timer returns a decommutator to Search if the input clock is lost. You can read the status of each of the eight decommutators over the PCI bus to determine the quality of the input data to each channel. Figure 1 is a simplified block diagram of the card.

1.2 PCM SYNCHRONIZATION

You decommutate TDM (Time Division Multiplexed) data by first locating a fixed pattern and then determining that the pattern repeats at fixed intervals. The pattern is recognized by a digital correlator that accepts 64 sequential bits and compares them to a programmed reference pattern using a programmable mask to exclude *don't care* bit positions.

A minor frame is defined as *the binary values contained between two frame sync patterns*. You write the reference *frame sync pattern* and *mask* to each decommutator as two 64-bit words. Exactly 64 bits must be written into the digital correlator regardless of the actual sync pattern length. To achieve this the software extends any shorter patterns to 64 bits by adding *don'tcare* bits ahead of the first bit of the sync pattern, and then processing the data from the input to the 64 bit correlator register. This defines the minor frame as *starting with the bit following the sync pattern*.

Data within each frame is normally syllabized into words of fixed or variable length that are coded in some binary representation of the initial data. The sync pattern is different than the data words in that it is not a number in the quantitative sense, since all bits have the same weight. The sync pattern is further distinguished from a number in that it may or may not contain embedded don't care bits. Don't care bits, despite being within the pattern, may not have a consistent value in each minor frame and therefore cannot be used for sync pattern recognition. You remove the don't care bits from the comparison with the mask word. Only those bits that have their corresponding mask bits set to 1 are tested.

Because telemetry data is often transmitted or stored imperfectly due to system noise constraints, absolute correlation is not always possible. When determining the sync pattern location, you must often allow a programmable number of conflicts to occur in an otherwise acceptable pattern. This number is referred to as the *sync pattern tolerance*. The Model 650 provides sync pattern tolerances

from zero to fifteen bits.

The correlator also recognizes an acceptable *complement sync* pattern. The complement sync function handles *complement frame sync* formats where the major frame is synchronized by complementing the minor frame sync pattern, as well as *alternate complement* frame synchronization, where the frame sync pattern is complemented on alternate minor frames. The complement sync pattern is also used in conjunction with *automatic polarity resolution*.

1.3 FRAME SYNCHRONIZATION MODES

The FSVU has several pattern recognition modes you can use to acquire main-frame synchronization. In Complement Frame Sync mode, the synchronizer accepts one occurrence of the complement sync pattern preceded and followed by normal sync patterns. This prevents the synchronizer from detecting an error when the *Major Frame* sync word is encountered. In Alternating Complement Sync mode the synchronizer accepts the first appearance of either the normal or complement of the pattern, and then conditions itself to detect the complement of that pattern at the next occurrence. The successive occurrence of two patterns of the same polarity is treated as an error. In Automatic Polarity Correction/Search mode, the synchronizer requires the appearance of *two consecutive* sync patterns of either the normal or complement of the sync pattern, and adjusts the polarity of the entire input stream based upon the detected sync pattern polarity. In VERIFY or LOCK the detection of two consecutive complemented sync patterns within the End Of Frame window will complement the input polarity. This mode enables the system to correct for phase inversion in the input PCM stream.

The main problem when decommutating TDM data is in finding and maintaining the sync pattern location in noise-contaminated data. This problem is further complicated by the occurrence of data segments in the minor frame that have the same value as the sync pattern. The occurrence of these false sync patterns is increased as the correlator sync error tolerance is increased. The method of finding the sync location and minimizing the time lost when a false sync is encountered is accomplished by *Sync Strategy* logic. The sync strategy is programmable and offers several modes, each offering different advantages, depending upon the PCM format, the quality of the sync pattern, and the amount of noise contamination.

Frame synchronization normally consists of first looking for a pattern anywhere in the stream while in Search, and then applying a frame length window to avoid false sync patterns that may occur in the data stream. The frame is normally of fixed length and the pattern should recur at a specific bit interval. However, when the input to the bit synchronizer lacks sufficient transitions, and/or contains excessive noise, the bit synchronizer clock may drift and produce excessive or insufficient clock pulses during the frame. This results in a *frame bit slip* condition, which can result in the loss of synchronization.

The Bit Slip Window of 1, 2, or 3 bits may correct for the resultant bit sync error. However, if the sync pattern is not properly chosen (when the telemetry encoding was designed), the pattern, shifted by several bit positions with random data, may look like an acceptable pattern (especially when the sync pattern error tolerance is large) and the results may resynchronize the frame erroneously. Therefore, the frame synchronizers must provide a programmable sync strategy that will achieve rapid synchronization while, at the same time, guarding against

false sync patterns produced by the variable patterns encountered in the PCM stream.

1.4 FRAME SYNCHRONIZATION STRATEGY

The FSVU provides reliable frame synchronization and data decommutation by using programmable frame sync strategy counters, a bits-per-minor-frame counter, a programmable bit-slip window, and programmable bit error tolerances.

Frame synchronization occurs during all three states of Search, Verify, and Lock. The FSVU includes **Fixed, Adaptive, & Burst** mode strategies. The Fixed strategy tests for a programmable number of good or bad sync patterns to determine when state changes should occur. The Adaptive strategies are for use with noisy data when the sync pattern *error tolerance* is enabled. This strategy tests the number of errors in the sync pattern and uses this value, relative to previous values, to qualify a sync pattern. You can program the sync strategy to transfer from Search to Verify or Lock. When the transfer is to Verify, you can program additional *flywheel* frames before the strategy returns to Search, and the number of consecutive good frames before the synchronizer will go into Lock. This adds confidence that the real sync pattern has been found. In Lock, you can program the number of consecutive bad frames before the synchronizer leaves Lock. The transfer from Lock is directly to Search, thus allowing rapid sync acquisition.

The **Fixed** strategy starts in Search, where the bit stream is scanned for the programmed sync pattern. Upon detecting a good pattern (as qualified by the sync error tolerance) the synchronizer enters the Verify state. A window is generated at the end of the minor frame by the bits per frame count and the bit slip window. If a good pattern is found within this window the consecutive *Good Frames from Verify to Lock* count (0 to 15) is tested and, if equal, the synchronizer will enter Lock. The count of zero allows the strategy to transfer directly from Search to Lock, bypassing the Verify state. When a good pattern is not found within the window, the consecutive *Bad Frames from Verify to Search* (1 - 15) is tested and, if equal, the synchronizer will return to Search. In Lock the correlator pattern is tested in the *End Of Frame* window. If the number of consecutive frames with bad patterns matches the *Lock to Search* count (1 - 15), the synchronizer will return to Search as selected by setup information.

The **Adaptive** strategy functions as follows. In Search, the frame sync unit will begin by searching for the frame sync pattern that meets the programmed frame sync error tolerance. When a candidate pattern occurs, the number of detected errors in the frame sync pattern will replace the contents of the sync error tolerance register. The strategy sequencer remains in Search for one minor frame as established by the bits per frame count. The input stream is tested continually for a pattern with fewer errors than those stored, and if a sync pattern with less errors is found before end of frame, the bits per frame counter is re-initiated and the new pattern error count replaces the sync pattern error tolerance. When a frame passes that does not contain a pattern that is better than the current sync pattern tolerance, and the pattern at the end of frame is acceptable, the frame sync unit will advance to Verify. The strategy now works as in fixed mode, with the error tolerance equal to the number of errors detected in the best pattern encountered in Search. If the required number of good patterns are found in the frame sync window, the frame sync unit will advance to Lock. If the frame

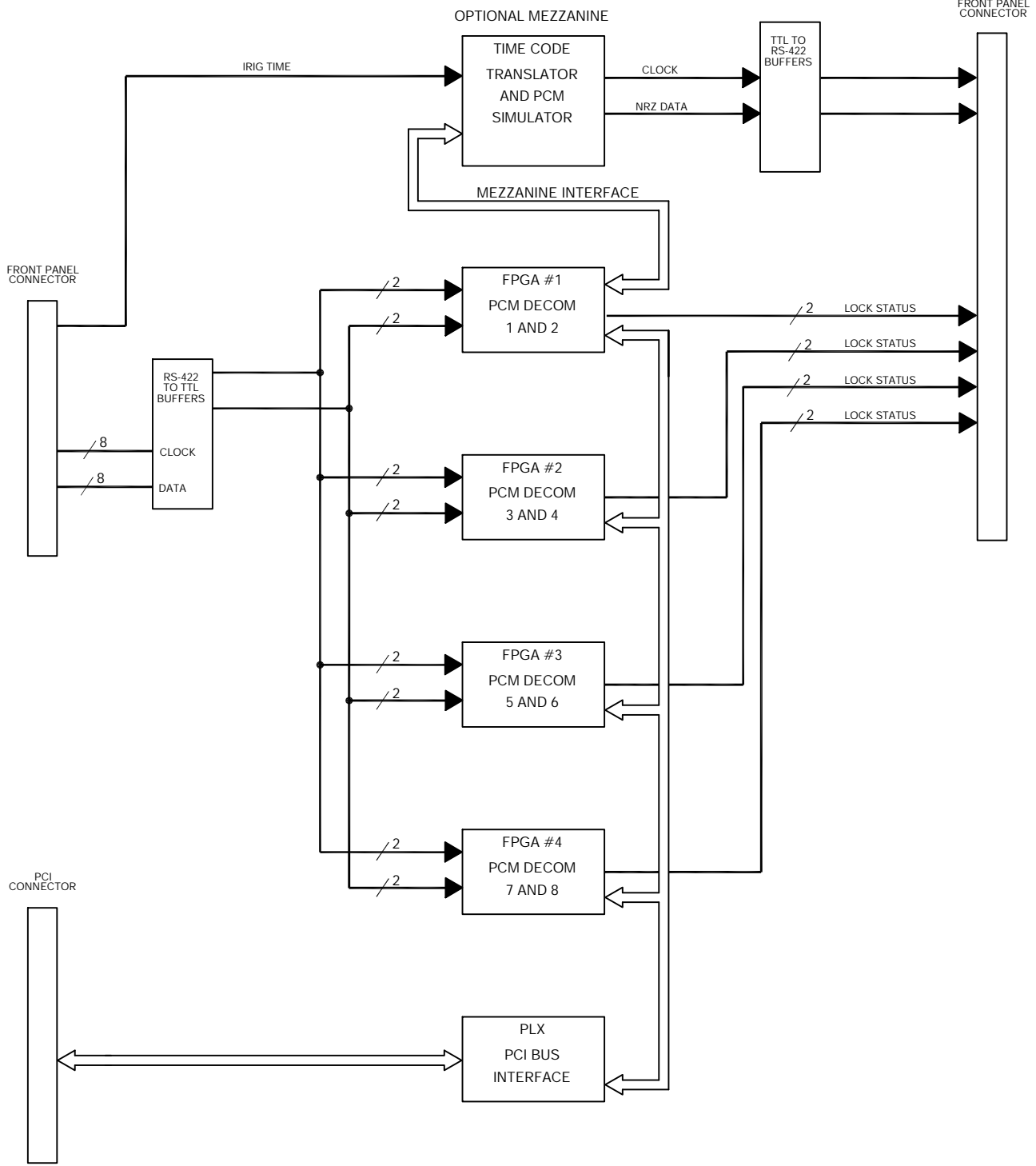


FIGURE 1 8 STREAM PCM DECOMMUTATOR BLOCK DIAGRAM

sync errors are greater than the tolerance established (when leaving Search), for the consecutive number of frames specified by the Verify to Search count (when in Verify), or the Lock to Search count (when in Lock), the frame sync unit reverts back to Search mode. When Search mode is re-entered, the sync error tolerance is set to the initially programmed value and the pattern search proceeds as previously described.

The **Burst** mode strategy is for synchronizing to formats that have variable length frame structures. In Search mode the frame sync unit continually searches the incoming data stream for a frame sync pattern that meets the frame sync error allowance. Once an acceptable pattern is detected the frame sync logic loads the *Bits Per Frame* counter and advances from Search to Lock. The Verify to Lock count is set to zero in Burst mode, bypassing the Verify state. Upon detecting a frame sync pattern with errors above the allowable limit at the normal End Of Frame location (dependent on the *Slip Window* and the *Bits Per Frame* settings), the frame sync unit falls back to Verify mode. In Verify mode the Bits Per Frame counter is re-initialized, and the fill data at the End Of Frame is continuously searched for an acceptable sync pattern. If one is found, the Bits Per Frame counter is loaded and the synchronizer returns to Lock. When the expected End Of Frame is detected (as determined by the Bits Per Frame counter) and an acceptable sync pattern has not been found, the synchronizer returns to Search as directed by the Verify to Search count. The transfer from Lock is to Verify rather than Search as in the Fixed and Adaptive modes, and the count is preset to 1. The Verify to Search count determines the number of frame intervals in which the strategy will remain in Verify mode.

