
Table of Contents

1. INTRODUCTION	1
2. SYSTEM DESCRIPTION	1
3. ST-106 VERSION 1.0 RELEASE NOTES	2
3.1 UPGRADES AND NEW FEATURES	2
3.2 KNOWN DISCREPANCIES	2
4. SYSTEM REQUIREMENTS	3
5. INSTALLATION	3
5.1 ST-106 ADDRESS SETTING	3
5.1.1.1 ST-106 Connector Locations and Pin-Outs	4
5.2 SOFTWARE INSTALLATION	5
6. GETTING STARTED.....	6
6.1 SYSTEM START-UP	6
6.1.1 <i>Self Test Loop-Back Configuration</i>	8
6.1.2 <i>Beginning the Self Test</i>	9
6.2 CONFIGURING THE DECODING OPERATIONS	11
6.2.1 <i>Removing the Viterbi FEC Encoding</i>	11
6.2.2 <i>Generating a Bit Raster Display</i>	12
6.2.3 <i>Derandomizing the Data Stream</i>	13
6.2.4 <i>Differential Decoding the Data Stream</i>	15
7. ST-106 DECODER OPERATION.....	18
7.1 START-UP	18
7.2 MAIN MENU.....	18
7.2.1 <i>System Configuration Menu</i>	19
7.2.1.1 Setting Board Address.....	20
7.2.2 <i>Decoder Setup Parameters</i>	21
7.2.2.1 ST-106 Board Number (Selecting a Specific ST-106)	21
7.2.2.2 Selecting Input Signal Source	22
7.2.2.3 Selecting Input Rate and Modulation Type	22
7.2.2.4 Output Clock Select	22
7.2.2.5 Input Gating Select	23
7.2.2.6 Decoder Self-Test	23
7.2.2.7 Lock Status	24
7.2.3 <i>Configuring the Decoder</i>	24
7.2.3.1 Viterbi Decoder Menu	25
7.2.3.1.1 Constraint Length 7 (K=7) Decoding	26
7.2.3.1.2 Constraint Length 6 (K=6) Decoding	28
7.2.3.2 Differential Decoder Menu	28
7.2.3.3 Derandomizer Menu	29
7.2.3.3.1 Feedthru Derandomizer Menu	29
7.2.3.3.2 Additive Derandomizer Menu	31
7.2.3.4 Frame Synchronizer Menu	34
7.2.3.4.1 Idle Based Frame Sync	34
7.2.3.4.2 Fixed Length Frame Sync	36
7.2.3.4.3 Variable Length Frame Sync	37
7.2.3.5 Take & Skip Function	38
7.2.4 <i>Other Decoder Controls and Displays</i>	39

7.2.4.1 SEND Button.....	39
7.2.4.2 Cancel Button	39
7.2.4.3 Saving and Recalling Decoder Configurations	39
7.2.4.4 Saving Data to Disk	40
7.2.4.4.1 ASCII Data Storage Format	42
7.2.4.4.2 Hex Data Storage Format	43
7.2.4.4.3 Binary Data Storage Format	44
7.2.4.4.4 Character Mapped Data Storage Format	44
7.2.4.5 Displaying a Data Bit Raster	46
8. THEORY OF OPERATION	48
9. SPECIFICATIONS	50

List of Figures

FIGURE 2-1 ST-106 DECODER / DERANDOMIZER	1
FIGURE 5-1 ST-106 CONNECTOR AND ADDRESS JUMPER LOCATIONS.....	3
FIGURE 5-2 ST-106 MOUNTING BRACKET CONNECTIONS	4
FIGURE 5-3 SIGTEK GROUP AND ST-106 ICON	6
FIGURE 6-1 ST-106 MAIN CONTROL MENU.....	7
FIGURE 6-2 SYSTEM CONFIGURATION MENU	8
FIGURE 6-3 ST-106 SELF TEST CONFIGURATION	9
FIGURE 6-4 DECODER SELF TEST WINDOW.....	10
FIGURE 6-5 TEST DATA AND CLOCK OUTPUT.....	10
FIGURE 6-6 VITERBI DECODING SETUP MENU.....	12
FIGURE 6-7 BIT RASTER CONTROL WINDOW	13
FIGURE 6-8 DERANDOMIZER CONTROL WINDOW.....	14
FIGURE 6-9 DERANDOMIZER OUTPUT BIT RASTER	15
FIGURE 6-10 DIFFERENTIAL DECODER SETUP MENU	16
FIGURE 6-11 DIFFERENTIAL DECODER OUTPUT BIT RASTER (R8 LRS)	17
FIGURE 7-1 MAIN DECODER CONTROL MENU.....	18
FIGURE 7-2 SYSTEM CONFIGURATION MENU	19
FIGURE 7-3 BOARD CONFIGURATION MENU.....	21
FIGURE 7-4 DECODER SELF-TEST MENU.....	24
FIGURE 7-5 VITERBI DECODER MENU FOR CONSTRAINT LENGTH 7 (K=7).....	26
FIGURE 7-6 Q1650 ENCODER POLYNOMIALS	27
FIGURE 7-7 Q1650 DECODER INTERNAL PROCESSING FUNCTIONS	27
FIGURE 7-8 Q1650 V.35 DERANDOMIZER IMPLEMENTATION	27
FIGURE 7-9 VITERBI DECODER MENU FOR CONSTRAINT LENGTH 6 (K=6).....	28
FIGURE 7-10 DIFFERENTIAL DECODER SETUP MENU	29
FIGURE 7-11 DIFFERENTIAL DECODER IMPLEMENTATION.....	29
FIGURE 7-12 FEEDTHRU DERANDOMIZER SETUP MENU	30
FIGURE 7-13 PROGRAMMABLE FEEDTHRU DERANDOMIZER	30
FIGURE 7-14 FEEDTHRU DERANDOMIZER EXAMPLE (TAPS 0,17,20).....	31
FIGURE 7-15 ADDITIVE DERANDOMIZER MENU	32
FIGURE 7-16 ADDITIVE DERANDOMIZER IMPLEMENTATION.....	32
FIGURE 7-17 ADDITIVE DERANDOMIZER SIGNAL MODEL	33
FIGURE 7-18 UNIQUE WORD EDIT MENU.....	34
FIGURE 7-19 IDLE BASED FRAME SYNCHRONIZER SETUP MENU.....	35
FIGURE 7-20 IDLE BASED FRAME EXAMPLE	35
FIGURE 7-21 FIXED LENGTH FRAME SYNCHRONIZER SETUP MENU	36
FIGURE 7-22 FIXED LENGTH FRAME EXAMPLE.....	36
FIGURE 7-23 VARIABLE LENGTH FRAME SYNCHRONIZER SETUP MENU.....	37
FIGURE 7-24 VARIABLE LENGTH FRAME EXAMPLE	37
FIGURE 7-25 TAKE & SKIP FUNCTION MENU	38
FIGURE 7-26 "TAKE AND SKIP" EXAMPLE.....	38
FIGURE 7-27 DECODER CONFIGURATION FILE SAVE.....	40
FIGURE 7-28 OUTPUT DATA FILE CONTROL MENU.....	41
FIGURE 7-29 FRAME SYNC SETUP FOR DATA STORAGE EXAMPLE	42
FIGURE 7-30 ASCII FORMATTED COLLECT ("MSB TO LSB").....	43
FIGURE 7-31 HEX FORMATTED COLLECT ("MSB TO LSB").....	43
FIGURE 7-32 CHARACTER MAP FILE EXAMPLE.....	45
FIGURE 7-33 BIT RASTER CONTROL WINDOW.....	46
FIGURE 7-34 BIT RASTER DISPLAY.....	47
FIGURE 8-1 ST-106 BLOCK DIAGRAM	48

List of Tables

TABLE 5-1 ST-106 J6 CONNECTOR PIN ASSIGNMENTS.....5
TABLE 5-2 ST-106 J8 RS-422 PIN ASSIGNMENTS.....5
TABLE 5-3 ST-106 J5 RS-422 PIN ASSIGNMENTS.....5
TABLE 9-1 ST-106 DECODER SPECIFICATIONS.....50

1. Introduction

This manual describes the installation and operation of the SIGTEK ST-106 Decoder / Derandomizer. Section 2 provides a brief description of the ST-106 and its applications. Section 3 consists of release notes for the current software release. Section 4 details the system requirements for proper ST-106 operation. Section 5 discusses board installation and configuration. Section 6 provides a “getting started” overview of board operation based on the decoder loop-back test. Section 7 provides complete details on board operation. Section 7 briefly discusses the ST-106 theory of operation. Board specifications are supplied in Section 8.

2. System Description

The ST-106 Decoder / Derandomizer is a versatile, single card decoder designed to perform a variety of FEC decoding and data synchronization functions. Applications of the ST-106 include satellite earth station systems and link performance analysis using a variety of error correction schemes. The ST-106 is designed to be used with the companion ST-105 Wideband Tuner-Demodulator to create a turn-key receiving system. The ST-106 is designed for use in an IBM style PC-AT or compatible.

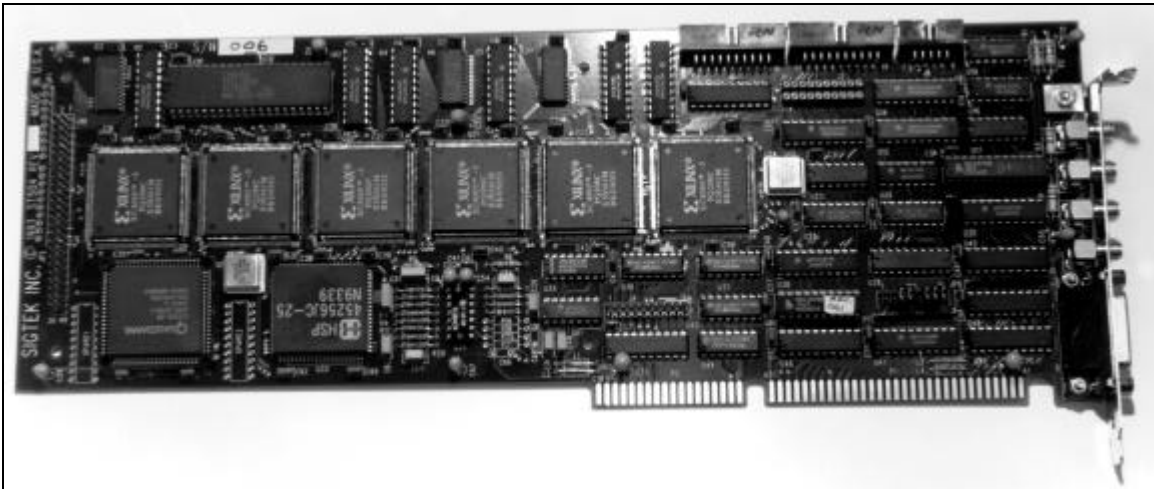


Figure 2-1 ST-106 Decoder / Derandomizer

Demodulated receive data is input in either hard or soft decision formats. Decoding of the constraint length 6 or 7 convolutional codes is performed by two ASIC chips. Other bit stream processing functions such as frame synchronization, derandomization, decimation and differential decoding are performed by six FPGAs interconnected via a proprietary TDM bus. The system software allows these functions to be chained together in any order.

The six field programmable gate arrays (FPGAs) allow decoding and bit processing functions to be reprogrammed to suite specialized end-user needs.

The ST-106 is available in two configurations. The Standard configuration uses six XILINX 4005 FPGA's to perform bit processing functions. The ST-106A uses six XILINX 4010 FPGA's for special applications requiring more complex circuitry.

The ST-106 control software is an easy to use Windows application. Decoding functions on the ST-106 are configured by selecting from a series of pull-down menus. Operation of the ST-106 software is described in Section 7.

3. ST-106 VERSION 1.0 RELEASE NOTES

This section describes upgrades and new features of ST-106 software version 1.0 along with a list of problems and discrepancies which are known to exist on the current version.

3.1 Upgrades and New Features

The following upgrades and new features are included in ST-106 software version 1.0

- Bit raster display of data streams (see Section 7.2.4.5)
- Input burst activity gating when connected to ST-105 Demodulator (see Section 7.2.2.5)
- Binary data storage for more efficient data storage (see Section 7.2.4.4.3)
- Character mapped text storage for non-standard character mappings (see Section 7.2.4.4.4)
- Self-test data stream generation which allows loop-back testing (see Section 7.2.2.6)
- Support for the ST-106A board. (see Section 7.2.1.1).

3.2 Known Discrepancies

This section describes problems and discrepancies which are known to exist on the current version of the ST-106 software. These problems will be remedied and a new software release issued. If any additional problems are found with the ST-106 are discovered, please notify SIGTEK at the address on the front of this manual.

- Rate 1/3 K=7 decoding not operational.
- K=7 QPSK operation limited to 1/2 rate only and only via the ST-105 interface.

4. System Requirements

The following minimum system requirements are necessary for installation and operation of the ST-106 software and hardware:

Computer System:	IBM PC-AT or compatible (486DX-33 MHz processor recommended)
Operating System:	DOS 5.0 or later with Windows 3.1 required.
Display:	VGA or SVGA monitor required.
Disk Space:	5 Mbytes minimum free.
Memory:	8 Mbytes minimum recommended.
Floppy Drive:	3.5" 1.44 Mbyte floppy drive required for software installation.