1.5 BIT SLIP CORRECTION

Bit slip Correction permits you to program the frame synchronizer to accept sync patterns occurring in bit positions adjacent to the expected position in the frame. The sync pattern may occur *exactly* at the expected location (one bit window); *one* bit position early or late (three bit window), *two* bit positions prior or after the expected position (five bit window), or *three* bit positions prior or after the expected position (seven bit window), and still be detected as an acceptable sync pattern. This feature enables the unit to maintain synchronization during excessive noise bursts or data dropouts in the input stream when the bit synchronizer cannot maintain synchronization with the PCM stream.

1.6 DOCUMENT CONVENTIONS

In this document, register addresses and address offsets are hexadecimal numbers. Where it is necessary to refer to a hexadecimal number in the text, we use the C programming convention *0xNN* to refer to hexadecimal number *NN*. Bits in a register are numbered in decimal. The term *Device n* refers to an address destination on the TDP system A-Bus. The A-bus is the output data bus, and has eight possible destination devices. Although we do not use all eight, those destinations we do use are dedicated to specific functions in the TDP system. The term *DIT* refers to a data message in the TDP system. It stands for "Data, ID, and Time," the three components of a TDP data message. We can label a DIT by the value of its ID tag, for example "DIT 0xFFF1," the DIT that conveys the once per millisecond value of a time message. Frequently we use a functional label instead, for example, "the MILLISECOND DIT".

SPECIFICATIONS

MODEL PCI 650 - FRAME SYNCHRONIZATION VERIFICATION UNIT

FEATURE	DESCRIPTION
INPUT	
Sources	Eight channels, each accepting RS-422 differential 0° clock & NRZ data
Impedance	120 Ω input impedance, TTL compatible
Bit Rate	Up to 32 Megabits per second
Polarity	Programmable, with automatic polarity correction
Frame Length	Programmable, 16 to 16384 bits
SYNCHRONIZATION	
Mainframe Sync	Provides for programmable sync pattern and mask, complement pattern recognition, and variable length frame decommutation. Pattern length up to 64 bits.
Alternate Complement Sync	Synchronizes to formats in which the minor frame sync pattern is complemented on alternate frames
Complement Frame Sync	Synchronizes to formats that complement the minor frame sync pattern at a major frame rate
Automatic Polarity Inversion	Input polarity is inverted when two consecutive complemented sync patterns are found
Sync Modes	Fixed, Adaptive, and Burst
Sync Strategy	Search, Verify, and Lock
Sync Error Tolerance	0 to 15 errors, programmable
Sync Slip Window	0, ±1, ±2, & ±3 bits, programmable
Data Polarity	Normal, Inverted, and Automatic detection
Clock Rate Monitor	A programmable delay counter is provided to return the synchronizer to Search if the clock input is lost
OUTPUTS	
Time	The Time Code Translator can be read from the PCI bus.
PCM Status	A status word is available for each PCM frame synchronizer via the PCI bus.
Discrete Status	The Lock status of each frame synchronizer is output as an RS-422 signal.
Serial Setup Output	A serial RS-422 output allows you to send data from the PCM bus to an external device.
Mezzanine Card	A mezzanine connector supports an optional Time Code Translator/Generator (with or without PCM/PAM Simulator). The mezzanine card provides the following signals:
IRIG B In	Amplitude modulated IRIG A, B, or G with 100mV to 10 V peak to peak signal input amplitude. Simulator output is RS-422 NRZ-L data and 0° clock.
REQUIREMENTS	
Power	+5VDC at 1A, +3.3VDC at 2A
Temperature	Operating: 0 to +40°C; non-operating: -40 to +86°C
Relative Humidity	Up to 90% non-condensing
Air Flow	30 linear FPM
Shock	Operating: 6G; non-operating: 50G
Vibration	Operating: 0.5G, 5 to 2000 Hz; non-operating: 1.2G, 5 to 500 Hz

Specifications are subject to change without notice.

ACROAMATICS, INC. PRODUCT WARRANTY

Acroamatics, Inc. warrants all equipment manufactured by it to be free from defects in design, materials, and workmanship for a period of twelve months from the date of acceptance by the Customer. Acroamatics, Inc. will repair without charge all parts of said products that are returned to the factory within the warranty period, provided that the equipment is returned prepaid to the factory within twelve months after the date of acceptance, or that the defect is reported, in writing, within twelve months after the date of acceptance, and provided that inspection by Acroamatics, Inc. discloses that the defects are as above specified. Equipment found to be defective will, at Acroamatics, Inc.'s option, be replaced or repaired, and returned via surface transportation, prepaid. With the exception of the twelve-month warranty set forth above, Acroamatics, Inc. makes no express warranties, no warranties of merchantability, and no warranties that extend beyond the description on the face hereof. In no event will Acroamatics, Inc. be liable for consequential damages of any kind. For products sold to the United States Government, under procurements governed by the Federal Acquisition Regulations (FAR), any part of this PRODUCT WARRANTY that conflicts with an applicable FAR clause, incorporated actually or by reference in the purchase contract, shall be supplanted by the FAR clause. Acroamatics, Inc.

SECTION 2 INSTALLATION

2.1 GENERAL

This section contains installation information for the Acroamatics Model 650 PCI Frame Synchronization Verification Unit (FSVU). The card part number is 6011650.

2.2 UNPACKING

Using proper ESD-protection procedures, open the cardboard shipping container and remove the card from the anti-static bag. Retain the container, anti-static bag, and foam packaging material for use if you must return the card.

2.3 FACTORY RETURN

When you return a card to the factory for repair or modification, include as much information as possible describing the failure mode or the modification/update you want.

Pack the card for shipment by wrapping it in the anti-static bag. Place the card into the shipping container, protecting it with the foam packing, and secure the container with reinforced tape. Provide the name and phone number of a technical contact we can talk to regarding the card.

Call Acroamatics at (805) 967-9909 to get a RMA number before returning any equipment to the factory, and include the RMA number in any correspondence or shipments to Acroamatics.

2.4 INSTALLING

The FSVU card mounts in a standard PCIbus slot. Mounting dimensions are shown in the assembly Drawing in Section 6 of this manual. Slide the card into one of your empty PCI slots and seat the card firmly. Screw down the card. Make the front panel cable connections appropriate to your system. Remove the PCI card by pulling firmly on the outside of the ears.

2.5 CONNECTORS

The following pages contain tables of information on all the connections into and out of the Model 650 FSVU.

TABLE 2-1 CONNECTOR LIST FOR MODEL 650		
CONN	FUNCTION	DESCRIPTION
P01-A	PCI Bus Connections	
P01-B	PCI Bus Connections	
J01	Mezzanine Connection	
RC1	External Interface	
TC1	Test Connector 1	
TC2	Test Connector 2	
TC3	Test Connector 3	
TC4	Test Connector 4	
TC5	Consult Factory	
TC6	Consult Factory	
TC7	Consult Factory	
TC8	Consult Factory	

**TABLE 2-2
CONNECTOR P01-A**

Note: (\$) signals not used on this card

PIN	SIGNAL	FUNCTION
01	TRST# (\$)	Test Reset
02	+12V	+12 Volts
03	TMS (\$)	Test Mode Select
04	4TDIOST	Test Input
05	+5V	+5 Volts
06	9PCIINTA	Interrupt A
07	INTC# (\$)	Interrupt C
08	+5V	+5 Volts
09	Reserved	
10	+Vi/o	
11	Reserved	
12		
13		
14	Reserved	
15	9PCIRST	Reset Signal
16	+Vi/o (\$)	
17	PCIGNT	Grant
18	GND	Ground
19	Reserved	
20	4PCIAD30	Address Data Bus Bit 30
21	+3.3V	+3.3 Volts
22	4PCIAD28	Address Data Bus Bit 28
23	4PCIAD26	Address Data Bus Bit 26
24	GND	Ground
25	4PCIAD24	Address Data Bus Bit 24
26	4PCIISL	Initialization Device Select
27	+3.3V	+3.3 Volts
28	4PCIAD22	Address Data Bus Bit 22
29	4PCIAD20	Address Data Bus Bit 20
30	GND	Ground
31	4PCIAD18	Address Data Bus Bit 18
32	4PCIAD16	Address Data Bus Bit 16

TABLE 2-2 (continued)		
CONNECTOR P01-A		
PIN	SIGNAL	FUNCTION
33	+3.3V	+3.3 Volts
34	9PCIFRM	Cycle Frame
35	GND	Ground
36	9PCITRY	Target Ready
37	GND	Ground
38	9PCISTP	Stop
39	+3.3V	+3.3 Volts
40	SDONE (\$)	Snoop Done
41	SBO# (\$)	Snoop Back Off
42	GND	Ground
43	4PCIPAR	Parity
44	4PCIAD15	Address Data Bus Bit 15
45	+3.3V	+3.3 Volts
46	4PCIAD13	Address Data Bus Bit 13
47	4PCIAD11	Address Data Bus Bit 11
48	GND	Ground
49	4PCIAD09	Address Data Bus Bit 09
50		
51		
52	9PCICBE0	Data path 0, AD[7:0]
53	+3.3V	+3.3 Volts
54	4PCIAD06	Address Data Bus Bit 06
55	4PCIAD04	Address Data Bus Bit 04
56	GND	Ground
57	4PCIAD02	Address Data Bus Bit 02
58	4PCIAD00	Address Data Bus Bit 00
59	+Vi/o (\$)	
60	REQ64# (\$)	Request 64-bit Transfer
61	+5V	+5 Volts
62	+5V	+5 Volts

TABLE 2-3 CONNECTOR P01-B		
Note: (\$) signals not used on this card		
PIN	SIGNAL	FUNCTION
01	-12V	-12 Volts
02	TCK (\$)	Test Clock
03	GND	Ground
04	4TDIOST	Test Output
05	+5V	+5 Volts
06	+5V	+5 Volts
07	INTB# (\$)	Interrupt B
08	INTD# (\$)	Interrupt D
09	PRSNT1#	Card Present #1
10	Reserved (\$)	
11	PRSNT2#	Card Present #2
12		
13		
14	Reserved (\$)	
15	GND	Ground
16	PCICLK	PCI Clock Signal
17	GND	Ground
18	9PCIREQ	Request
19	+Vi/o (\$)	
20	4PCIAD31	Address Data Bus Bit 31
21	4PCIAD29	Address Data Bus Bit 29
22	GND	Ground
23	4PCIAD27	Address Data Bus Bit 27
24	4PCIAD25	Address Data Bus Bit 25
25	+3.3V	+3.3 Volts
26	9PCICBE3	Data Path 3, AD[31:24]
27	4PCIAD23	Address Data Bus Bit 23
28	GND	Ground
29	4PCIAD21	Address Data Bus Bit 21
29	4PCIAD19	Address Data Bus Bit 19
31	+3.3V	+3.3 Volts
32	4PCIAD17	Address Data Bus Bit 17

TABLE 2-3 (continued)		
CONNECTOR P01-B		
PIN	SIGNAL	FUNCTION
33	9PCICBE2	Data Path 2, AD[23:16]
34	GND	Ground
35	9PCIIRY	Initiator Ready
36	+3.3V	+3.3 Volts
37	9PCIDSL	Device Select
38	GND	Ground
39	9PCILCK	Lock
40	9PCIPERR	Data Parity Error
41	+3.3V	+3.3 Volts
42	9PCISERR	System Error
43	+3.3V	+3.3 Volts
44	9PCICBE1	Data Path 1, AD[15:08]
45	4PCIAD14	Address Data Bus Bit 14
46	GND	Ground
47	4PCIAD12	Address Data Bus Bit 12
48	4PCIAD10	Address Data Bus Bit 10
49	GND	Ground
50		
51		
52	4PCIAD08	Address Data Bus Bit 08
53	4PCIAD07	Address Data Bus Bit 07
54	+3.3V	+3.3 Volts
55	4PCIAD05	Address Data Bus Bit 05
56	4PCIAD03	Address Data Bus Bit 03
57	GND	Ground
58	4PCIAD01	Address Data Bus Bit 01
59	+Vi/o (\$)	
60	ACK64# (\$)	Acknowledge 64-bit Transfer
61	+5V	+5 Volts
62	+5V	+5 Volts

**TABLE 2-4
CONNECTOR J01**

PIN	SIGNAL	FUNCTION
01	+5V	+5 Volts
02	GND	Ground
03	+5V	+5 Volts
04	GND	Ground
05	+3.3V	+3.3 Volts
06	GND	Ground
07	+3.3V	+3.3 Volts
08	GND	Ground
09	4IRGIN	IRIG Time Input
10		
11	GND	Ground
12		
13	GND	Ground
14		
15		
16	GND	Ground
17		
18		
19	GND	Ground
20		
21		
22	GND	Ground
23		
24		
25	GND	Ground
26	4SIMCLK	Simulator Clock (+) RS-422
27	4SIMCLK	Simulator Clock (-) RS-422
28	GND	Ground
29	4SIMDAT	Simulator Data (+) RS-422
30	4SIMDAT	Simulator Data (-) RS-422
31	GND	Ground
32		

TABLE 2-4 (continued)		
CONNECTOR J01		
PIN	SIGNAL	FUNCTION
33		
34	GND	Ground
35	GND	Ground
36	GND	Ground
37	TIMPUL	Control Pull Up
38		
39	TIMPUL	Control Pull Up
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50	GND	Ground
51		
52	LOCRST-	Local Reset-
53		
54	TIMEDS-	Time Data Strobe-
55		
56	LOCWRT	Local Write
57		
58	LOCAD1	Local Address Bit 1
59		
60	LOCAD2	Local Address Bit 2
61		
62	LOCAD3	Local Address Bit 3
63		
64	LOCAD4	Local Address Bit 4
65		
66	LOCBS0	Local Data Bus Bit 00
67		
68	LOCBS1	Local Data Bus Bit 01
69		

TABLE 2-4 (continued)		
CONNECTOR J01		
PIN	SIGNAL	FUNCTION
70	LOCBS2	Local Data Bus Bit 02
71		
72	LOCBS3	Local Data Bus Bit 03
73		
74	LOCBS4	Local Data Bus Bit 04
75		
76	LOCBS5	Local Data Bus Bit 05
77		
78	LOCBS6	Local Data Bus Bit 06
79		
80	LOCBS7	Local Data Bus Bit 07
81		
82	LOCBS8	Local Data Bus Bit 08
83		
84	LOCBS9	Local Data Bus Bit 09
85		
86	LOCBS10	Local Data Bus Bit 10
87		
88	LOCBS11	Local Data Bus Bit 11
89		
90	LOCBS12	Local Data Bus Bit 12
91		
92	LOCBS13	Local Data Bus Bit 13
93		
94	LOCBS14	Local Data Bus Bit 14
95		
96	LOCBS15	Local Data Bus Bit 15
97	-12V	-12 Volts
98	GND	Ground
99	+12V	+12 Volts
100	GND	Ground

**TABLE 2-5
CONNECTOR RC1**

PIN	SIGNAL	FUNCTION
01	GND	Ground
02	GND	Ground
03	4FCLK1	Sync 1 Clock + [RS-422]
04	9FCLK1	Sync 1 Clock - [RS-422]
05	4FCLK2	Sync 2 Clock + [RS-422]
06	9FCLK2	Sync 2 Clock - [RS-422]
07	4FCLK3	Sync 3 Clock + [RS-422]
08	9FCLK3	Sync 3 Clock - [RS-422]
09	4FCLK4	Sync 4 Clock + [RS-422]
10	9FCLK4	Sync 4 Clock - [RS-422]
11	4FCLK5	Sync 5 Clock + [RS-422]
12	9FCLK5	Sync 5 Clock - [RS-422]
13	4FCLK6	Sync 6 Clock + [RS-422]
14	9FCLK6	Sync 6 Clock - [RS-422]
15	4FCLK7	Sync 7 Clock + [RS-422]
16	9FCLK7	Sync 7 Clock - [RS-422]
17	4FCLK8	Sync 8 Clock + [RS-422]
18	9FCLK8	Sync 8 Clock - [RS-422]
19	4FDAT1	Sync 1 Data + [RS-422]
20	9FDAT1	Sync 1 Data - [RS-422]
21	4FDAT2	Sync 2 Data + [RS-422]
22	9FDAT2	Sync 2 Data - [RS-422]
23	4FDAT3	Sync 3 Data + [RS-422]
24	9FDAT3	Sync 3 Data - [RS-422]
25	4FDAT4	Sync 4 Data + [RS-422]
26	9FDAT4	Sync 4 Data - [RS-422]
27	4FDAT5	Sync 5 Data + [RS-422]
28	9FDAT5	Sync 5 Data - [RS-422]
29	4FDAT6	Sync 6 Data + [RS-422]
30	9FDAT6	Sync 6 Data - [RS-422]
31	4FDAT7	Sync 7 Data + [RS-422]
32	9FDAT7	Sync 7 Data - [RS-422]
33	4FDAT8	Sync 8 Data + [RS-422]
34	9FDAT8	Sync 8 Data - [RS-422]

TABLE 2-5 (continued)		
CONNECTOR RC1		
PIN	SIGNAL	FUNCTION
35	GND	Ground
36	GND	Ground
37	4STAT1	Status Bit 1
38	GND	Ground
39	4STAT2	Status Bit 2
40	GND	Ground
41	4STAT3	Status Bit 3
42	GND	Ground
43	4STAT4	Status Bit 4
44	GND	Ground
45	4STAT5	Status Bit 5
46	GND	Ground
47	4STAT6	Status Bit 6
48	GND	Ground
49	4STAT7	Status Bit 7
50	GND	Ground
51	4STAT8	Status Bit 8
52	GND	Ground
53	GND	Ground
54	4IRIGIN	IRIG Time Input
55	GND	Ground
56	GND	Ground
57	4SIMCLK	Simulator Clock + [RS-422]
58	9SIMCLK	Simulator Clock - [RS-422]
59	4SIMDAT	Simulator Data + [RS-422]
60	9SIMDAT	Simulator Data - [RS-422]
61	4SDATIN	Serial Data In + [RS-422]
62	9SDATIN	Serial Data In - [RS-422]
63	4SIMCLK	Serial Data Out + [RS-422]
64	9SIMCLK	Serial Data Out - [RS-422]
65	GND	Ground
66	GND	Ground
67		
68	LOCRST-	Local Reset -

TABLE 2-6 CONNECTOR TC1		
PIN	SIGNAL	FUNCTION
01	4CLKS1	Decom 1 Clock
02	4DATS1	Decom 1 Data
03	4FRMS1	Decom 1 Frame Time
04	4CLKS2	Decom 2 Clock
05	4DATS2	Decom 2 Data
06	4FRMS2	Decom 2 Frame Time
07	GND	Ground

TABLE 2-7 CONNECTOR TC2		
PIN	SIGNAL	FUNCTION
01	4CLKS3	Decom 3 Clock
02	4DATS3	Decom 3 Data
03	4FRMS3	Decom 3 Frame Time
04	4CLKS4	Decom 4 Clock
05	4DATS4	Decom 4 Data
06	4FRMS4	Decom 4 Frame Time
07	GND	Ground

TABLE 2-8 CONNECTOR TC3		
PIN	SIGNAL	FUNCTION
01	4CLKS7	Decom 7 Clock
02	4DATS7	Decom 7 Data
03	4FRMS7	Decom 7 Frame Time
04	4CLKS8	Decom 8 Clock
05	4DATS8	Decom 8 Data
06	4FRMS8	Decom 8 Frame Time
07	GND	Ground

TABLE 2-9 CONNECTOR TC4		
PIN	SIGNAL	FUNCTION
01	4CLKS5	Decom 5 Clock
02	4DATS5	Decom 5 Data
03	4FRMS5	Decom 5 Frame Time
04	4CLKS6	Decom 6 Clock
05	4DATS6	Decom 6 Data
06	4FRMS6	Decom 6 Frame Time
07	GND	Ground

TABLE 2-10 CONNECTOR TC5 (consult Factory)		
PIN	SIGNAL	FUNCTION
01	+3.3V	+3.3 Volts
02	AC1TMS	
03	AC1TD1	
04	AC1TCK	
05	AC1PRA	
06	AC1TPRB	
07	GND	Ground

TABLE 2-11 CONNECTOR TC6 (consult Factory)		
PIN	SIGNAL	FUNCTION
01	+3.3V	+3.3 Volts
02	AC2TMS	
03	AC2TD1	
04	AC2TCK	
05	AC2PRA	
06	AC2TPRB	
07	GND	Ground

TABLE 2-12 CONNECTOR TC7 (consult Factory)		
PIN	SIGNAL	FUNCTION
01	+3 . 3V	+3.3 Volts
02	AC3TMS	
03	AC3TD1	
04	AC3TCK	
05	AC3PRA	
06	AC3TPRB	
07	GND	Ground

TABLE 2-13 CONNECTOR TC8 (consult Factory)		
PIN	SIGNAL	FUNCTION
01	+3 . 3V	+3.3 Volts
02	AC4TMS	
03	AC4TD1	
04	AC4TCK	
05	AC4PRA	
06	AC4TPRB	
07	GND	Ground

SECTION 3 OPERATION

3.1 INTRODUCTION TO PLX

The Model 650 FSVU PCI card is setup and controlled through a set of registers that are memory-mapped into the PCI address space. The interface to the PCI bus is implemented using a PLX Technology PCI 9050 ASIC. This chip contains the logic and interface registers to communicate between a peripheral interface and the PCI bus. The 9050 supports target and bus master modes. For this application, the 650 FSVU card is a target (registers are both readable and writable) and the bus mastering capabilities (DMA) are not used.

The PLX chip interfaces to the PCI bus through six programmable blocks of PCI address registers. The size and starting address of each block is established by six special PCI system configuration registers, called *Base Address Registers*. You address these registers with special *System Configuration* instructions. The PLX PCI9050 uses two of the six PCI base address registers for internal PCI bus addressable registers and provides two base address registers for peripheral addressing. The last two PCI base address registers are unused. The PLX transfers data to or from the PCI bus to an internal bus that serves as a local bus to the PCMD. The local bus provides 32 data and 32 address lines. The data on the local bus is also programmably mapped with each block of PCI address space having a corresponding programmable block address on the local bus. Two blocks of PCI memory support the PCM frame synchronizer and the mezzanine card. The first 128 byte block provides registers that setup and control the eight frame synchronizers, input frame status, and provide a serial port that controls the matrix card. A second 64 byte block controls the *Time Code Translator/Generator & PCM Simulator mezzanine cards*.

The program allocates the PCI memory addresses by using the PCI 9050 system configuration registers, a set of 16 double word registers that provide configuration information for each PCI interface slot. The configuration registers are manipulated by special instructions. The system configuration space provides 64 bytes of setup and control registers, configured as 16 32-bit words. Six of the 16 32-bit registers provide the six base address registers that are read from the PCI bus, and which the program uses to establish the size and type of registers required by the peripherals. The registers may be mapped as PCI memory or I/O. The PLX 9050 is controlled via a 256 byte block of addressable registers that are offset by Base Address Register 1. The PCMD uses two blocks of memory that can have any starting address in the 4.0GB of PCI memory address space. These blocks are accessed via Base Address Registers 3 and 4. The PLX 9050 provides tracking base address registers that re-map the selected PCI data to corresponding addresses on the local bus.

3.2 PCI SYSTEM CONFIGURATION REGISTERS

Each PCI device possesses a block of 64 configuration bytes that are reserved for configuration register implementation. The format or usage of the first 16 bytes is predefined by the PCI specification. This area is referred to as the device's configuration header region. A program will read the configuration registers to uniquely identify the hardware that resides in each PCI slot. Several registers identify the vendor and subassembly, and possibly the PROM address for software configuration. Other registers control system level programs that establish PCI bus latency and interrupt control. Still other registers provide size and base address for PCI bus memory or I/O address allocation. The system configuration registers are shown in TABLE 3-1. The PCI configuration registers are provided as a part of the PLX 9054. You can preload registers 0, 8, 2C & 3C from a serial EEPROM as a part of system initialization. Most of the configuration registers can be written from the local bus, but the PCI FSVU does not use this feature.