5. Installation

5.1 ST-106 Address Setting

Figure 5–1 shows the locations the ST-106's connectors and address jumpers. The ST-106 I/O address is set at the factory and will require change this address conflicts with other devices in the computer. Installing a jumper on JP5 sets the corresponding address line to logic '0'. In the illustration, the address jumpers are configured for board address 320 hex. The ST-106 occupies four consecutive I/O addresses starting with the configured board address.

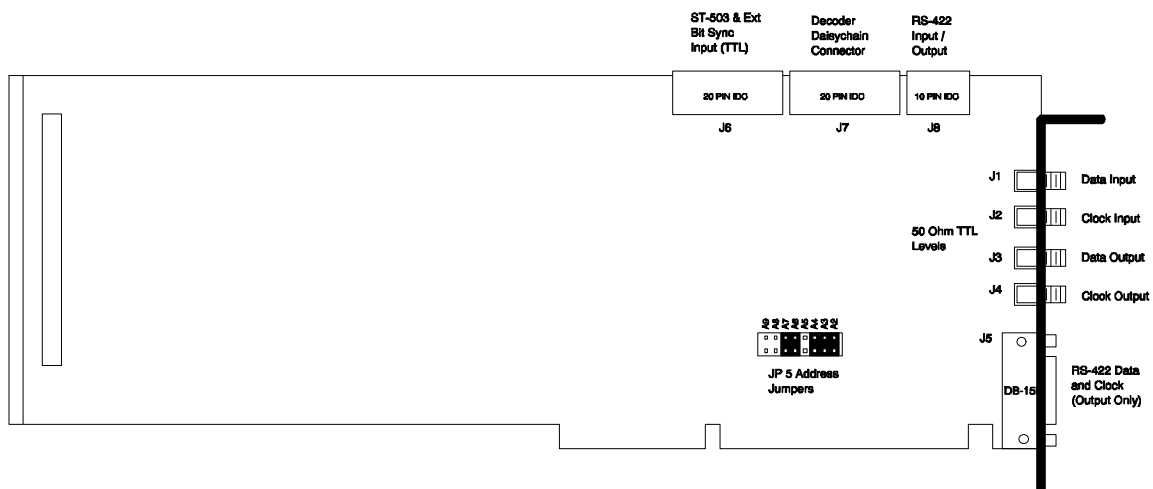


Figure 5–1 ST-106 Connector and Address Jumper Locations

5.1.1.1 ST-106 Connector Locations and Pin-Outs

Locations the ST-106's connectors are shown in Figure 5–1. Figure 5–2 shows location of connectors on the mounting bracket of the board. SMA connectors J1 and J2 allow hard-decision (single bit) data and clock to be input to the ST-106 at TTL levels. Input selection via the system software is discussed in Section 7.2.2.2. These inputs are terminated in 50 ohms. SMA connectors J3 and J4 output decoded data and clock at TTL levels and can drive 50 ohm coaxial cables. Decoded data and clock is also available in RS-422 format at connector J5. Connector J6 is designed to interface directly to the ST-105 Demod/Bit Sync via a ribbon cable to allow soft-decision (multiple-bit) data input to the ST-106 decoder. Connector J8 is an auxiliary RS-422 data and clock interface. Table 5–1 gives pin assignments for connector J6; Table 5–2 for J8; Table 5–3 for J5.

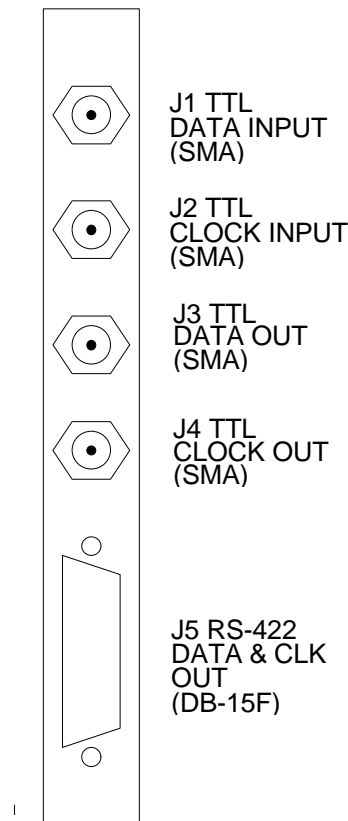


Figure 5–2 ST-106 Mounting Bracket Connections

Table 5–1 ST-106 J6 Connector Pin Assignments

Pin Number	Description
2	I Soft Decision Data In 0 (LSB)
4	I Soft Decision Data In 1(MSB)
6	I Soft Decision Data In 2 (Sign Bit)
8	Q Soft Decision Data In 0 (LSB)
10	Q Soft Decision Data In 1(MSB)
12	Clock In
14	Q Soft Decision Data In 2 (Sign Bit)
16	Burst Activity Input
17 to 20	GND
	Other Pins Unused

Note: Soft decision data is in Sign Magnitude format. For hard decision (single bit) data, use only the sign bits, (pin 6 and pin 14).

Table 5–2 ST-106 J8 RS-422 Pin Assignments

Pin Number	Description
1	Data in +
2	Data in -
3	Clock in +
4	Clock in -
5	Data out +
6	Data out -
7	Clock out +
8	Clock out -

Table 5–3 ST-106 J5 RS-422 Pin Assignments

Pin Number	Description
5	Data out +
6	Data out -
7	Clock out +
8	Clock out -
1	Gnd

5.2 Software Installation

To install the ST-106 software, insert the 3.5" installation disk into drive B: and type B:install from the DOS prompt. The installation software will automatically install the software in the computer's C: drive. At the conclusion of software installation, the SIGTEK Group with the ST-106 icon will appear as shown in Figure 5–3.

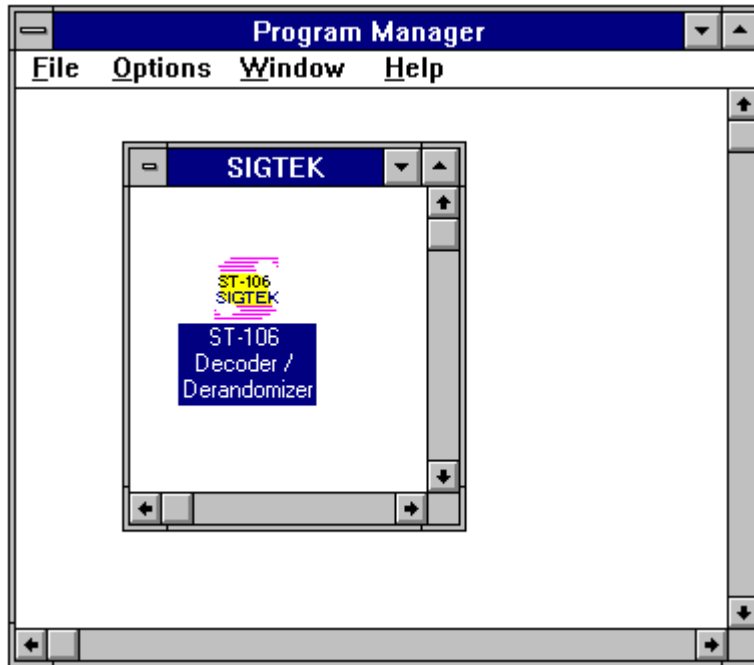


Figure 5–3 SIGTEK Group and ST-106 Icon

6. GETTING STARTED

This section is designed to quickly demonstrate the main features of the ST-106 system and allow the user to become familiar with its operator interface. The section is based on the ST-106 loop-back self-test feature. A detailed discussion of each of the operations performed in this section can be found in the appropriate parts of Section 7.

6.1 System Start-Up

To start the ST-106 control software from Windows, open the SIGTEK program group and double click on the ST-106 decoder icon. The ST-106 main control menu shown in Figure 6–1 will appear.

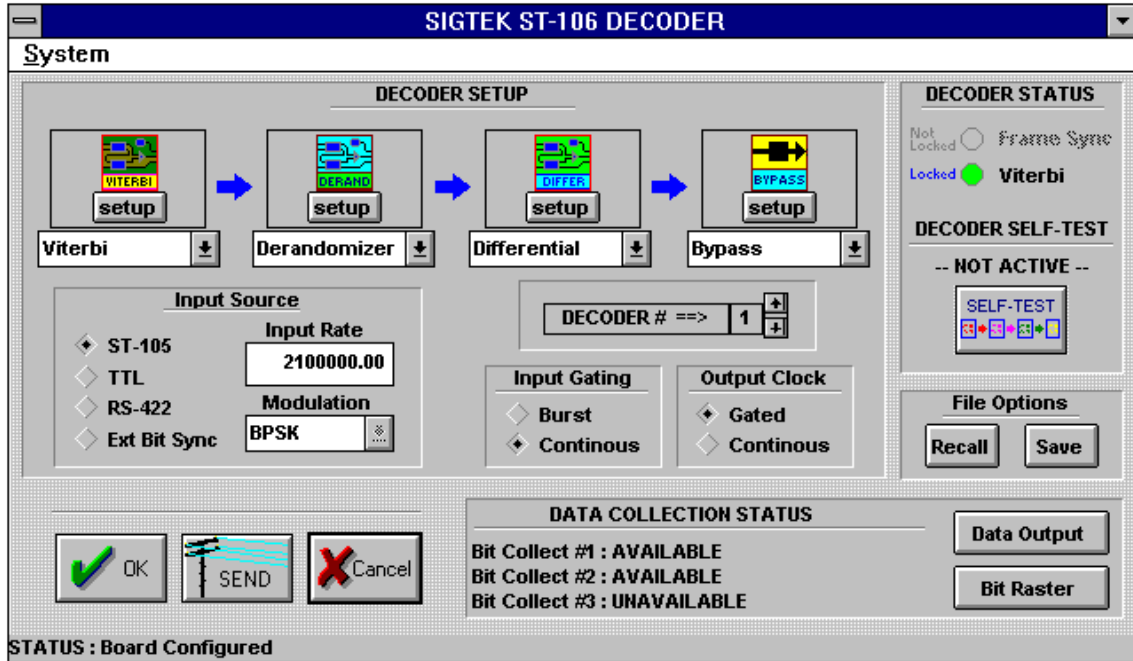


Figure 6–1 ST-106 Main Control Menu

Select System Configuration under the System pulldown menu and verify that the System Configuration Menu appears as shown in Figure 6–2. The System Configuration Menu lists which boards are installed and operational in the system. Verify that one board is installed at address 0x320 (factory default I/O address). If not, consult section 7.2.1.1 for properly setting the board address in software.

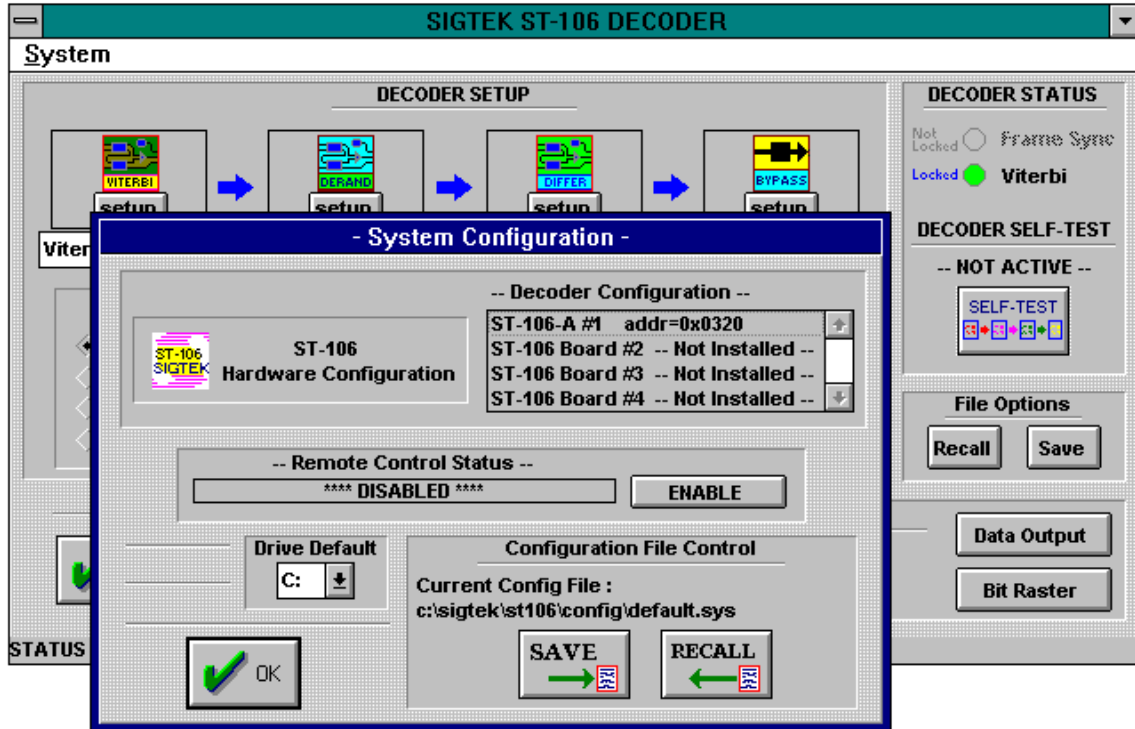


Figure 6–2 System Configuration Menu

6.1.1 Self Test Loop-Back Configuration

In the self test mode, the ST-106 Decoder/Derandomizer board is configured to generate an encoded and randomized test data stream typical of those seen on VSAT and SATCOM networks. When in self-test mode, the test data stream is output on SMA connector J3 and clock on J4. The loop-back test requires that the J3 data output be connected to J1 data input and J4 clock output to J2 clock input as shown in Figure 6–3. Two SMA to SMA cables will be required to perform the test signal loop-back.