3.2.1 Device Configuration Register Description

Of the 64 bytes of configuration information, only the first twelve bytes and byte fifteen are mandatory. You can read the remaining 51 bytes of information as zero words if the PCI device does not use them.

3.2.2 Mandatory Device Configuration Registers

The following describes the mandatory device configuration registers that must be implemented in every PCI device.

3.2.2.1 Vendor ID Register

This 16-bit register identifies the device manufacturer. This read-only register contains a hardwired value assigned by a central authority (the PCI SIG) controlling number issuance. The value FFFFh is reserved and must be returned by the Host/PCI bridge when an attempt is made to perform a Configuration Read from a non-existent device's Vendor ID Configuration Register. The bridge must respond with a vendor ID of FFFFh. The resulting master abort signal is not considered to be an error.

3.2.2.2 Device ID Register

This 16-bit register contains a value assigned by the device manufacturer to identify the type of device.

3.2.2.3 Command Register

This register provides basic control over the ability of the device to respond to and/or perform PCI accesses. The 16-bit register has bits 0-9 defined, with bits 10-15 reserved for future use. Bits 10-15 must return zeros. The bit functions are described in TABLE 3-2. The designer only implements the bits that make sense for the device. For example, a device with I/O but no memory requires bit zero but not bit one. After reset, bits 0-2 are cleared to zero. This disables the device (except that it remains responsive to configuration accesses) until it is configured and enabled by the configuration software.

PCI DEVICE CONFIGURATION REGISTERS						
Byte Addr hex	Local Addr hex	a*	Bits ← 31 - 24 →	Bits ← 23 - 16 →	Bits ← 15 - 08 →	Bits ← 07 - 00 →
0	0	Y	Device ID		Vendor ID	
4	4	Y	Status Register		Command Register	
8	8	Y	Class Code			Revision
C	C	N	BIST	Header Type	Latency Timer	Cache Line Size
10	10	N	Base Address 0			
14	14	N	Base Address 1			
18	18	N	Base Address 2			
1C	1C	N	Base Address 3			
20	20	N	Base Address 4			
24	24	N	Base Address 5			
28	28	N	Card Bus CIS Pointer			
2C	2C	Y	Subsystem ID		Subsystem Vendor ID	
30	30	N	Expansion ROM Base Address			
34	34	N	Reserved			
38	38	N	Reserved			
3C	3C	Y	Max Latency	Min Grant	Interrupt Pin	Interrupt Line
a* = Configurable from EEPROM						

TABLE 3-1. PCI Device Configuration Registers

CONFIGURATION COMMAND REGISTER	
BIT	FUNCTION
0	I/O Access Enable. When set to one, any implemented device I/O address decoders respond to PCI I/O accesses. Zero disables the function, and default is zero.
1	Memory Access Enable. When set to one the device responds to PCI memory accesses (if the device implements any memory address decoders). Zero disables the function and the default is zero.
2	Master Enable. When set to one enables the device to act as a bus master (if it has bus master capability). Zero disables the function and the default setting is zero.
3	Special Cycle Recognition. When set to one enables the device to monitor for PCI special cycles (if designed to do so). Zero ignores special cycles and is the default.
4	Memory Write and Invalidate Enable. When set to one the device can generate the memory write and invalidate command. When set to zero the device uses memory write commands instead. This bit must be implemented by bus masters capable of generating the memory write and invalidate command. Software should not enable this bit until the device's cache line size configuration register is initialized with the system cache line size. Reset clears this bit.
5	VGA Palette Snoop Enable. When set to one, this bit instructs its VGA-compatible device to perform palette snooping. In a non-VGA graphics device, reset sets this bit to one, enabling palette snooping.
6	Parity Error Response. This bit is required for all devices. When set to one, the device can report parity errors (PERR#). When set to zero, the device ignores parity errors. This bit is cleared by reset.
7	Wait Cycle Enable. Controls whether the device does address/data stepping. Devices that never use stepping must hardwire this bit to zero. Devices that always use stepping must hardwire this bit to one. Devices that can work both ways must implement this bit as read/writable and initialize it to one after reset. The PCI PCMD does not perform address/data stepping.
8	System Error Enable. When set to one the device can drive the SERR# line. A zero disables the device SERR# output driver. State after a reset is a zero. All devices that use SERR# must implement this bit. This bit and bit six must be set when reporting address parity errors.
9	Fast Back-to-Back Enable. Optional bit for bus masters. If a bus master is capable of performing fast back-to-back transactions to different targets, this bit is used to enable or disable this function. If all targets on the PCI bus that the master resides on are fast back-to-back capable, configuration software can use this bit to enable this master's ability to perform fast back-to-back transactions to different targets. State after Reset is zero.
10-15	Reserved

TABLE 3-2. CONFIGURATION COMMAND REGISTER

3.2.2.4 Status Register

The Status Register tracks the status of PCI bus-related events. A device must implement the bits that relate to the device functionality. This register can be read from, but writes are handled differently than normal. On a write, bits can be cleared but not set. Writing a one to a bit clears the bit. This method simplifies programming. After reading the state and ascertaining which error bits are set, a program clears the bits by writing the value that was read back to the register. TABLE 3-3 below describes the status register bits.

CONFIGURATION STATUS REGISTER	
BIT	FUNCTION
4-0	Reserved. Hardwired to zero.
5	66MHz-Capable. One = device can run at 66MHz. Zero = Capable of 33MHz. Value is hardwired by designer
6	UDF Supported. One = device supports User Definable Features. Zero = device does not support UDFs. Value is hardwired in by designer
7	Fast Back-to-Back Capable. This read only bit indicates whether or not the target device supports fast back-to-back transactions with different targets. It must be hardwired to zero if the device does not support this feature and to a one if it does.
8	Data Parity Reported. This bit is only implemented by bus masters and is set only if the following conditions are met. The reporting bus master was the initiator and set PERR# itself (during a read) or detected it asserted by the target (during a write); and the Parity Error Response bit in the master's command configuration register is set to one.
9-10	Device Select (DEVSEL#) Timing. These bits are read only and define the slowest DEVSEL# timing for a target device (except configuration accesses). 00b = fast, 01b = medium, 10b = slow, and 11b is reserved.
11	Signaled Target Abort. Set by the target device whenever it terminates a transaction with a target-abort. A device that is incapable of signaling target-abort does not need to implement this bit.
12	Received Target Abort. Bus master sets whenever a transaction is terminated by a target-abort from the currently-addressed target. All bus masters must implement this bit.
13	Received Master Abort. A master sets this bit whenever a transaction (except for a special cycle) is terminated due to a master-abort. All bus masters must implement this bit.
14	Signaled System Error (SERR#). This bit is set whenever a device generates a System Error on the SERR# line. If incapable of generating SERR#, then not needed.
15	Detected Parity Error. This bit is whenever a device detects a parity error (even if parity error reporting is disabled by bit six of the command register).

TABLE 3-3. CONFIGURATION STATUS REGISTER

3.2.2.5 Revision ID Register

This 8-bit value is assigned by the manufacturer to identify the revision number of the device.

3.2.2.6 Class Code Register

This 24-bit read only register is divided into three sub-registers: *base class*, *sub-class*, & *programming interface*. The register identifies the basic function of the device (for example a mass storage controller), a more specific sub-class (for example an IDE controller), and, in some cases, a register-specific programming interface (such as the VGA register set). The upper byte defines the basic class type, the middle byte a sub-class within the basic class, and the lower byte defines the programming. This register implements Plug and Play peripheral devices in the Windows environment. This card generates a class code of FFhex which indicates that the card does not fit into any of the defined class codes.

3.2.2.7 Header Type Register

Byte fifteen provides the header type register. Bits 0-6 define the format of the device configuration header. Currently, the only header format defined other than the one pictured in TABLE 3-1 (header type 00h) is header type one (PCI-to-PCI bridge format). Future versions of the specification may define other formats. This card provides a header type 0 response. In addition, bit seven defines the device as a single (bit seven = 0) or multi-function (bit seven = 1) device. During configuration, the programmer determines if there are any other functions in this package that require configuration by testing the state of bit seven. This card is a single function device.

3.3 OPTIONAL CONFIGURATION REGISTERS

The configuration registers described in the following paragraphs may be optional or mandatory depending on the device type. They only need to be implemented if a device supports the affected function.

3.3.1 Cache Line Size Register

Mandatory for a master that uses the *memory write and invalidate* command. Also mandatory for memory targets that support cacheline wrap addressing, and for memory targets that the processor may cache from.

This read/write configuration register specifies the system cache line size. This card is configured as a memory target into memory space that the host processor does not cache from, so the configuration software must set its cache line size to zero. The target can then ignore its snoop result inputs (SDONE and SBO#) when participating in a transaction.

3.3.2 Latency Timer: Timeslice Register

Mandatory (read/writable) for masters that are capable of performing a burst consisting of more than two data phases. The latency timer defines the minimum amount of time, in PCI clock cycles, that the bus master can retain ownership of the bus. The bus master decrements its latency timer by one on the rising edge of each clock after it initiates a transaction. The bus master may continue its transaction until either: It has completed the overall burst transfer, or it has exhausted its timeslice (LT value) and it has been preempted (lost its GNT# to another PCI master), whichever comes first.

Every bus master that performs bursts of more than two data phases, some or all of the time, must implement the LT (latency timer register) as a read/write register. This can be implemented as a hardwired read only register by masters that can burst two or less data phases, but the hardwired value returned must not exceed sixteen. Be aware that a hardwired value of zero is only permitted if the master never performs more than two data phases and that the zero means the master has a null timeslice. The net effect is that the master has to yield the bus after the first data phase if it immediately lost its GNT# to another master. Target devices do not implement this bit.

3.3.3 BIST Register

Both master and target devices may implement this register. If a device contains a built-in-self-test (BIST) it must implement this register. This card does not support the BIST function and therefore returns a zero when this register is read.

3.3.4 Base Address Registers

Required by the PLX9054 and the FSVU.

Almost all devices use some memory address space, I/O address space, or both. On power-up, the system must be automatically configured so that each device I/O and memory function occupies mutually exclusive address ranges. In order to accomplish this the system must be able to detect how many memory and I/O address ranges a device requires and the size of each. In addition, the system must have the capability of programming a device's address decoders in order to assign its I/O and memory non-conflicting address ranges. The six base address configuration registers located in byte addresses 10-27hex of the *device configuration header space* permit this relocatability. Each register is 32 bits (64 bits if the memory block it describes can be located anywhere within 64-bit address space) and implements a programmable memory or I/O address decoder. TABLES 3-4 and 3-5 show the two possible formats of a base address configuration register. If bit zero is returned as a zero, the register is a *memory address decoder*. A one indicates an *I/O address decoder*. If one decoder is implemented, it must be implemented as *base address register zero*. If two decoders are implemented, they must be implemented as *base address registers zero and one*, etc. During configuration, the configuration software will stop looking for base address registers in a device's header when it detects an unimplemented base address register.

In a PC environment, I/O space is densely populated and will only become more so in the future. For this reason and because some processors are only capable of performing memory transactions, the specification strongly recommends that

the device designer provide a memory base address register to map a device register set into memory space. Optionally, an I/O base address register may also be included. This gives the configuration software the flexibility to map the device register set at least into memory space, and if an I/O base address register is also provided, into I/O space as well. The device driver associated with the device can then choose whether to communicate with its device register set through memory or I/O space.

The PLX9054 uses base address registers 0 and 1 to establish the base address of a block of 256 bytes of address that provide local configuration, runtime, and DMA registers. Base address register 0 establishes the base address for memory mapped access, and base address 1 the base address for I/O access. You can also access these registers from the local bus. They are offset by 80hex bytes when using the *PLX chip select* line.

Base address registers 2 and 3 define memory or I/O spaces that are used by the local memory on the FSVU card. The PCI address space defined by register 2 is a 128 byte block, accessed by 32 bit words, that is remapped by a local configuration register at PLX local configuration address 4hex. This register establishes the base address on the local bus, which PLX calls *local address space 0*, and is used by the 650 FSVU. The PCI address space (defined by register 3) is a 64 byte block, accessed as 16 bit words, that is remapped by the local configuration register at address F4hex. This register establishes the base address for local address space 1, used by the mezzanine card. Only two base addresses are provided to the PCI user interface. Configuration base address registers 4 and 5 are not used.

3.3.5 Memory Base Address Register

In a memory base address register (see TABLE 3-4), bits 1-2 define where the block of memory can be located (in the first MB, the first 4GB, or anywhere in 64-bit memory space). If this bit field indicates that the memory is 64-bit addressable, then this base address register occupies two double words of configuration space, not one. The first double word sets the *lower 32 bits* and the second double word sets the *upper 32 bits* of the memory base address. Bit three defines the block of memory as prefetchable or not. A block of memory space may be marked as prefetchable only if the following can be guaranteed.

There are no side effects from reads (the Read doesn't alter the contents of the location)

A Read always returns all bytes irrespective of the byte enable settings

Permitting bridges to post writes for the address range doesn't cause errors

Permitting bridges to merge writes within this range doesn't cause errors

The memory is not cached from by the host processor

As an example, the address decoder for a block of memory mapped I/O ports would hardwire the PREFETCHABLE bit to zero, while the address decoder for regular memory (non-cacheable) would hardwire it to one. The configuration software uses this bit to determine the memory address ranges that the host/bridge can safely utilize prefetching in during reads and posting during writes.

Bits 31-4 (63-4 if the memory block it describes can be located anywhere within 64-bit address space) make up the base address field and determine the size of the memory block and to set its start address.

If a memory device requires up to 4KB of memory space, the specification suggests that the memory range be set at 4KB to minimize the number of bits to be resolved by the address decoder.

Base Address	P	T		0
←31 - →4	3	2	1	0

P = Prefetchable

T = Type:

00 = Locate anywhere in 32-bit address space;

01 = Locate below 1MB;

10 = Locate anywhere in 64-bit address space;

11 = Reserved

0 = Memory space indicator

TABLE 3-4. Memory Base Address Register Format

3.3.6 I/O Base Address Register

In an I/O base address register, bit zero returns a one to indicate that I/O space is required. Bit one is reserved and should always return zero. Bits 31-2 make up the base address field and determine the size of the I/O block and to set its start address. The specification requires that a device that maps its control register set into I/O space must not consume more than 256 locations per I/O base address register.

Base Address	R	1
←31 - →2	1	0

R = Reserved

1 = I/O space is required indicator

TABLE 3-5. I/O Base Address Register Format

3.3.7 Determining Block Size and Assigning Address Range

When the configuration program probes a base address register, it is seeking to learn the following:

Is the base address register implemented?

Is it a memory or an I/O address decoder?

How much memory or address space does it require and with what alignment?

After ascertaining this information, the program uses the base address register to assign a start address to the address decoder. You can access all of this information by writing all ones to the double word and then reading it back. A return value of zero indicates that the base address register isn't implemented. Since the base address registers must be implemented sequentially, discovery of

the first unimplemented register indicates the device has no more decoders to be programmed. By scanning the returned value (assuming its non-zero) upwards from bit four of a memory base address register or from bit two of an I/O base address register, the program determines the size of the required address space. The binary-weighted value of the first one bit found indicates the required amount of space.

After determining the size and type of address space the device requires, the program specifies the start address by writing the appropriate value to the register. This highlights a PCI constraint: *all address spaces assigned are a power of two in size and are naturally aligned*. As an example, it is possible to program the memory address decoder for a 1MB block of memory to start on the one, two, or three MB boundary. It is not possible to set its start address at the 1.5, 2.3, or 3.7 MB boundary.

The smallest memory address decoder is implemented as a base address register that permits bits 31-4 to be programmed. Since the binary-weighted value of bit four is 16, 16 bytes is the smallest memory block you can design for a PCI memory decoder.

The first base address register implemented must occupy double word 04d and any additional base address registers must occupy sequential double words afterwards. When the program detects an unimplemented base address register, it knows that there are no more to detect.

A device that requires the mapping of control registers into both memory and I/O space must implement two base address registers: one describing its memory block requirement and the other describing its I/O space requirement. The device driver can only use one of these two control blocks and must leave the other unused. Because some processors are only capable of accessing memory locations (Motorola 68000 series processors) devices should always allow control registers to be mapped into memory space.

3.3.8 Expansion ROM Base Address Register

The 650 FSVU card does not contain an expansion ROM.

3.3.9 CardBus CIS Pointer

This optional register is in devices that share silicon between cardbus and PCI. This register is not used in the FSVU card.

3.3.10 Subsystem Vendor ID and Subsystem ID Registers

Optional. This register pair was added in revision 2.1 of the PCI specification. A PCI functional device may be contained on a card or be embedded within a subsystem. Two cards or subsystems that use the same PCI functional device core logic will have the same vendor and device IDs. These two optional registers uniquely identify the add-in card or subsystem that the functional device resides within. Software can distinguish the difference between cards or subsystems manufactured by different vendors but with the same PCI functional device on the card or subsystem. The subsystem vendor ID is obtained from the SIG, while the vendor supplies its own subsystem ID. A value of zero in these registers indicates there isn't a subsystem vendor and subsystem ID associated with the device.

3.3.11 Interrupt Pin Register

Required if a PCI device generates interrupt requests . The read only interrupt pin register defines which of the four PCI interrupt request pins, INTA# through INTD#, a PCI functional device is connected to. The values one through four correspond to PCI interrupt request lines INTA# through INTD#. A return value of zero indicates that the device doesn't use interrupts.

3.3.12 Interrupt Line Register

Required if a PCI device generates interrupt requests . The read/writable interrupt line register identifies which of the system interrupt request lines on the interrupt controller the device's PCI interrupt request pin (as specified in its interrupt pin register) is routed to. In a PC environment, for example, the values zero to fifteen (00 - 0Fhex) in this register correspond to IRQ0 through IRQ15. The value 255, or FFhex, indicates unknown or no connection. The values from sixteen through 254, inclusive, are reserved.

The operating system or device driver can examine a device's interrupt line register to determine which system interrupt request line the device will use to issue requests for service (and, therefore, which entry in the interrupt table to hook).

3.3.13 Min_Gnt Register: Timeslice Request

Optional for a bus master and not applicable to non-master devices . This read only register is implemented by bus master devices and not by target devices. The register indicates how long the master will like to retain PCI bus ownership (in order to attain good performance) whenever it initiates a transaction. The value hardwired into this register indicates how long a burst period the device needs, in increments of 250 nanoseconds. A value of zero indicates the device has no stringent requirement in this regard.

3.3.14 Max_Lat Register: Priority-Level Request

Optional for a bus master and not applicable to non-master devices . The PCI specification states that this read only register specifies how often the device needs access to the PCI bus, in increments of 250 nanoseconds. A value of zero indicates the device has no stringent requirement in this regard.

3.4 PLX9054 CONTROL REGISTER ACCESS

The PLX 9054 uses the base address registers 0 and 1 to establish the base address of a block of B8hex bytes of address that provides local configuration runtime registers and DMA registers. Base address register 0 establishes the base address for memory mapped access and base address 1 the base address for I/O access. These registers may also be accessed from the local bus. They are offset by 80hex bytes when using the PLX chip select line.

3.4.1 Local Configuration Registers

The local configuration registers are shown in TABLE 3-6 on the following page. For a complete description see the PLX PCI 9054 Data Book.

The primary function of the local configuration registers is to provide remapping. Registers that control the memory address allocations when transferring data from the PCI bus to the local bus. As a part of system initialization, you can preload most of the registers in the local configuration from a serial EEPROM.

3.4.2 Runtime Registers

The PLX PCI 9054 provides a block of registers that are accessible from the PCI bus at the byte address shown, which are relative to the PCI System Configuration base address 0. You can also access these registers from the system bus. The byte address shown in TABLE 3-7 is relative to the PLX Chip Select address. These registers primarily provide mailbox message data transfers between the PCI bus and the system bus. The runtime registers are shown in TABLE 3-7. See the PLX PCI 9054 Data Book for a complete description of each register function.

3.4.3 DMA Control Registers

The PLX PCI 9054 provides a block of registers that are accessible from the PCI bus at the byte address shown, which are relative to the PCI System Configuration base address 0. You can also access these registers from the system bus. The byte address shown in TABLE 3-8 is relative to the PLX Chip Select address. These registers primarily provide DMA data transfers between the PCI bus and the system bus. The DMA control registers are shown in TABLE 3-8. See the PLX PCI 9054 Data Book for a complete description of each register function.

PLX LOCAL CONFIGURATION REGISTER							
PCI Addr	Lcl Accs	Bits ← 31 - 24 →	Bits ← 23 - 16 →	Bits ← 15 - 08 →	Bits ← 07 - 00 →	b*	a*
00	80	Range for PCI-to-Local Address Space 0				Y	Y
04	84	Local Base Address (remap) for PCI-to-Local Address Space 0				Y	Y
08	88	Mode/DMA Arbitration				Y	Y
0C	8C	Reserved	Serial eeprom Write Protected Addr Boundary	Local Miscellaneous Control	Big/Little Endian Descriptor	Y	Y
10	90	Range for PCI-to-Local Expansion ROM				Y	Y
14	94	Local Base Address (remap) for PCI-to-Local Expansion ROM & BREQ0 Control				Y	Y
18	98	Local Bus Region Descriptors (Space 0 & expansion ROM) for PCI-to-Local Accesses				Y	Y
1C	9C	Range for Direct Master-to-PCI				Y	Y
20	A0	Local Base Address for Direct Master-to-PCI Memory				Y	Y
24	A4	Local Base Address for Direct Master-to-PCI I/O Configuration				Y	Y
28	A8	PCI Base Address (remap) for Direct Master-to-PCI				Y	Y
2C	AC	PCI Configuration Address Register for Direct Master-to-PCI I/O Configuration				Y	Y
F0	170	Range for PCI-to-Local Address Space 1				Y	Y
F4	174	Local Base Address (remap) for PCI-to-Local Address Space 1				Y	Y
F8	178	Local Bus Region Descriptor (space 1) for PCI-to-Local Accesses				Y	Y
FC	17C	PCI Base Dual Address Cycle (remap) for Direct Master-to-PCI (upper 32 bits)				Y	N
a* = Configurable from EEPROM b* = Configurable from PCI/Local bus							

TABLE 3-6. PLX Local Configuration Register

PLX RUNTIME REGISTERS							
PCI Addr	Lcl Accs	Bits ← 31 - 24 →	Bits ← 23 - 16 →	Bits ← 15 - 08 →	Bits ← 07 - 00 →	b*	a*
40	C0	Mailbox Register 0 (refer to note)				Y	Y
44	C4	Mailbox Register 1 (refer to note)				Y	Y
48	C8	Mailbox Register 2				Y	N
4C	CC	Mailbox Register 3				Y	N
50	D0	Mailbox Register 4				Y	N
54	D4	Mailbox Register 5				Y	N
58	D8	Mailbox Register 6				Y	N
5C	DC	Mailbox Register 7				Y	N
60	E0	PCI-to-Local Doorbell Register				Y	N
64	E4	Local-to-PCI Doorbell Register				Y	N
68	E8	Interrupt Control Status				Y	N
6C	EC	Serial eeprom Control, PCI Command Codes, User I/O Control, and Init Control				Y	N
70	F70	Device ID		Vendor ID		N	N
74	F74	Unused		Revision ID		N	N
78	F78	Mailbox Register (refer to note)				Y	N
7C	F7C	Mailbox Register (refer to note)				Y	N

a* = Configurable from EEPROM b* = Configurable from PCI/Local bus

TABLE 3-7. PLX Runtime Registers

PLX DMA CONTROL REGISTERS							
PCI Addr	Lcl Accs	Bits ← 31 - 24 →	Bits ← 23 - 16 →	Bits ← 15 - 08 →	Bits ← 07 - 00 →	b*	a*
80	100	DMA Channel 0 Mode				Y	N
84	104	DMA Channel 0 PCI Address				Y	N
88	108	DMA Channel 0 Local Address				Y	N
8C	10C	DMA Channel 0 Transfer Byte Count				Y	N
90	110	DMA Channel 0 Descriptor Pointer				Y	N
94	114	DMA Channel 1 Mode				Y	N
98	118	DMA Channel 1 PCI Address				Y	N
9C	11C	DMA Channel 1 Local Address				Y	N
A0	120	DMA Channel 1 Transfer Byte Count				Y	N
A4	124	DMA Channel 1 Descriptor Pointer				Y	N
A8	128	Reserved		DMA Ch 1 Cmd/Status	DMA Ch 0 Cmd/Status	Y	N
AC	12C	Mode/DMA Arbitration				Y	N
B0	130	DMA Threshold				Y	N
B4	134	DMA Ch 0 PCI Dual Address Cycle (upper 32 bits)				Y	N
B8	138	DMA Ch 0 PCI Dual Address Cycle (upper 32 bits)				Y	N

a* = Configurable from EEPROM b* = Configurable from PCI/Local bus

TABLE 3-8. PLX DMA Control Registers

3.5 PCI FSVU CONTROL REGISTERS

The host computer interfaces to control registers on the FSVU via the PCI bus and the PLX 9054. The registers provide status messages from the eight PCM frame synchronizers and setup the PCM formats. An optional *Time Code Generator/Translator & PCM/PAM Format Simulator* mezzanine card may be attached to and controlled from the FSVU. System Configuration Registers in the PLX 9054 chip identify the FSVU card. These registers are in a 128 byte page, with the base address specified by system configuration base address 2.