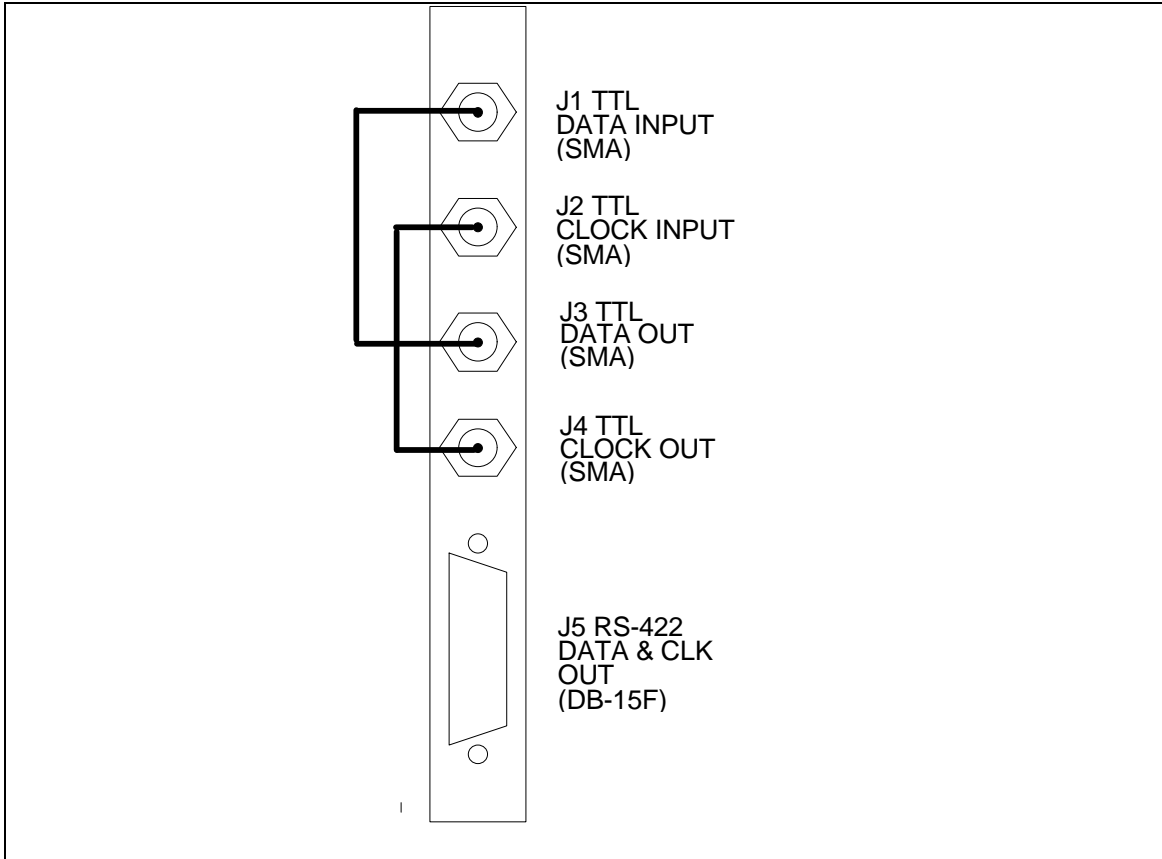


Figure 6-3 ST-106 Self Test Configuration

6.1.2 Beginning the Self Test

The first step in beginning the selftest is configuring the ST-106 decoder to generate the test data stream. To generate the test data stream from decoder #2, first press the Self Test button which results in the Self Test Window shown in Figure 6-4. Select the K=7 convolutional encoder then press the Start Test button to begin generation of the selftest data stream.

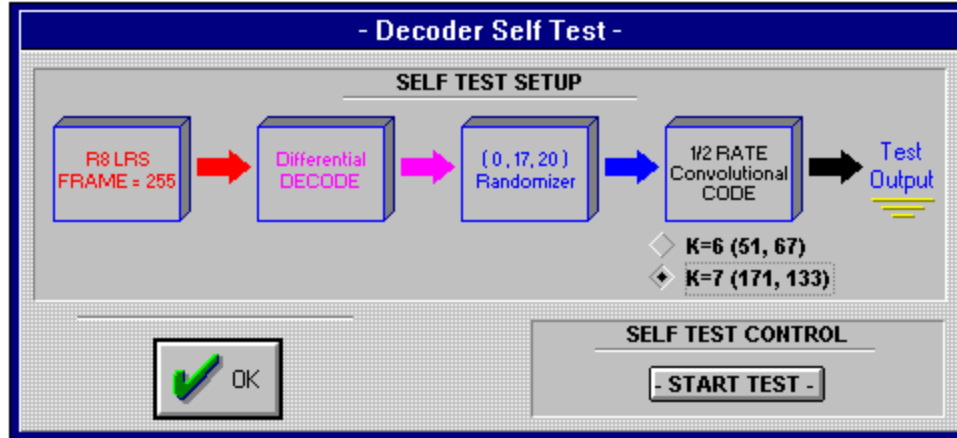


Figure 6-4 Decoder Self Test Window

The test data consists of an R8 LRS clocked at approximately a 100 kbits/sec rate. This LRS will repeat every 255 bits. The LRS is then randomized with a V.35 feedthrough randomizer, differentially encoded, then FEC (forward error correction) encoded with a rate 1/2 K=7 Viterbi (convolutional) code. The rate 1/2 Viterbi code results in approximately a 200 kbits/sec transmitted data stream. The data stream output from J3 can be observed using an oscilloscope as shown in Figure 6-5. The data stream will appear as a random stream of bits due to the encoding and randomization.

Figure 6-5 Test Data and Clock Output

6.2 Configuring the Decoding Operations

The Decoder Control Window is designed to visually represent the signal flow of operations which must be performed on the input bit stream in order to recover the underlying data. Each of the four boxes across the center of the menu can be configured to perform a separate function such as Viterbi decoding, derandomization, differential decoding, etc. Selection of the function is performed using the pull-down menus associated with each box as shown in Figure 6–1. The following sections will walk through each step involved in decoding the test signal and recovering the underlying data.

6.2.1 Removing the Viterbi FEC Encoding

The goal of the decoding process is to “undo” each of the encoding functions shown but in reverse order. Since the last encoding function is application of a 1/2 rate Viterbi FEC, this will be the first decoding function performed. (Note: The term “Viterbi encoding” although commonly used is somewhat incorrect. It is more accurate to use the term “convolutional encoding” to describe this form of FEC encoding. “Viterbi” is the name given to one type of decoding algorithm for convolutional codes.)

To configure the decoder to remove the Viterbi FEC, select Viterbi from the pull-down menu in the first (leftmost) block in the window then hit the Setup button in the box. This will display the Viterbi Decoding Setup Menu shown in Figure 6–6.

When the ST-106 is configured to generate the test data stream, the normal decoder output is disabled and the test data stream output is substituted. Also note that as the decoder configurations are changed, the test mode is automatically disabled each time a new decoder parameters are downloaded. For this reason, the test mode must be re-enabled each time the decoder configuration is changed during the test.

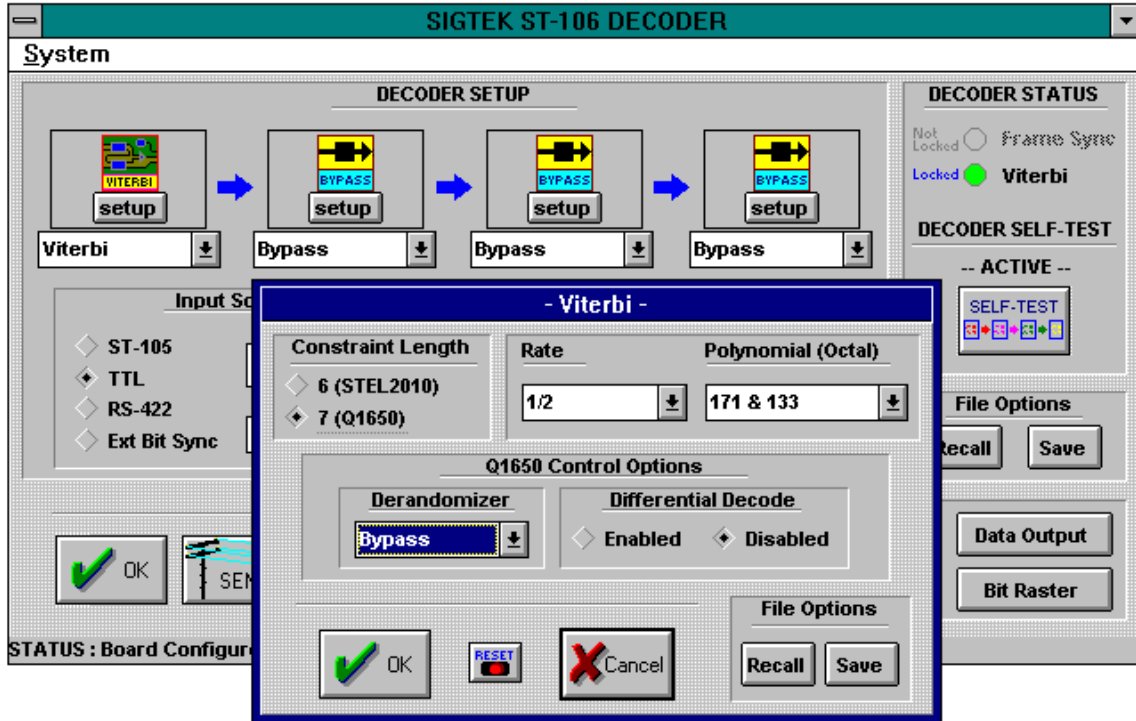


Figure 6–6 Viterbi Decoding Setup Menu

The FEC used is a 1/2 rate constraint length 7 ($K=7$) convolutional code. Set each of the decoding parameters as shown in Figure 4-20 then press the OK button to exit the Viterbi menu. Make sure that at this point, the remaining 3 blocks are set to Bypass so they are not performing any function. Press the Send button on the main decoder window to download the parameter changes to the hardware.

An initial indication that the decoder is properly decoding the Viterbi FEC is that the Viterbi Lock Status indicator in the upper left corner of the main decoder menu will be active. If the Viterbi lock is not active, it may be necessary to re-enter the Viterbi setup menu and press the RESET button to reset the Q1650 decoder chip.

6.2.2 Generating a Bit Raster Display.

The bit raster display provides a graphical snapshot of data being processed by the decoder. To generate a bit raster display, press the Bit Raster button under Data Collection Status in the upper right portion of the decoder window. This generates the Bit Raster Control window shown in Figure 4-21. The bit raster displays a snapshot of collected bits with marks (ones) represented as “x” and spaces (zeros) as “.”. The bits are displayed or “rastered” on a particular line width in order to reveal structure in the data due to repetitive patterns. To display the output of the Viterbi decoding process, set the Line Width to 255 and press the PLOT button. After a short delay, a bit raster display

similar to that shown in Figure 6-7 will be displayed. Note that at this point in the processing, the data still appears random.

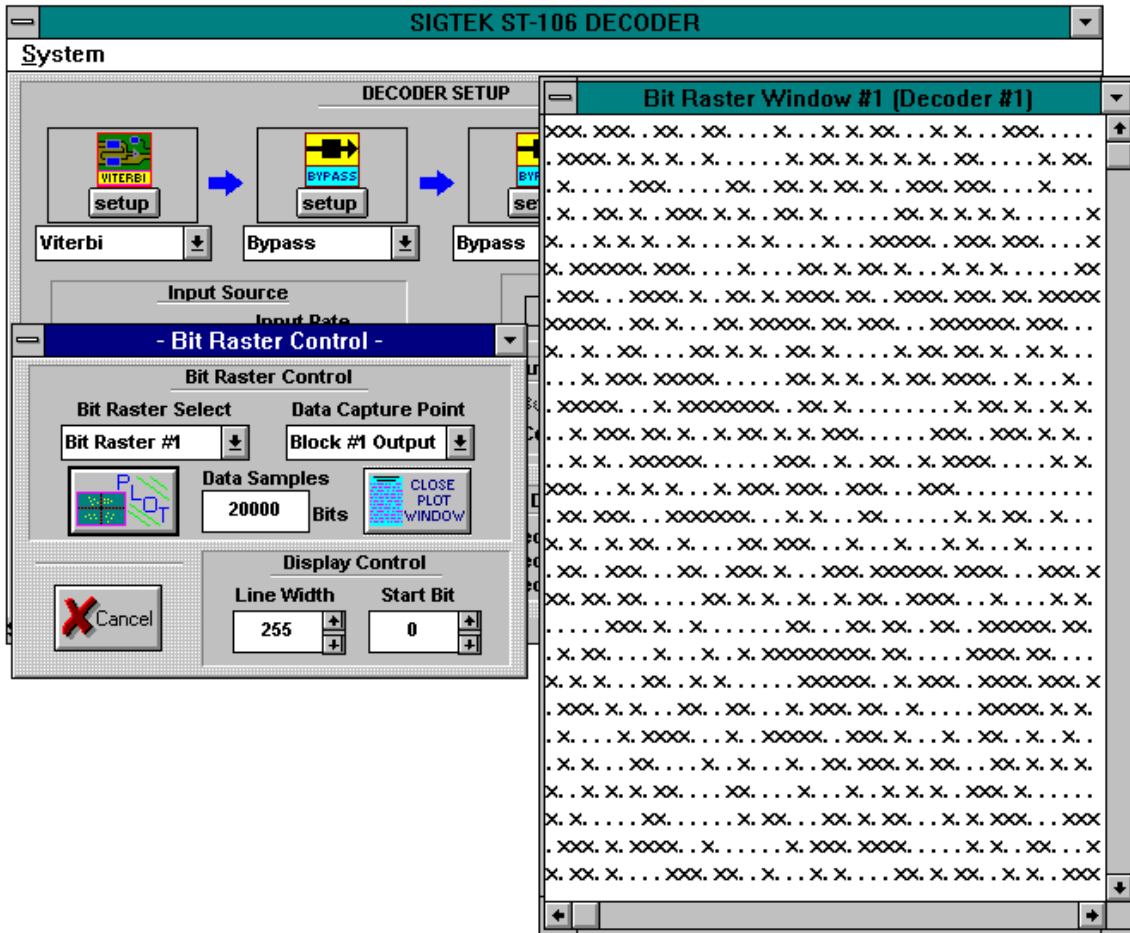


Figure 6-7 Bit Raster Control window

6.2.3 Derandomizing the Data Stream.

Removing the V.35 randomizer is the next step in the decoding chain. To configure the decoder to remove V.35 randomizer, select Derandomizer from the pull-down menu in the second block (from the left) then hit the Setup button in the box. This will display the Derandomizer Menu shown in Figure 6-8.

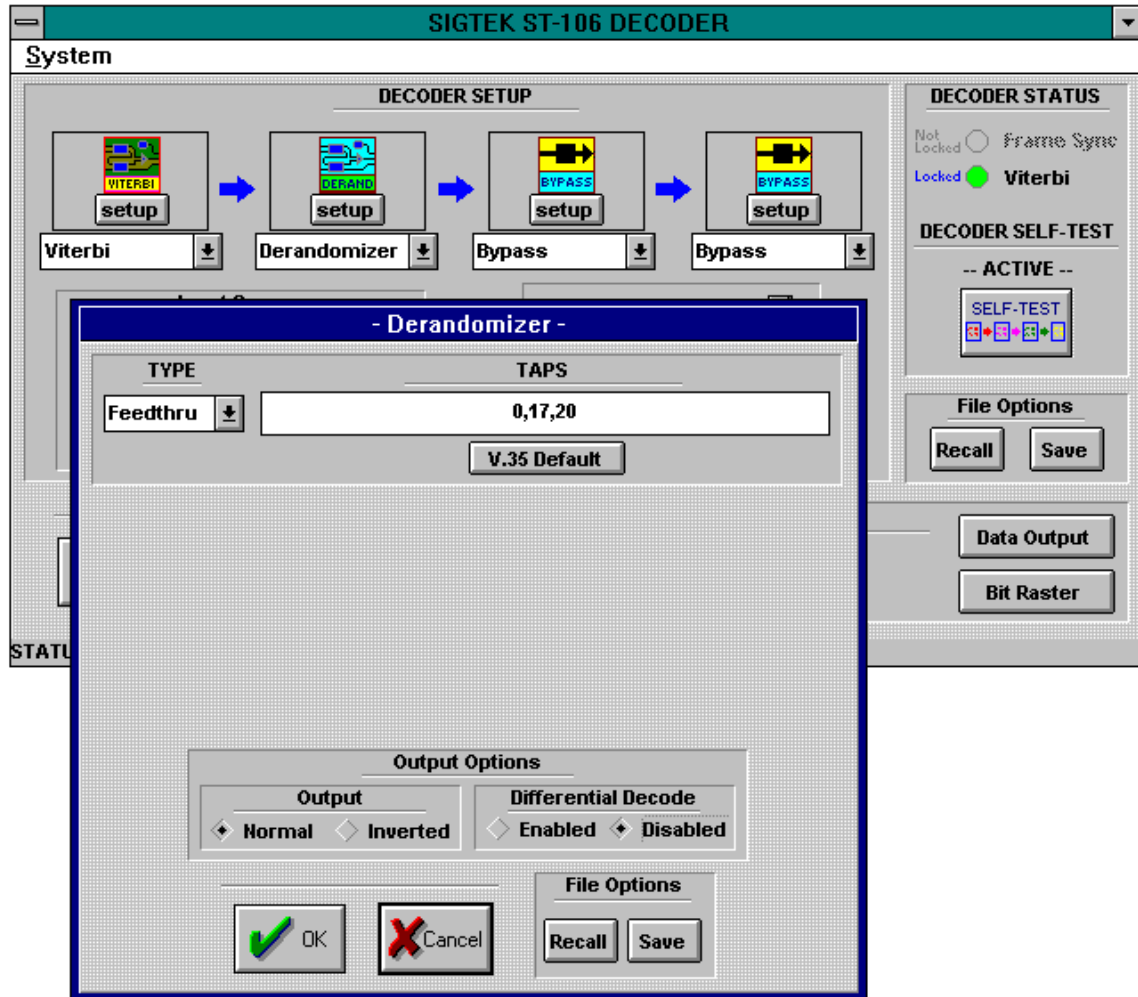


Figure 6–8 Derandomizer Control window

Set each of the derandomizer parameters as shown in Figure 6–8 then press the OK button to exit the derandomizer menu. Make sure that at this point, the first block is still set to process the Viterbi FEC and that the last two blocks are set to Bypass. Press the Send button on the main decoder window to download the parameter changes to the hardware.

To verify that the derandomization is correct, generate a bit raster of the derandomizer output by setting the Data Capture point to Block #2 Output as discussed in the previous section. When rastered on a line width of 255, repetitive structure will begin to appear in the derandomized data stream as shown in Figure 6–9.

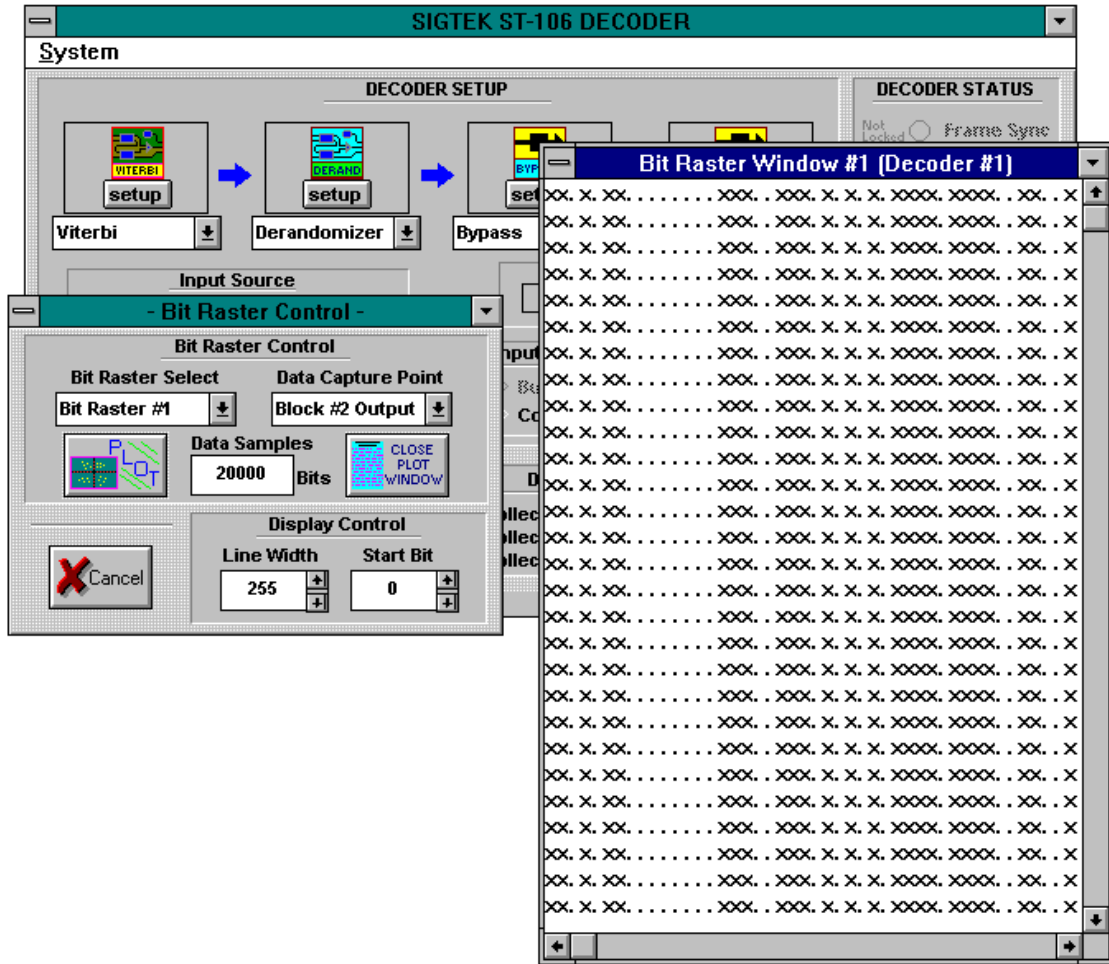


Figure 6–9 Derandomizer Output Bit Raster

6.2.4 Differential Decoding the Data Stream.

The final step in the decoding chain is to remove the differential encoding of the data stream to reveal the R8 LRS data. To configure the decoder to remove the differential encoding, select Differential from the pull-down menu in the third block (from the left) then hit the Setup button in the box. This will display the Differential Decoder Menu shown in Figure 6–10. Press the OK button to exit the differential decoder menu then press the Send button on the main decoder window to download the parameter changes to the hardware.

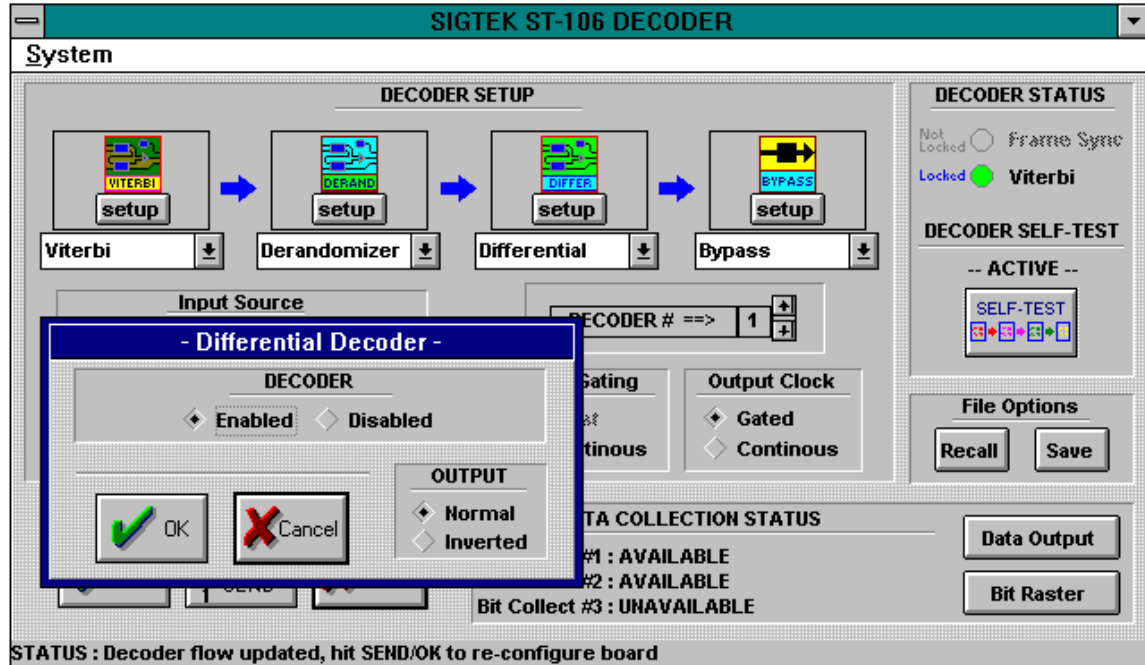


Figure 6–10 Differential Decoder Setup Menu

To verify that the differential decoding is correct, generate a bit raster of the differential decoding output by setting the Data Capture point to Block #3 Output as discussed in the previous section. When rastered on a line width of 255, the R8 LRS structure will appear as shown in Figure 6–11. The vertical and horizontal scroll bars can be used to display other portions of the bit raster.