The mezzanine control registers are accessed from a 64 byte page and are read at an offset from base address register 3. The FSVU PCI registers are accessed using double word addressing, and the mezzanine is accessed with single word (16 bit) instructions. The PCM registers are shown in TABLE 3-9, and the mezzanine registers are described in *Technical Manual for the Model 470 Timecode Generator/Translator & PCM/PAM Format Simulator*, Acroamatics Document Number 6000344.

3.5.1 Signature Register

The Signature Register provides a way to set four user-selected switches that identify the PCM streams that are processed by this FSVU card. Bits 0 and 1 specify the PCM streams as follows:

00=Not Valid, 01=Streams 1-8, 10=Streams 9-15, 11=Streams 16-24.

Bits 4 through 15 read the card number 650 as BCD digits.

3.5.2 Control Register

Byte addresses four and seven provide control functions. Bits 0-7 select individual PCM streams when the corresponding bit is set to 1, and deselect that stream when set to 0. Bit 0 controls PCM stream 1, etc. Any or all streams may be selected for the RUN command (bit 31 set to 1) or the STOP command (bit 30 set to 1). A single stream must be selected for program setup commands (bit 29 set to 1), and that stream must not be in RUN during a setup sequence. Bit 24 is set to 1 when the Serial Data Register is ready to accept data. Bit 25 is set to 1 when a serial message has been read from the serial channel and is ready to be input by the host.

3.5.3 PCM Status Registers

Byte addresses 8 through F provide eight 8-bit status registers, one for each of the eight PCM streams. The LSB (first bit) of the register is set to 1 when that PCM stream is in SEARCH. The second bit is set to 1 when the input to the correlator is inverted with respect to the input data. The third bit is set to 1 when the input clock transition time exceeds 100 milliseconds. The fourth bit is set to 1 when the sync pattern correlator detects that the sync pattern is misplaced by ± 1 , ± 2 , or ± 3 bits, but is within the user-programmed sync window, i.e. a bit slip has been corrected. The fifth bit is set to 1 when the corresponding PCM stream is in RUN. The sixth bit is not used. The seventh bit is set to 1 when the sync strategy is in VERIFY mode. The MSB is set to 1 when the frame synchronizer is in LOCK mode.

PL650 PCI CONTROL REGISTER FUNCTIONS									
Addr hex	Function	Bits		Bits		Bits		Bits	
		← 31	24 →	← 23	16 →	← 15	12 11 08 →	← 07	4 3 00 →
0	Signature Reg. Sw.	0		0		6	5	0	switches
4	Control Reg. see below								
8 C	Status Reg. see below	Frame Sync 4 Frame Sync 8	Frame Sync 3 Frame Sync 7	Frame Sync 2 Frame Sync 6	Frame Sync 1 Frame Sync 5				
10	Strategy Reg. see notes below								
14	Pattern(LSB)	Pattern						→LSB	
18	Pattern(MSB)	MSB←		Pattern					
1C	Mask(LSB)	Mask						→LSB	
20	Mask(MSB)	MSB←		Mask					
24	Bits per frame	Not Used			← Frame Length →				
28-3C	Not Used								
40-56	Ser. Data Reg. see below	←		32-bit Word				→	
5A-5C	Not Used								

TABLE 3-9. 650 PCI Control Register Functions

650 CONTROL REGISTER BIT FUNCTIONS														
Addr	FUNCTION	31			24		23		16					
4	Control Register	a	b	c			d	e						
Addr	FUNCTION	15			8		7		0					
4	Control Register						8	7	6	5	4	3	2	1

a=RUN bit, b=STOP bit, c=SETUP bit, d=Serial Data Output Ready bit, e=Serial Data Input Ready bit, 8-1=Frame synchronizer Select bits

650 STATUS REGISTER BIT FUNCTIONS (typical - Frame Synchronizers 1 through 8)								
Bit	1	2	3	4	5	6	7	8
Function	Search	Inverted	Loss	Slip	Run	not used	Verify	Lock

TABLE 3-9 (continued). 650 PCI Control & Status Register Bits

650 STRATEGY REGISTER BIT FUNCTIONS							
Addr	FUNCTION	31		24		23	22← →16
10	Strategy Register	a	b	c	d		e

a=Sync Type (31-30)	b=Mode (29-28)	c=Input (27-26)	d=Window (25-24)
-----	-----	-----	-----
Normal 0 0	Fixed 0 0	Normal 0 0	1 bit Window 0 0
Alternate Comp 0 0	Burst 0 1	Invert x 1	3 bit Window 0 1
Complement 1 0	Adaptive 1 0	Auto Polarity 1 x	5 bit Window 1 0
Not Used 1 1	Not Used 1 1		7 bit Window 1 1

e=Sync Pattern Length (bits 22-16). A 7 bit field indicating the number of bits that are participating in the sync pattern, & used to generate complement sync patterns.

650 SERIAL DATA REGISTER BIT FUNCTIONS							
Addr	FUNCTION	31	24	23	16	15	8 7 0
40-54	Serial Data Register	Data			Data		

TABLE 3-9 (continued). 650 PCI Strategy & Serial Data Register Bits

3.5.4 Sync Strategy Register

Byte addresses 10 through 13 provide setup information that controls the frame synchronization strategy. The word is divided into the fields described below.

3.5.4.1 Verify to Search Count (Bits 0-3)

This field contains the number of consecutive missed sync patterns (1-15) that will be tolerated before returning to Search.

3.5.4.2 Verify to Lock Count (Bits 4-7)

This field contains the number of consecutive good sync patterns (0-15) that must be detected before advancing to Lock. A count of 0 allows the strategy to move from Search directly to Lock.

3.5.4.3 Lock to Search Count (Bits 8-11)

This field contains the number of consecutive bad sync patterns (1-15) that will be tolerated before returning to Search.

3.5.4.4 Error Tolerance Count (Bits 12-15)

This field contains the number of erroneous bits (0-15) that will be tolerated in the sync pattern and still meet the requirement for an acceptable sync pattern.

3.5.4.5 Sync Pattern Length (Bits 16-22)

This field contains the number of bits in the sync pattern. The pattern length is necessary in determining an acceptable Complement Sync pattern.

3.5.4.6 Bit Slip Window (Bits 24-25)

This field controls the bit slip window by providing a one bit window (no bit slip allowed) through a seven bit window ± 3 bits).

3.5.4.7 Input Polarity (Bits 26-27)

Bit 26 is set to 1 when you expect the input to be inverted. Bit 27 is set to 1 to allow Automatic Polarity Determination. The automatic Polarity Determination logic operates in two modes:

(1) In Search, when an acceptable complemented pattern is found, the *bits per frame* counter is set to 1 and the strategy remains in Search for one frame. If an acceptable Normal pattern is found, the strategy logic reloads the bits per frame count and transfers to Verify. If an acceptable sync pattern is not detected, and a second complement pattern is found in the sync window, the input polarity is complemented, the bits per frame count is reloaded, and the strategy advances to Verify.

(2) In Verify or Lock, when two consecutive acceptable complement sync patterns occur in the sync window, the second pattern is treated as an acceptable pattern and the input polarity is complemented. This allows the auto-polarity to work in Verify or Lock. The Verify to Search and Lock to Search counts must be set to two or more, since the first complement pattern will be seen as a missed pattern.

3.5.5 Sync Mode (bits 28-29)

The Sync Mode field controls Fixed, Burst, & Adaptive strategies as follows.

3.5.5.1 Fixed Mode

In this mode the frame sync unit will have three levels of operation: *Search*, *Verify*, & *Lock*. While in *Search* mode the frame sync unit will search for the defined frame sync pattern that meets the frame sync error allowance. Once an acceptable pattern is found the frame sync unit will load the bits per frame counter, then switch to *Verify* mode. Once in *Verify*, if a programmable number (0-15) of consecutive good patterns are found at the proper location (dependent on the slip window and bits per frame settings) the frame sync unit will switch to *Lock* mode. If a programmable number (1-15) of bad frames are encountered in *Verify*, the frame synchronizer will revert to *Search*. In *Lock* mode, at the expected End Of Frame bit (as defined by the Bits Per Frame counter and the Slip Window), the sync pattern is tested. The frame sync unit will drop back to *Search* if a programmable number (1-15) of consecutive frame sync patterns are detected with more than the programmed allowable errors.

3.5.5.2 Adaptive Mode

In this mode the frame sync unit begins in Search mode, hunting for a frame sync pattern in the data stream that meets the *Frame Sync Error Tolerance*. Once a candidate is found, the bits per frame counter is loaded and the number of detected errors in the candidate frame sync pattern is stored for comparison with the following possible frame sync pattern. The frame sync unit will remain in Search for one frame and, if a sync pattern with less errors than the current detected error count is found, the bits per frame counter will be reloaded. The new pattern error value will update the previous count and the synchronizer will search for a better pattern for one frame time. When a frame sync pattern with equal or less errors than the current detected error count occurs at the expected end of frame, the frame sync unit will advance to Verify. In Verify and Lock modes the frame synchronizer strategy works as in FIXED mode, using the Acceptable Error Tolerance established in Search.

3.5.5.3 Burst Mode

Burst mode is for synchronizing to formats that have variable length frame structures. In Search mode the frame sync unit continually searches the incoming data stream for a frame sync pattern that meets the frame sync error tolerance. Once an acceptable pattern is detected, the frame sync unit loads the bits per frame counter and advances from Search directly to Lock (Verify to Lock count set to 0). Upon detecting a frame sync pattern with errors above the allowable limit at the proper location (dependent on the Slip Window and the Bits Per Frame settings), the frame sync unit falls back to Verify mode. The Verify state processes the variable length *fill data* at the end of each frame preceding the sync pattern. In Verify mode the bits per frame counter is reinitialized, and the incoming stream is continuously searched for an acceptable sync pattern. If one is found, the bits per frame counter is loaded and the synchronizer returns to Lock. When the expected end of frame is detected (as determined by the frame length counter) and an acceptable sync pattern has not been found, the synchronizer returns to Search. If the fill data exceeds the number of bits in the frame, the Verify state is extendable by increasing the Verify to Search count.

3.5.6 Frame Sync Pattern Format Definition (Bits 30-31)

Bits 30 & 31 define three types of PCM formats that effect sync pattern recognition, as described below.

3.5.6.1 Normal

Bits 30 & 31 are both set to 0. The frame is always terminated with a sync pattern of the same polarity. If the sync pattern is found to be inverted, it should remain inverted. Normal polarity is selected when bits 26 & 27 are both 0. Inverted polarity is selected when bit 26 is set to 1, 27 to 0.

3.5.6.2 Alternate Complement

Bit 30 is set to 1, bit 31 to 0. In this format the sync pattern is complemented on alternate frames. This method of synchronization reduces the likelihood of false synchronization.

3.5.6.3 Frame Complement Sync

Bit 30 is set to 0, bit 31 to 1. In some subcommutated formats, the major frame is identified by complementing the *minor frame sync pattern* for one minor frame. When this mode is selected, an acceptable *complemented sync* pattern that is preceded and followed by normal sync patterns is treated as a good sync pattern.

3.5.6.4 Sync Pattern

The two long words at addresses 14 through 1B contain the sync pattern.

3.5.6.5 Sync Pattern Mask

The two long words at addresses 1C through 23 contain the sync pattern mask. The mask bits are set to 1 to enable the corresponding pattern bit to be compared with the PCM stream.

3.5.6.6 Bits Per Frame

The frame length in bits is contained in bits 0-15 of this word.

3.5.7 Serial Data Registers

The six long word registers starting at address 40 send serial setup messages through Stream One 6011650 to the 6011476 Input Selector Matrix. The messages are 16-bit words that select the input to each frame sync (up to 24) and the input to each Best Source Output (up to 12). Each of the 16-bit serial setup words is divided into two bytes. In the upper byte (bits 8-15), six bits select either the input source of each frame sync, or which input goes to each best source output. A byte address of 1 to 24 selects the corresponding frame sync; a byte address of 33 to 44 selects the corresponding best source output. If the upper byte selects a frame sync, then the lower byte will select Input (1), Simulator (2), or Ground (0). If the upper byte selects a best source output, the lower byte selects Ground (0), Any one of 24 Inputs (1-24), or Simulator (25). During initial setup the serial messages are written and read from address 40 of the frame sync card servicing PCM streams 1 through 8. Two 16-bit messages can be packed into the 32-bit word. When the Best Source Selector is in RUN mode, the host can use registers 40 through 54 to allow six word *block move* instructions to transfer 12 BSS messages quickly so that changing all 12 best source outputs simultaneously will not slow the BSS algorithm.

SECTION 4 THEORY OF OPERATION

4.1 BLOCK DIAGRAM

This section contains a block diagram of the Model 650 FSVU card.

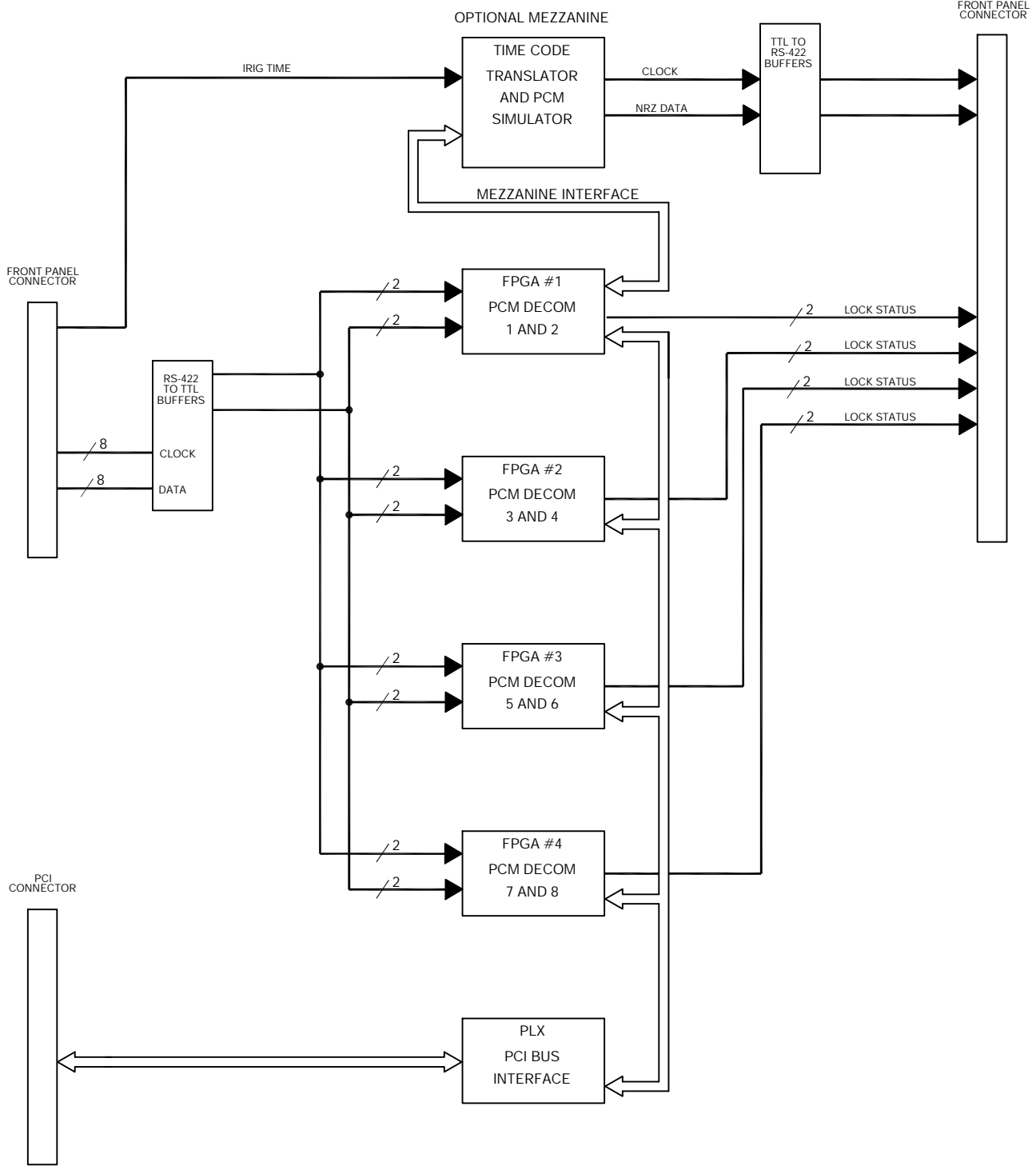
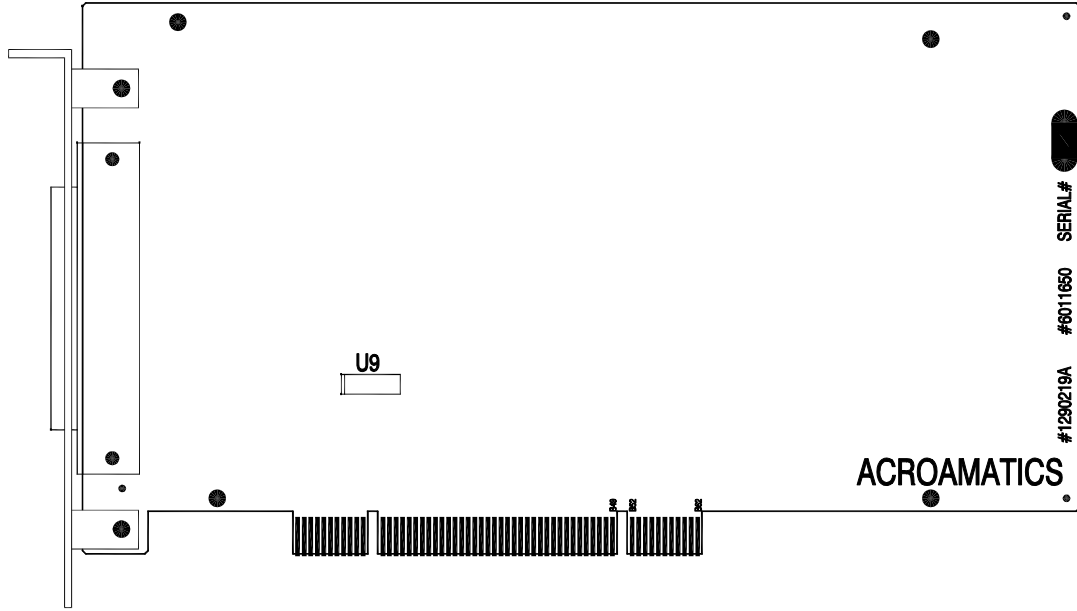


FIGURE 1 8 STREAM PCM DECOMMUTATOR BLOCK DIAGRAM

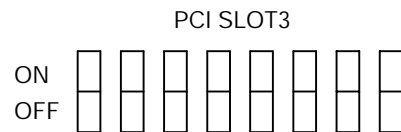
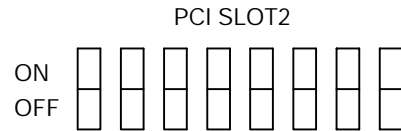
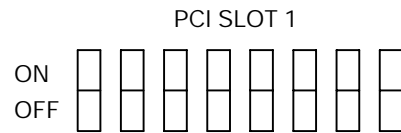
SECTION 5 ADJUSTABLE SWITCH & JUMPER SETTINGS

5.1 DESCRIPTION

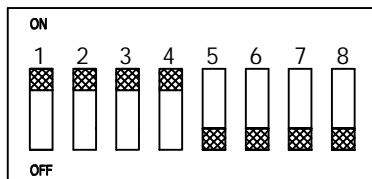
The Model 650 FSVU PCI card has one eight-position switch at U9. Positions 1-4 select the card number. Switches 3 & 4 are always ON, and switches 5-8 are always OFF. Switches 1 & 2 select the PCM card number as shown on the following page.



SWITCH U9				
1	2	3	4	
ON	ON	ON	ON	PCM CARD 1 (STREAM 1-8)
OFF	ON	ON	ON	PCM CARD 2 (STREAM 9-16)
ON	OFF	ON	ON	PCM CARD 3 (STREAM 17-24)



ON = 0
OFF = 1



SHIPPED AS _____

CARD 6011650 _____

SERIAL# / REV. _____

CUSTOMER _____ JOB# _____

CONFIGURED BY _____ DATE _____

QC CHECK BY _____ DATE _____

SWITCH SETTINGS FOR 8 STREAM PCI DECOM

SECTION 6 DRAWINGS

6.1 INTRODUCTION TO THE DRAWINGS

Section 6 contains a complete technical drawing package describing your 650 FSVU card. the drawings in this section are keyed to your card.

6.1.1 Drawing System

Acroamatics Drawing numbers are seven digit numbers which can also have a two digit dash number. The first four digits represent a drawing class, and wherever a drawing may be part of a standard drawing package, drawing numbers are issued so that all drawings which are part of the package share the same last three digits. In the following discussion "xxx" represents the number keyed to the the card part number (6011xxx). Individual parts are classified within the same drawing system, but are assigned serially without regard to other assemblies.

The PC Card Reference package includes the following drawings:

FOR CARD PART NUMBER 60116xx:

60116xx	Card Assembly Drawing
81116xx	Card List of Materials
21116xx	Card Schematic Drawing

6.1.2 Drawing Package Organization

This section of the manual contains the physical *drawings*, as opposed to the *schematic* drawings, which are found in Section 7.

The Drawings section includes the card component assembly drawing 60115nn and the card List Of Materials (LOM). LOM's include sufficient information to facilitate ordering replacement parts either from Acroamatics or from the original component manufacturer. LOMs list parts by Acroamatics Part Number in the column headed *PART NO.* The component manufacturer is identified as *VENDOR.* Parts for which ACROAMATICS is listed as vendor are proprietary components available only from Acroamatics, Inc. Integrated Circuits which are industry standard are listed as *GENERIC*, and may be obtained from any reliable vendor. Other parts for which a specific source is listed may be available from other sources. When substituting parts from vendors other than those specifically listed, be certain that the components are truly interchangeable.

The last column (Reference) of the List of Materials lists the assembly location or locations. An assembly location can contain a socket as well as the component plugged into the socket.

For example

U15
S1
74ALS244

This example shows that location U15 contains socket S1 and an IC of type 74ALS244. Resistors, capacitors, and other components are shown in a similar fashion, and are referenced using common industry abbreviations.