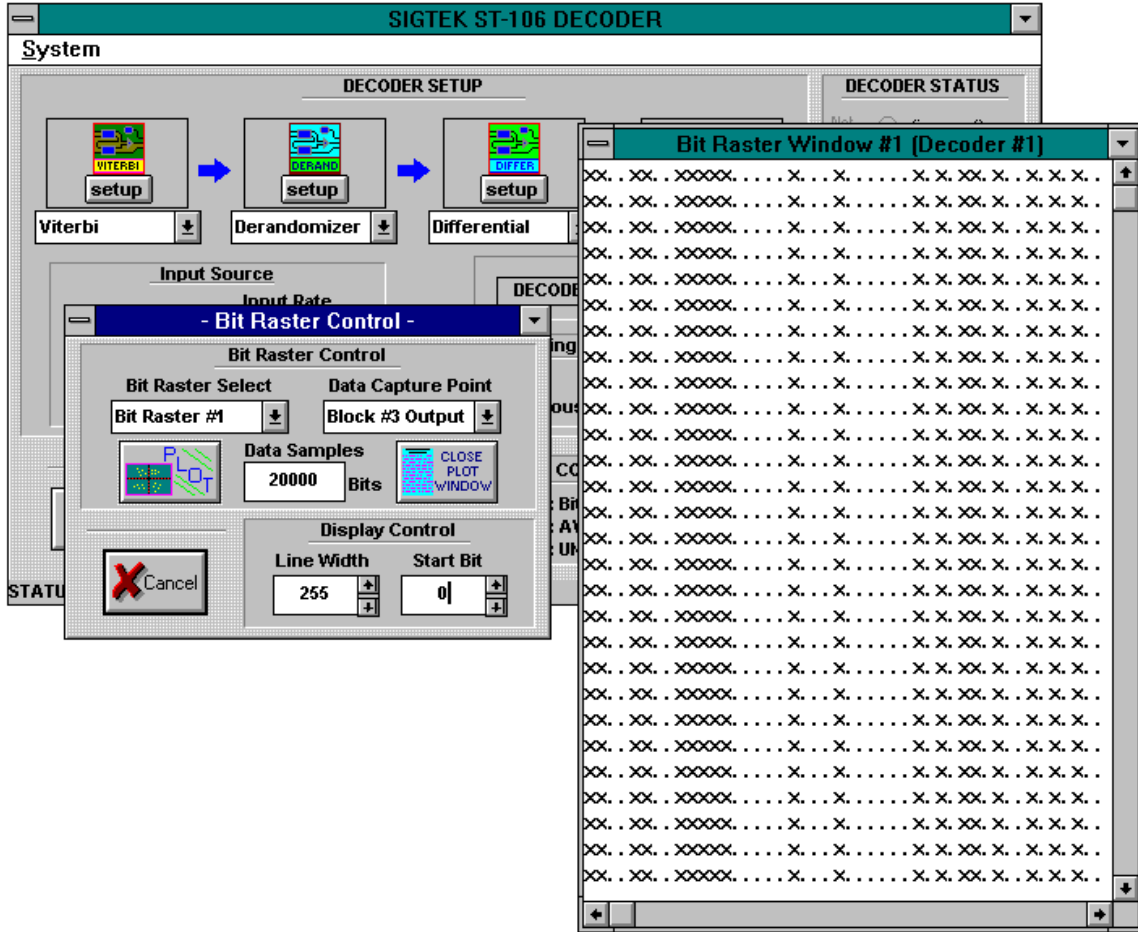


Figure 6–11 Differential Decoder Output Bit Raster (R8 LRS)

7. ST-106 Decoder Operation

7.1 Start-Up

To start the ST-106 control software from Windows, open the SIGTEK program group and double click on the ST-106 decoder icon. The ST-106 main control menu shown in Figure 7–1 will appear.

7.2 Main Menu

The ST-106 main control menu is shown in Figure 7–1. The main control menu consists of four pull-down menus which are used to configure the flow of decoding operations. Configuring the decoding flow will be discussed in Section 7.2.2. The main menu also contains other miscellaneous functions which will be described in the remainder of this section.

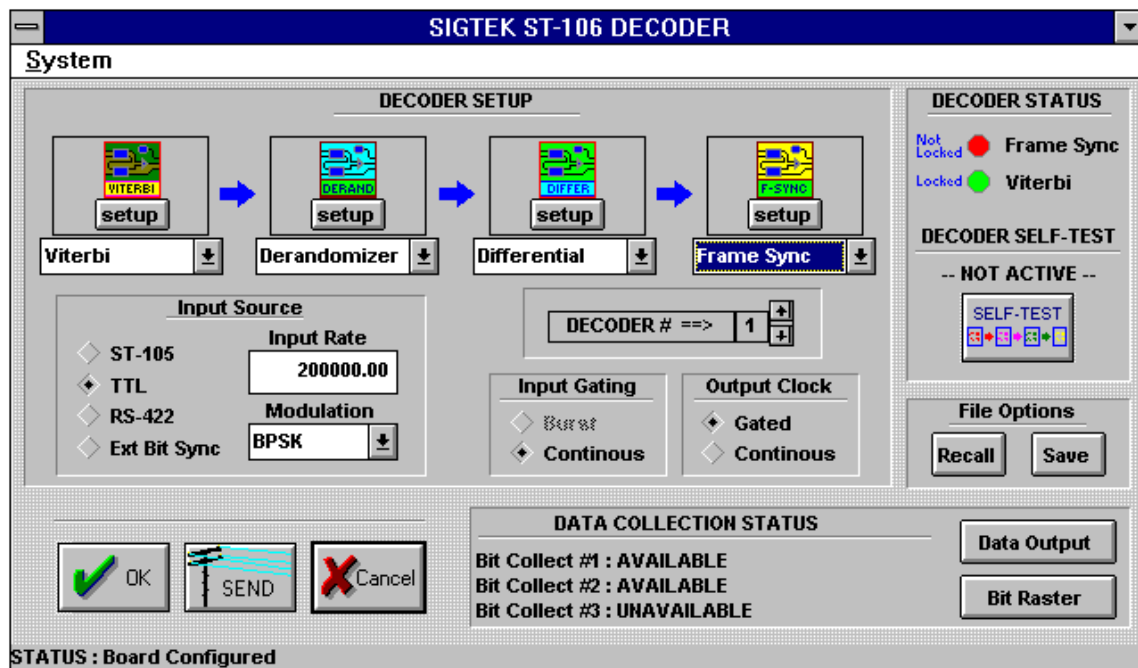


Figure 7–1 Main Decoder Control Menu

The ST-106 Decoder main control menu is organized as follows: Decoding operations are configured using the series of four pull-down menus as shown in Figure 7–1. The operator can select from a list of several decoding functions to perform in each block. Arrows between the blocks indicate the direction of signal flow. Lock status is provided in the upper left portion of the menu. Other menu selections control input signal selection, data

storage to disk, bit rastering of data, and output clocking options. The *File Options* area is used to save and to recall setup files for individual boards. The details of these control mechanisms is described in greater detail in the following sections.

Any parameter changes made in the menu are not accepted until the user clicks on the *SEND* button in the bottom left corner. The *OK* button performs the same function as the *SEND* button except it also closes the control window. A *STATUS:* line at the bottom of the menu indicated if parameters have been changed requiring the *SEND* button to be pressed.

7.2.1 System Configuration Menu

The *System Configuration* menu shown in Figure 7-2 is accessed by selecting the *System Configuration* option in the *System* pull-down menu. The *System Configuration* Menu is where the I/O address for the ST-106 is set. It also allows multiple ST-105 and ST-106 boards to be configured in a system. After all changes to the *System Configuration* menu have been made, the *OK* button is pressed to accept the changes and return to the top level system display.

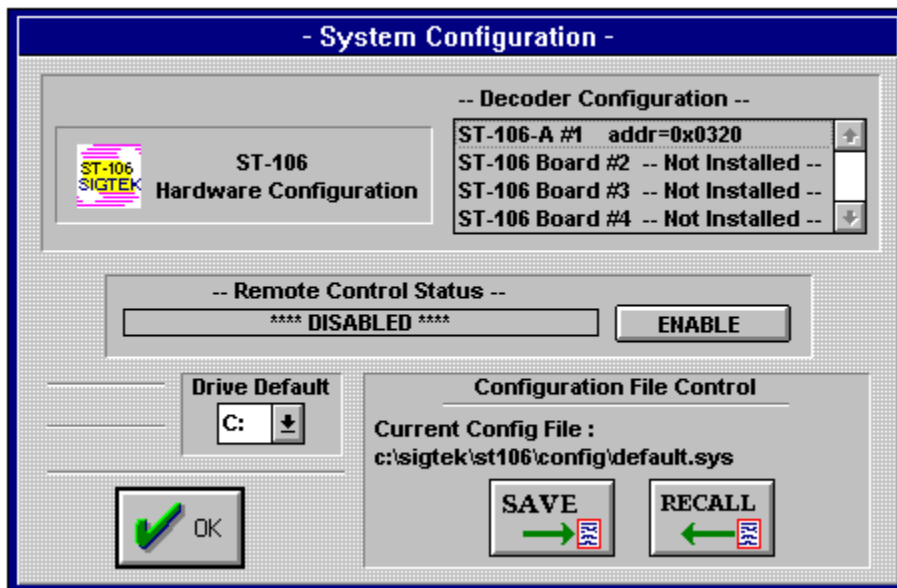


Figure 7-2 System Configuration Menu

7.2.1.1 Setting Board Address

To set a board I/O address for the ST-106 Decoder from the *System Configuration* Menu, double click on the board listed under Decoder Configuration. For example, double clicking on Decoder #1 results in the *Board Configuration* menu shown in Figure 7-3.

To modify the board address, select the *Installed* option under *Address Options*. The board address appears in hexadecimal notation and can be modified either by direct keyboard entry or using the arrows shown to increment or decrement the value. (The "0x" in front of the number is used to denote hexadecimal notation). The address entered must match the address jumper selection on the board as discussed in Section 5.1.

Also in the *Board Configuration* menu is an *Initial Configuration File* Option. This allows the operator to select from previously saved Decoder configuration files to be used to initially configure the board. Saving board configuration files is discussed later in this section. Decoder configuration files always end in the file extension .dcd.

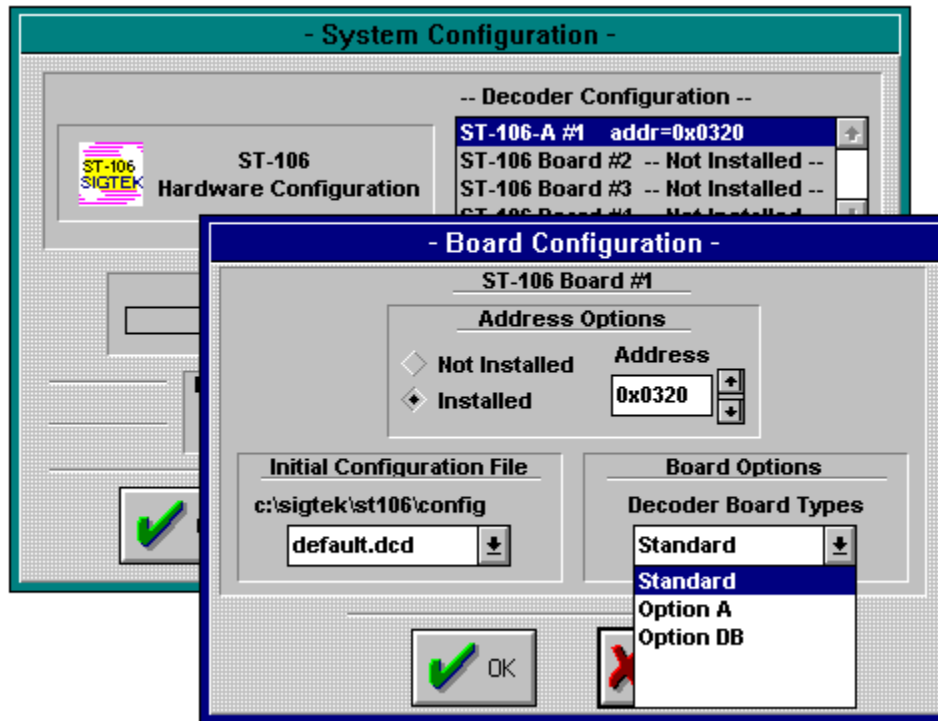


Figure 7-3 Board Configuration Menu

The decoder Board Configuration Menu has a block called board options which allows selection of three decoder board types: *Standard*, *Option A* and *Option DB*. Standard is for the standard ST-106 board configuration. Option A is for the ST-106A which is a standard ST-106 with the XC4005 FPGA chips replaced with XC4010 chips for special applications. Option DB is for a ST-106 with a daughter board attachment. The daughter board will implement additional functions such as sequential decoding and L1 demuxing.

Once all changes have been made to the *Board Configuration* menu, the *OK* button is pressed to accept the changes and return to the *System Configuration* menu. Pressing the *Cancel* button ignores any changes that may have been made and returns to the *System Configuration* menu.

7.2.2 Decoder Setup Parameters

7.2.2.1 ST-106 Board Number (Selecting a Specific ST-106)

Multiple ST-106 boards may be installed in a computer. The board select scroll box shows which ST-106 is currently attached to the control software.

7.2.2.2 Selecting Input Signal Source

The ST-106 accepts encoded data and clock from an external source on one of four inputs listed under *SOURCE* on the main menu. The four inputs are:

- 1) ST-105 Assumes 3-bit soft decision data and clock is input from an ST-105 Demodulator on ST-106 connector J6 (See Section 6).
- 2) TTL Assumes single-bit hard decision data at TTL levels is input on the 50 Ω SMA connector J1 on the board mounting bracket and clock is input on the 50 Ω SMA connector J2.
- 3) RS-422 Assumes single-bit hard decision data and clock at RS-422 differential levels is input on the 10 pin connector J8 on the board top edge. (See Section 6 for J8 pin-out information).
- 4) Ext Bit Sync Assumes single-bit hard decision data and clock is input from an external bit sync on ST-106 connector J6 (See Section 6).