6.1.3 Programmed Parts

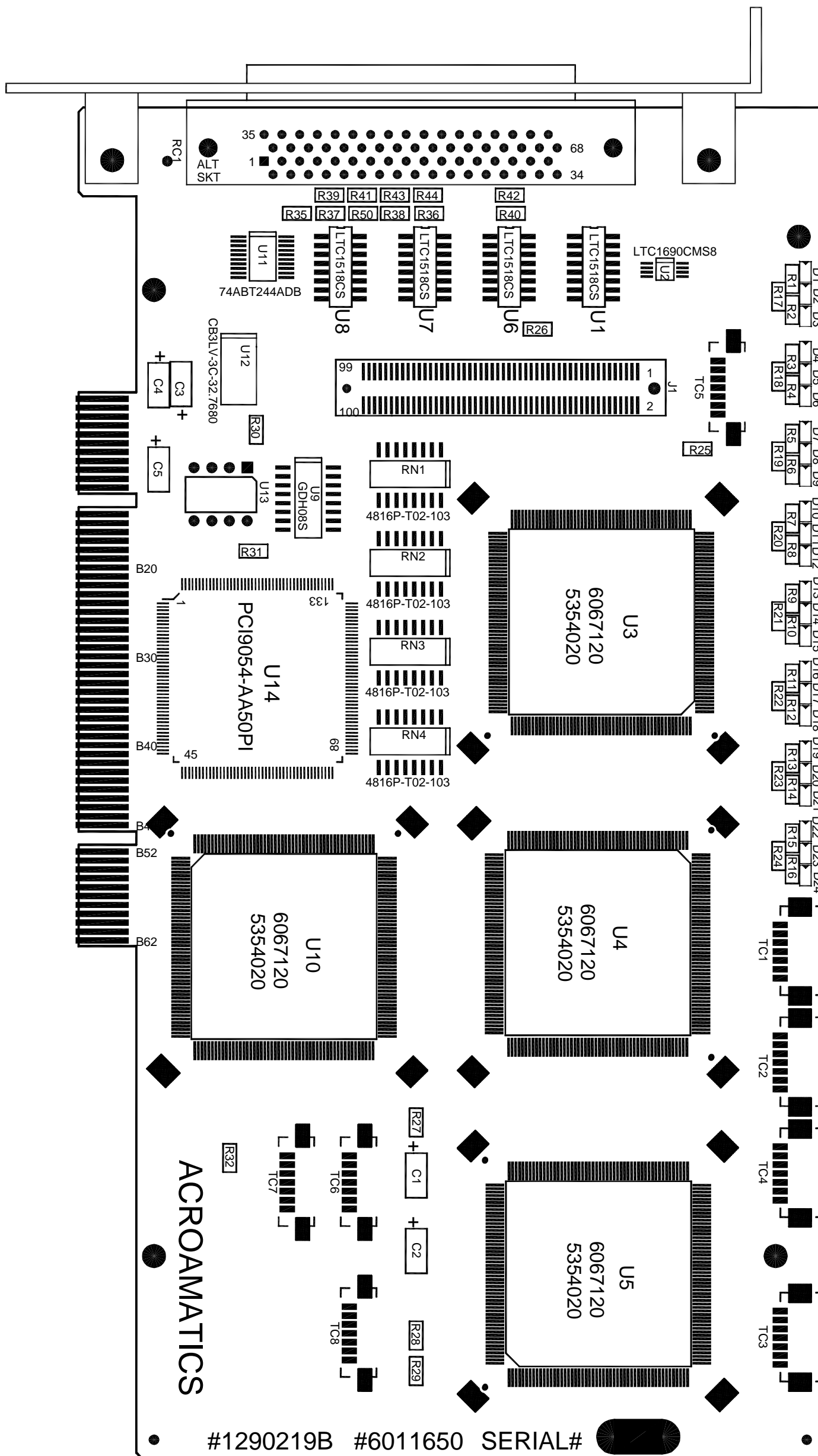
Your card includes programmed parts such as PROMs, EPROMs, EEPROMs, PALs, GALs, FPGAs, etc. If these are a permanent part of the hardware, they are documented on the List Of Materials for the PC card on which they are installed. Programmed parts are listed on the LOM twice; once as the unprogrammed part, with the Manufacturers Part Number, and also under the Acroamatics program number (606xxxx) with which they must be programmed to become the correct programmed part.

Programs for PROMs have part numbers in the series 6061xxx

Programs for EPROMs and EEPROMs have part numbers in the series 6062xxx

Programs for PALs and GALs have part numbers in the series 6064xxx

Programs for FPGAs have part numbers in the series 6067xxx

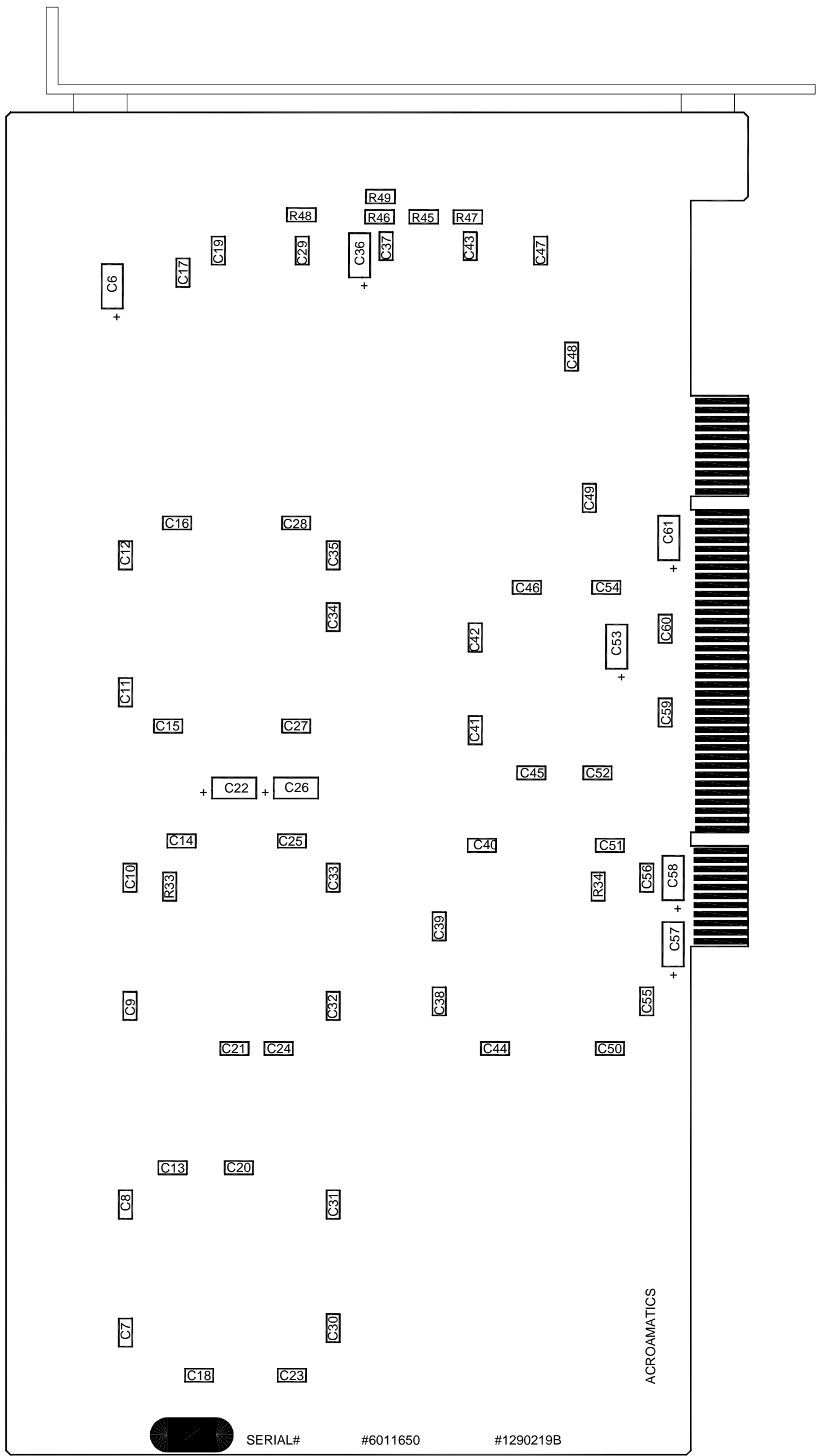


VIEW AS SEEN FROM COMPONENT SIDE

NOTE:

PROGRAM U3, U4, U5, U10, U13 PRIOR TO INSTALLATION

DR	B. GALAZIOS	1/02	ACROAMATICS <small>TELEMETRY SYSTEMS</small> <small>GOLETA, CAL. 93117</small>		
CHK					
A			ASSEMBLY, CIRCUIT CARD PCI 8 STREAM DECOM		
P					
D			SIZE	SCALE	DWG NO.
			B	NTS	6011650
	NEXT ASSY	USED ON	SHEET	2 OF 3	REV C
	APPLICATION				



VIEW AS SEEN FROM SOLDER SIDE

DR	B. GALAZIOS	1/02	ACROAMATICS <small>TELEMETRY SYSTEMS</small> <small>GOLETA, CAL. 93117</small>		
CHK					
A P P D			ASSEMBLY, CIRCUIT CARD PCI 8 STREAM DECOM		
			SIZE	SCALE	DWG NO.
			B	NTS	6011650
	NEXT ASSY	USED ON	SHEET	3 OF 3	REV C
	APPLICATION				

**LIST OF MATERIALS
PCI 8-STREAM DECOM**

8111650

PAGE 1 OF 1

ASSEMBLY PN 6011650

DRAWN BY JohnF

Jan 29 10:15

REVISION C

ENGINEERING APPROVAL _____ DATE _____

MANUFACTURING APPROVAL _____ DATE _____

NO.	PART NO	QNTY	DESCRIPTION	MANUFACTURERS PN	VENDOR	REFERENCE
1	1290219	1	PCB PCI 8 STREAM DCOM	1290219	ACROAMATICS	
2						
3	2796106	1	CONN PC 68P .05 SUB-D RTANGL	787082-7	AMP	RC1
4	2796104	4	CONN PC 7PIN 1.25 HEADER SMT	53398-0790	MOLEX	TC5-TC8
5	2796120	4	CONN PC 7P HDR RTANGLE SMT	53261-0790	MOLEX	TC1-TC4
6	2796125	1	CONN PC 100P SMT	61082-102000	BERG	J1
7						
8						
9						
10	1903069	48	CAP X7R .1uF 5% 50V SMT-805	C0805C104J5RAC	KEMET	C7-C21,C23-C25,C27-C35,C37-C52 C54-C56,C59,C60
10						
11	1922656	13	CAP TA 10uF 10% 20V SMT	T491B106K020AS	KEMET	C1-C6,C22,C26,C36,C53,C57,C58,C61
11			Acceptable substitute is:	ECS-T1DX106R	PANASONIC	
12	7680974	16	RES 121 OHM .1W 1% SMT-0805	ERJ-6ENF121	PANASONIC	R35-R50
13	7680913	24	RES 100 OHM .1W 1% SMT-0805	ERJ-6ENF100	PANASONIC	R1-R24
14	7680969	1	RES 18.2 OHM .1W 1% SMT-0805	ERJ-6ENF18.2	PANASONIC	R30
15	7680981	9	RES 10K .1W 1% SMT-0805	ERJ-6ENF10.0K	PANASONIC	R25-R29,R31-R34
16						
17	7690103	4	RES DIP 10K OHM 15R 16P SMT	4816P-T02-103	BOURNS	RN1-RN4
18						
19	5300244-91	1	IC OCTAL BUFFER NI TS SMT	SN74ABT244ADB	GENERIC	U11
20	5308518	4	IC QUAD DIFF RCVR 52Mbps SMT	LTC1518CS	LINEAR	U1,U6,U7,U8
21	5308690	1	IC DIFF DRIVER RECEIVER	LTC1690CMS8	LINEAR	U2
22	5370001	1	IC PCI BUS MASTER INTERFACE	PCI9054-AA50PI	PLX	U14
23	5354020	4	FPGA 32000 GATE 208P PQFP	A54SX32-PQ208	ACTEL	Unprogrammed FPGA
24	5352003	1	IC 4K BIT SERIAL EPROM	NM93CS66LEN	FAIRCHILD	Unprogrammed EEPROM
24			Acceptable substitute is:	93LC66B-1/P	MICROCHIP	
25	6062154	1	PCI PLX SETUP/8 STR PCM DECOM	5352003-REV.A	ACROAMATICS	U13,BSS154,Item 24
25						PROGRAM PRIOR TO INSTALLATION
26	6067120	4	DUAL FRAME SYNCHRONIZER	5354020-REV.A	ACROAMATICS	U3,U4,U5,U10, DCOM7120, Item 23
26						PROGRAM PRIOR TO INSTALLATION
27	9070015	1	DIP SWITCH 8-POS SLIDE SMT	GDH08S	AUGAT	U9
28	6350061	1	OSCIL. XTAL 32.768MHZ SMT 3V	CB3LV-3C-32.7680-T	CTS	U12
29	3570032	8	LED RED SMT 1.6 x 1.1mm	LNJ208R8ARA	Generic	D1,D4,D7,D10,D13,D16,D19,D22
30	3570033	8	LED GRN SMT 1.6 x 1.1mm	LNJ308R8LRA	Generic	D3,D6,D9,D12,D15,D18,D21,D24
31	3570034	8	LED AMBER SMT 1.6 x 1.1mm	LNJ408R8ZRA	Generic	D2,D5,D8,D11,D14,D17,D20,D23
32						
33	6730131	1	FR PNL-PCI 8 STREAM	9047-8240B	GOMPF	