7.2.2.3 Selecting Input Rate and Modulation Type

A coarse estimate of the input bit rate (also referred to as the encoded symbol rate or channel rate) is required by the ST-106 for correct clock generation. The *Input Rate* ranges from 2.4 kbps to 2.1 Mbps.

The *Modulation* parameter is used to determine the Viterbi decoder's node synchronization method. Node synchronization refers to the technique used by the Viterbi decoder to determine the order in which encoded symbols are received. When the modulation type is set to BPSK, the viterbi decoder will attempt to determine which symbol is the first symbol of a codeword. When set to QPSK, the decoder also attempts to account for the 90^o phase ambiguity associated with QPSK demodulators.

When the input source is set to ST-105, the *Input Rate* and *Modulation* parameters are automatically updated whenever the *Baud Rate* and *Modulation* parameters are entered in the ST-105 Menu. When the input source is set to TTL, RS-422 or Ext Bit Sync, the *Input Rate* and *Modulation* parameters must be manually entered.

7.2.2.4 Output Clock Select

The main decoder menu has an option for control of the output clock. If the *Output Clock* option is set to *Gated* (default), then the decoder output clock (found on connectors J8, J4 and J5) is output only when the decoder achieves lock. The lock signal can be generated either by the Viterbi Decoder or the Frame Sync functions. If both a Viterbi decoder and a Frame Sync are present in the decoding chain, output clock gating will be determined by the lock status of the last function in the chain (i.e., the function closest to the output). Gating the output clock can be useful if it is desired to clock data into a post processing device only when a valid lock is achieved. Data is valid on the rising edge of the output clock.

If the *Output Clock* option is set to *Continuous*, then the decoder outputs a clock continuous clock independent of the lock status.

7.2.2.5 Input Gating Select

In addition to data and clock, the ST-105 demodulator also supplies the ST-106 with a burst activity indicator which is active when the ST-105 is demodulating a burst of a short duration signal such as a TDMA signal. The main decoder menu has an Input Gating option which allows data to be collected only during bursts when the ST-106 is receiving data from an ST-105 demodulator.

When the *Input Source* is set to ST-105 on the decoder menu and the *Input Gating* set to *Burst*, then when the ST-105 is in burst mode processing (see Section 5.3), the ST-106 decoder will only input data from the ST-105 when the burst signal is active. This is useful in order to gate out the invalid random data which the demodulator outputs between signal bursts. When *Input Gating* is set to *Continuous*, all data received from the ST-105 is processed, regardless of the status of the burst indicator signal from the ST-105. Note that if the ST-105 is in continuous processing mode, the burst activity indicator output is always active.

Since the Input Gating function relies on the burst activity indicator from the ST-105, this function is only available when the decoder Input Source is set to ST-105. When the other input sources are selected, the *Input Gating* defaults to *Continuous* and all data input to the decoder is processed. This would be the case if the ST-106 data and clock input was connected to the output of an external bit sync and no burst activity signal was available.

7.2.2.6 Decoder Self-Test

The decoder has a self-test mode in which the decoder can generate an encoded and randomized data stream of known structure. This data stream can be decoded and derandomized by the board in order to exercise its major functions. The self-test mode is

entered by first pressing the *Self-Test* button on the main decoder menu. The self-test setup menu is shown in Figure 7-4.

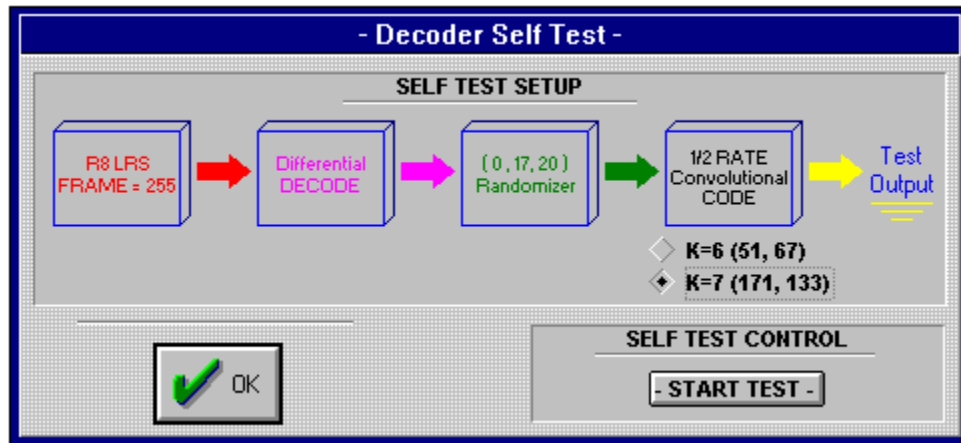


Figure 7-4 Decoder Self-Test Menu

The stream of data generated by the self-test consists initially of an R8 LRS clocked at a 100 kbits/sec rate. This LRS will repeat every 255 bits. The LRS is then randomized with a V.35 feedthru randomizer, differentially encoded, then FEC (forward error correction) encoded with a rate 1/2 K=7 or K=6 Viterbi (convolutional) code. The rate 1/2 Viterbi code results in a 200 kbits/sec transmitted data stream. The flow of the signal construction is illustrated graphically in the self-test menu.

The self-test data stream is generated by pressing the *Start Test* button in the self-test menu. This causes the decoder to output test data and clock which can be observed on rear panel connectors J3 and J4 respectively. The self-test is disabled by pressing the *Stop Test* button which appears while the test is active. The self-test is also disabled whenever a new decoder configuration is downloaded.

A description of a board loop back test based on based on the self-test data generation can be found in Section 6.

7.2.2.7 Lock Status

The lock status for the Viterbi decoder and Frame Sync functions is displayed in the upper right corner of the main menu. The indicators display red when the function is not locked and green when locked. The text “Locked” and “Not Locked” is also displayed next to the indicator.

7.2.3 Configuring the Decoder

The decoding operations are configured using the series of four pull-down menus designated Block #1 through Block #4 as shown in Figure 4-16. Arrows between the blocks indicate the direction of signal flow. The input to the first block (Block #1) is always from the selected input source to the ST-106. The output of the last block (Block #4) is always routed to the output of ST-106. The user has the following decoding functions from which to choose:

- Viterbi Decoding
- Derandomization
- Differential Decoding
- Frame Sync
- Take & Skip (previously called Decimation)
- Bypass

By selecting various decoding functions from the pull-down menus, the user can chain together the correct series of decoding steps to decode a signal of interest. The pull-down menu structure allows the order of these decoding steps to be easily interchanged. Pressing the "Setup" button below the block brings up the setup menu for the particular function selected in that block. The Bypass function is used to bypass a particular block and has no setup menu.

When parameters within a setup menu are changed and the setup menu exited, the status message "STATUS: Decoder flow updated, hit SEND to re-configure board" will appear in the lower left corner of the main menu. When this message appears, the SEND button must be pressed in order to download the new configuration parameters to the decoder. The remainder of this section discusses the setup menus for each decoding function.

7.2.3.1 Viterbi Decoder Menu

The Viterbi Decoder setup menu is shown in Figure 7-5. Selecting this function permits decoding of constraint length 6 and 7 convolutional codes typically used for forward error correction. Note that only one viterbi decoder at a time can be actively decoding. The RESET button issues a reset command to whichever decoder chip is currently selected.

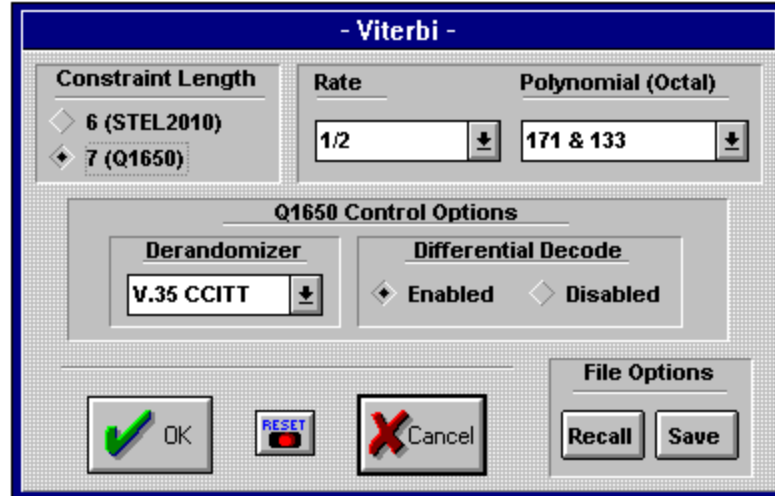


Figure 7-5 Viterbi Decoder Menu For Constraint Length 7 (K=7)

7.2.3.1.1 Constraint Length 7 (K=7) Decoding

The constraint length 7 (K=7) decoder setup menu is shown in Figure 7-5. The ST-106 uses a Qualcomm Q1650 Viterbi decoder IC to perform the K=7 decoding function. The user selects from the following code rates: 1/3, 1/2, 3/4 and 7/8. The encoding polynomials associated with the selected code rate are displayed in octal. Figure 7-6 illustrates the implementation of the encoder polynomials for the given code rates. The encoder polynomials used by the Q1650 are the industry standard K=7 polynomials used on such systems as INTELSAT IDR and IBS service.

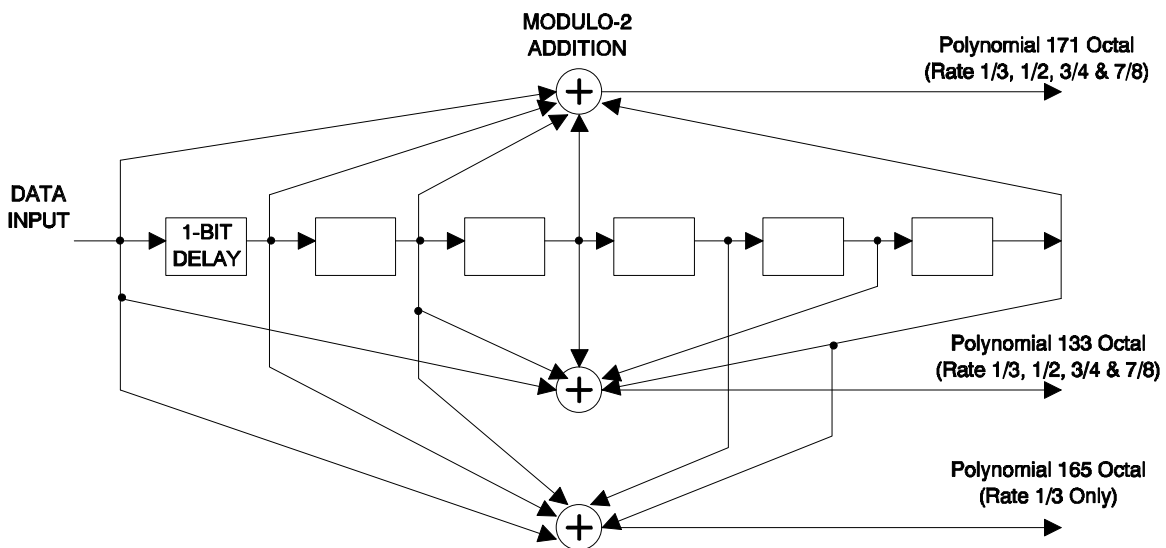


Figure 7-6 Q1650 Encoder Polynomials

The Q1650 has built in differential decoding and derandomization functions which can be enabled or disabled from the setup menu. Figure 7-7 illustrates the order in which these functions are performed in the Q1650. The derandomizer implemented by the Q1650 is a 20 tap V.35 feedthru derandomizer illustrated in Figure 7-8. The selection of V.35 INTELSAT and V.35 CCITT refers to how the derandomizer's "adverse state" is determined. The purpose of the adverse state is to guarantee an output transition once every 32 bits for long sequences of 0s or 1s.

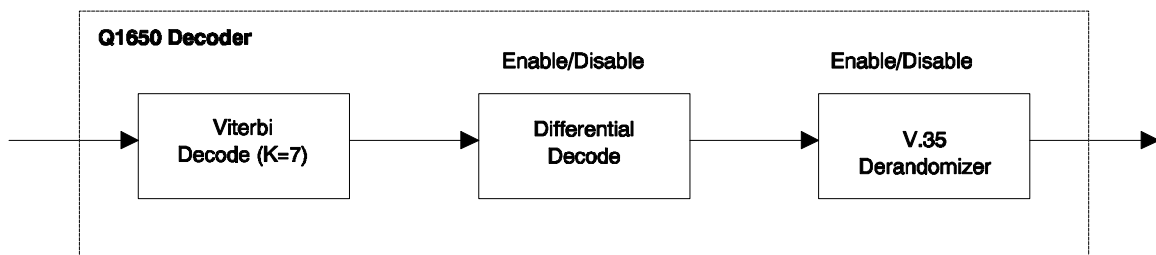


Figure 7-7 Q1650 Decoder Internal Processing Functions

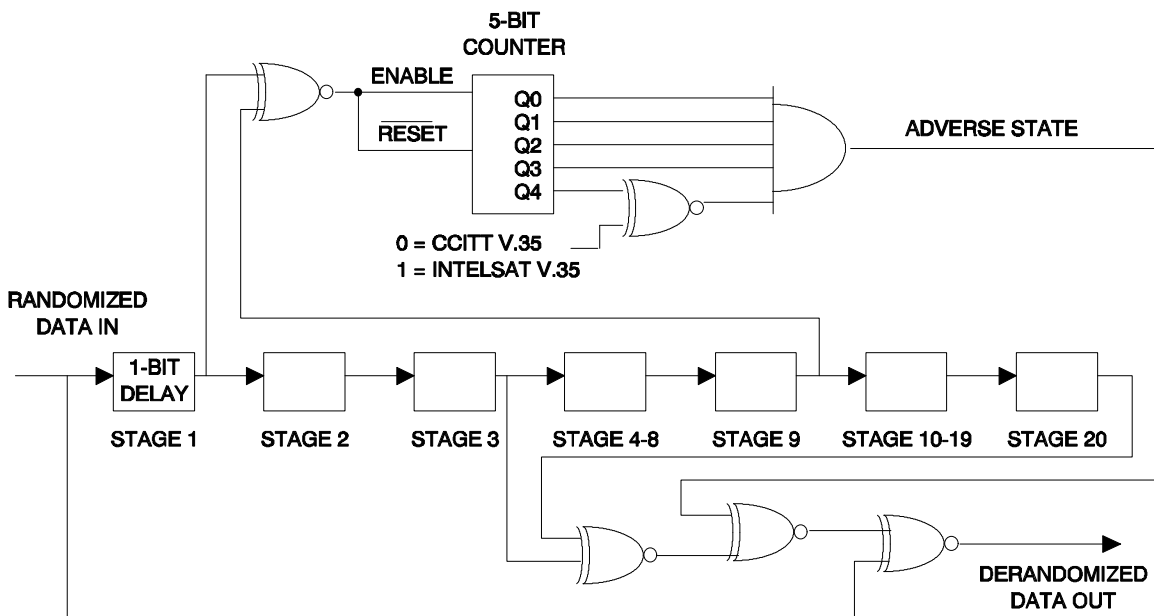


Figure 7-8 Q1650 V.35 Derandomizer Implementation

7.2.3.1.2 Constraint Length 6 (K=6) Decoding

The constraint length 6 (K=6) decoder setup menu is shown in Figure 7–9. The ST-106 uses a STEL-2010 Viterbi decoder IC to perform the K=6 decoding function. The user selects from code rates 1/3 and 1/2. The encoding polynomials for rate 1/2 are 51 and 67 octal and for rate 1/3 are 51, 67 and 75 octal. The STEL-2010 Decoder has no internal derandomizer or differential decoder options like the Qualcomm Q1650 discussed in the previous section.

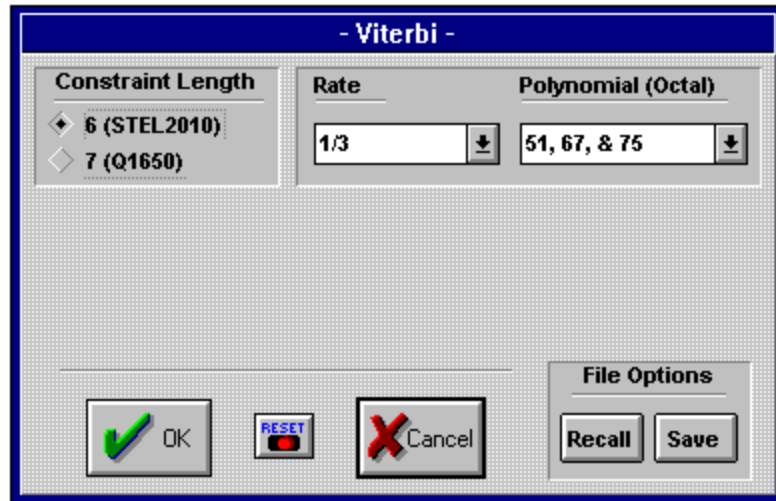


Figure 7–9 Viterbi Decoder Menu For Constraint Length 6 (K=6)

7.2.3.2 Differential Decoder Menu

The differential decoder setup menu is shown in Figure 7–10. The differential decoder performs the function illustrated in Figure 7–11. There are only controls associated with the differential decoder. The Enabled / Disabled determines if the differential decoder is active. Selecting Disabled for the differential decoder essentially bypasses the block. The Output control permits inversion of the differential decoder output. Note that only one differential decoder is allowed in the processing flow.

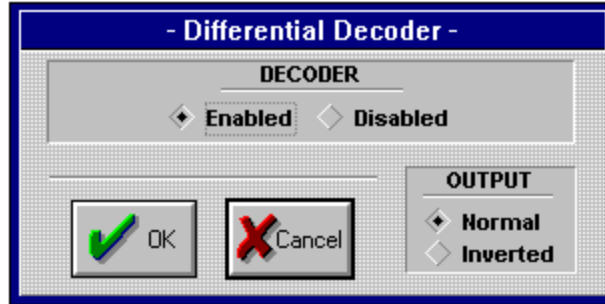


Figure 7–10 Differential Decoder Setup Menu

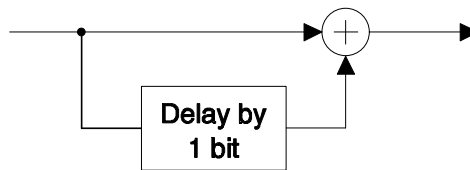


Figure 7–11 Differential Decoder Implementation

7.2.3.3 Derandomizer Menu

The ST-106 implements two types of 24-tap programmable derandomizers: a feedthru (also called self-synchronizing) derandomizer and an additive (also called synchronous) derandomizer. The differences between the two and their setup menus are discussed in the following sections.

7.2.3.3.1 Feedthru Derandomizer Menu

The feedthru derandomizer setup menu is shown in Figure 7–12. The function implemented by the feedthru derandomizer is illustrated in Figure 7–13. The feedthru derandomizer has 24 programmable feedback tap points. The tap points are entered by specifying the exponents of the polynomial which represents the derandomizer. The exponents may be entered in any order and can be separated by any non-number character (such as spaces, commas, etc.).

Figure 7–14 illustrates the selection of tap points for derandomizer specified by (0,17,20) corresponding to the binary polynomial $1 + x^{17} + x^{20} = x^0 + x^{17} + x^{20}$. This corresponds to a 20 tap feedthru derandomizer with the same tap points as the V.35 derandomizer discussed in Section 7.2.3.1.1. The *V.35 Default* button can be used to quickly set the taps for the industry standard V.35 randomizer (0,17,20).

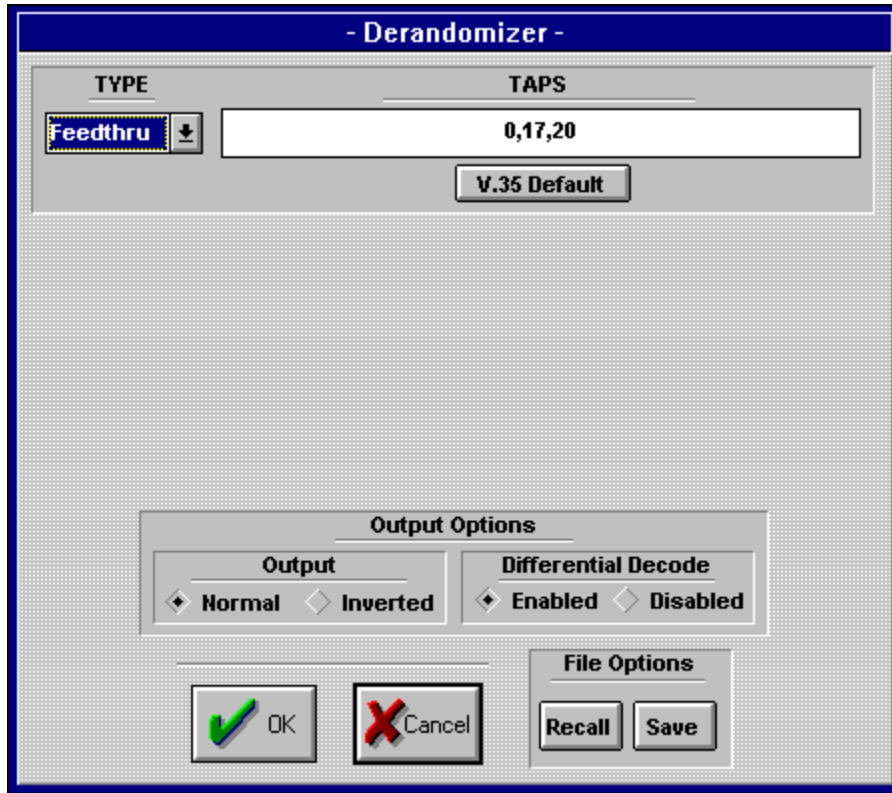


Figure 7-12 Feedthru Derandomizer Setup Menu

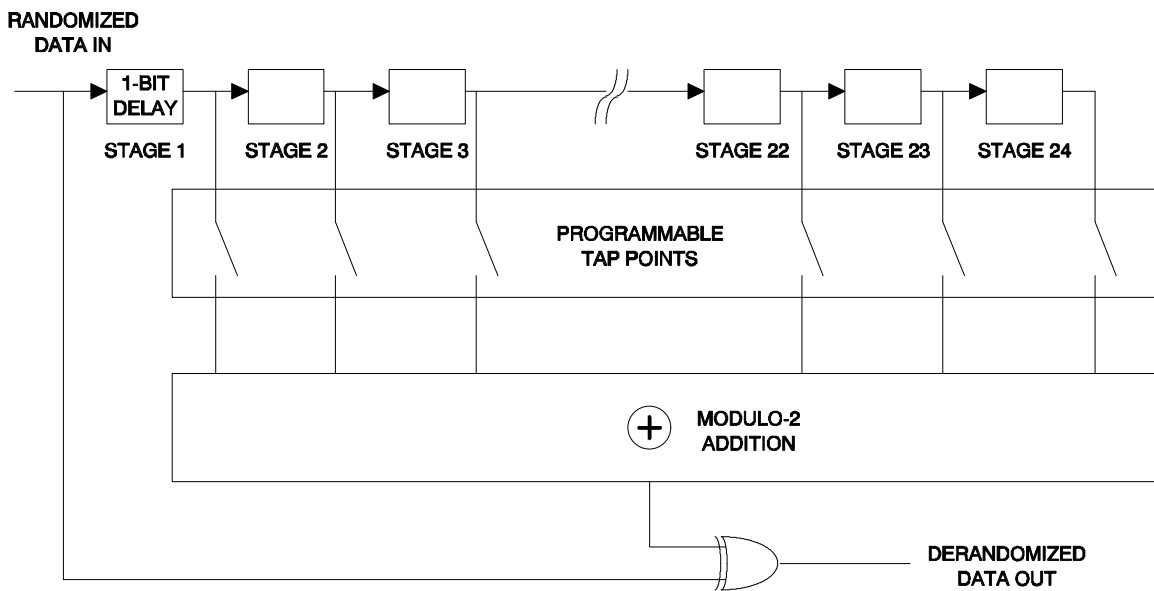


Figure 7-13 Programmable Feedthru Derandomizer

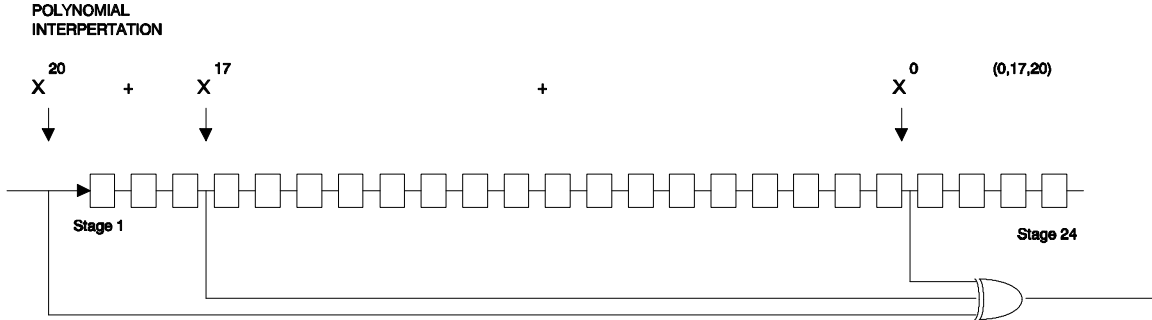


Figure 7–14 Feedthru Derandomizer Example (Taps 0,17,20)

Two controls are available to control the format of the derandomizer output. The *Output* control can be set to forward the data in normal polarity or invert the data before forwarding to the next function. The *Differential Decode* control is used to enable or disable a differential decoder circuit at the output of the derandomizer. This differential decoder performs the same function as discussed in Section 7.2.3.2. The differential decode output option is included in the derandomizer menu since it is common practice to use a differential decoder immediately after derandomization.

7.2.3.3.2 Additive Derandomizer Menu

The additive derandomizer setup menu is shown in Figure 7–15. The function implemented by the additive derandomizer is illustrated in Figure 7–16. The additive derandomizer implements a 24 tap linear recursive sequence (LRS) shift register circuit with a 24 bit initial fill. A truncation counter can be used to set the frame length of the derandomizer LRS. The initial fill is loaded upon detection of a unique word which is used to synchronize the derandomizer's LRS to that of the incoming signal. In this implementation, it is assumed that the unique word occurs in front of a block of randomized data and the unique word itself is not randomized as illustrated in Figure 7–17

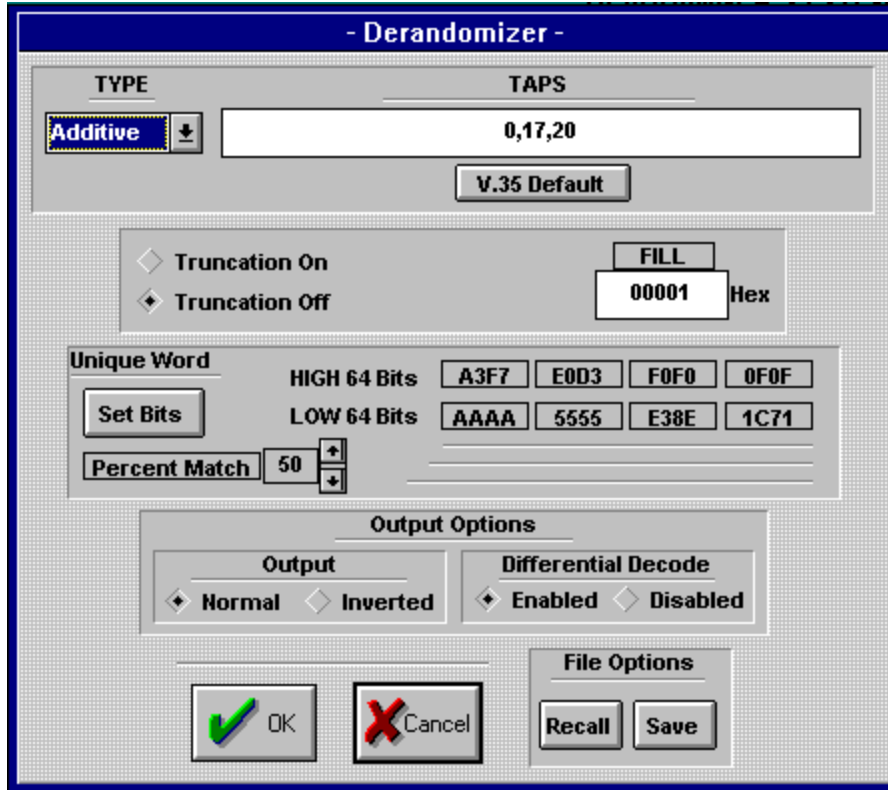


Figure 7-15 Additive Derandomizer Menu

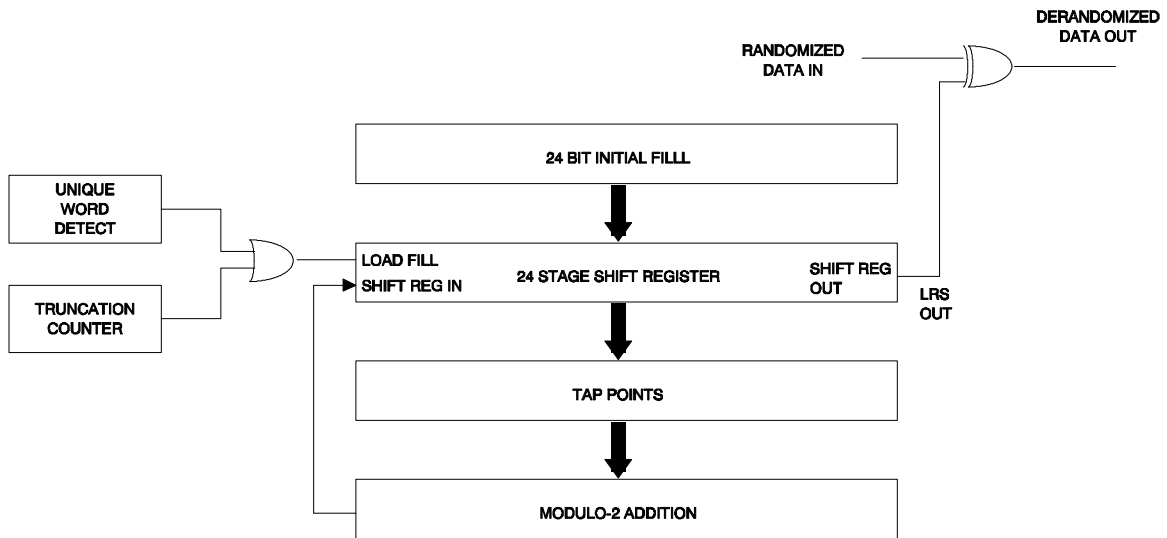


Figure 7-16 Additive Derandomizer Implementation



Figure 7–17 Additive Derandomizer Signal Model

The tap points are entered by specifying the exponents of the polynomial which represents the derandomizer. The *Initial Fill* parameter is specified on the menu by a 6 digit hexadecimal value. The truncation frame length is entered in decimal and has a maximum value of 32767.

The unique word pattern used to synchronize the start of the additive derandomizer can be up to 128 bits long. The unique word is set by hitting the *Set Bits* button which brings up the unique word edit menu shown in Figure 7–18. The unique word menu contains eight 16-bit fields which specify the 128 bit unique word. The unique word fields are entered in two formats: 1) binary (1 & 0) entry which can contain don't care bits which are entered as an "x" or 2) 4 digit hex values as shown in Figure 4-30. Note that when entering values for the unique word in hex, the hex number must be preceded by "-0X" in order to be recognized as hex. This notation is only required when entering hex values for the unique word. Unique words less than 128 bits long should be padded with leading don't care characters. The *Percent Match* parameter is provided to set a threshold for unique word detection and allows detection of corrupted sync patterns.

As with the feedthru derandomizer menu, two controls are available to control the format of the derandomizer output. The *Output* control can be set to forward the data in normal polarity or invert the data before forwarding to the next function. The *Differential Decode* control is used to enable or disable a differential decoder circuit at the output of the derandomizer. This differential decoder performs the same function as discussed in Section 7.2.3.2. The differential decode output option is included in the derandomizer menu since it is common practice to use a differential decoder immediately after derandomization.

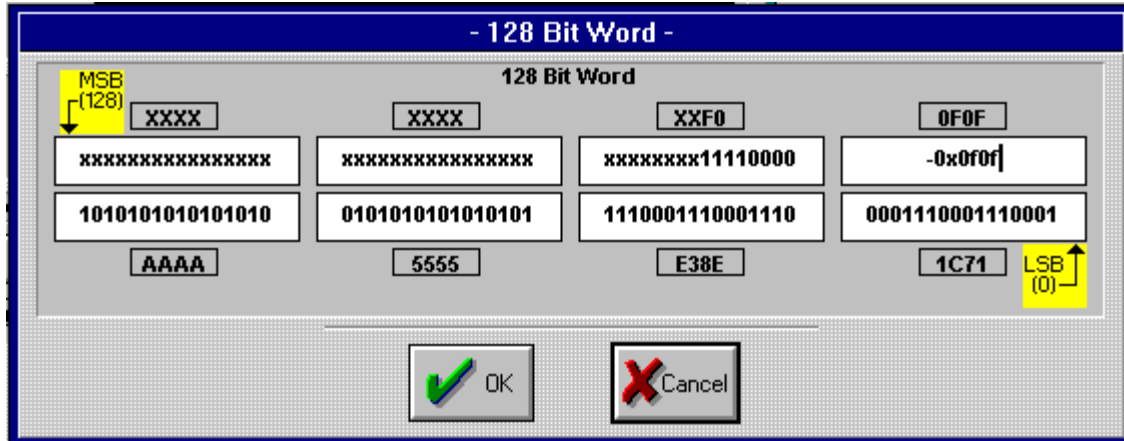


Figure 7–18 Unique Word Edit Menu

7.2.3.4 Frame Synchronizer Menu

Three types of frame synchronization circuits are available to determine the beginning and end of data frames within a decoded and derandomized data stream. The three types of generic frame synchronizers are described in the following sections.

7.2.3.4.1 Idle Based Frame Sync

The Idle Based Frame Synchronizer setup menu is shown in Figure 7–19. The Idle Based frame sync extracts frames of data which are separated by repeating idle patterns as illustrated in Figure 7–20. The circuit detects and synchronizes to the occurrence of idle patterns and will only forward data between idle patterns.

The idle pattern is entered as a Unique Word of up to 128 bits. The unique word is set by hitting the *Set Bits* button in the same manner as described in 7.2.3.3.2 for the additive derandomizer unique word. The unique word menu contains eight 16-bit fields which specify the 128 bit unique word. The unique word fields are entered in two formats: 1) binary (1 & 0) entry which can contain don't care bits which are entered as an "x" or 2) 4 digit hex values as shown in Figure 7–18. Note that when entering values for the unique word in hex, the hex number must be preceded by "-0X" in order to be recognized as hex. This notation is only required when entering hex values for the unique word. Unique words less than 128 bits long should be padded with leading don't care characters.

The *Frame Inversion* feature when enabled detects the occurrence of an inverted unique word and automatically outputs data frames with the correct polarity.

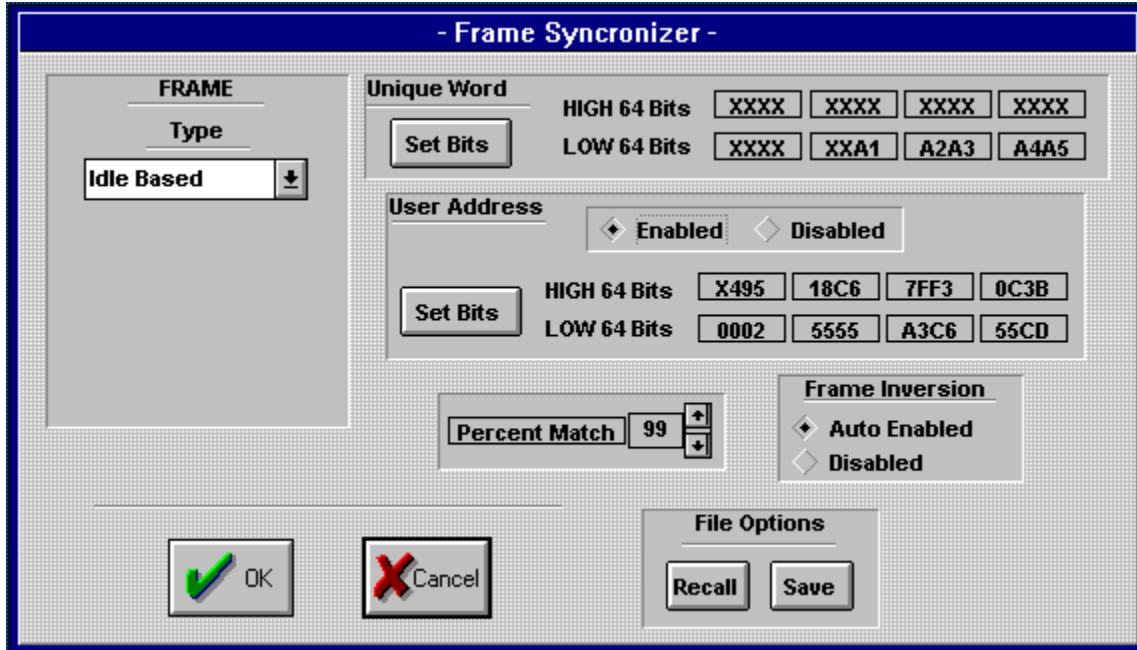


Figure 7–19 Idle Based Frame Synchronizer Setup Menu

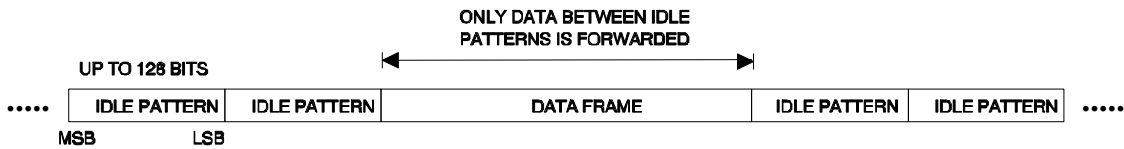


Figure 7–20 Idle Based Frame Example

The User Address Search function permits searches for patterns of up to 128 bits at the start of a frame. If the User Address function is enabled, then only frames containing the User Address pattern within the first 128 bits of the frame will be forwarded. If the User Address function is disabled, all frames are forwarded. The User Address pattern is set by hitting the *Set Bits* button in the same manner as described for editing the unique word.

7.2.3.4.2 Fixed Length Frame Sync

The Fixed Length Frame Synchronizer setup menu is shown in Figure 7–21. Operation of the Fixed Length frame sync is illustrated in Figure 7–22. The occurrence of the unique word indicates the start of a frame. Data is then forwarded for a fixed number of bits given by the Frame Length parameter. The maximum frame length is now 262,144 bits. Entering the unique word is identical to the method described for the Idle Based frame sync. Operation of the Percent Match, Frame Inversion and User Address Search is also the same.

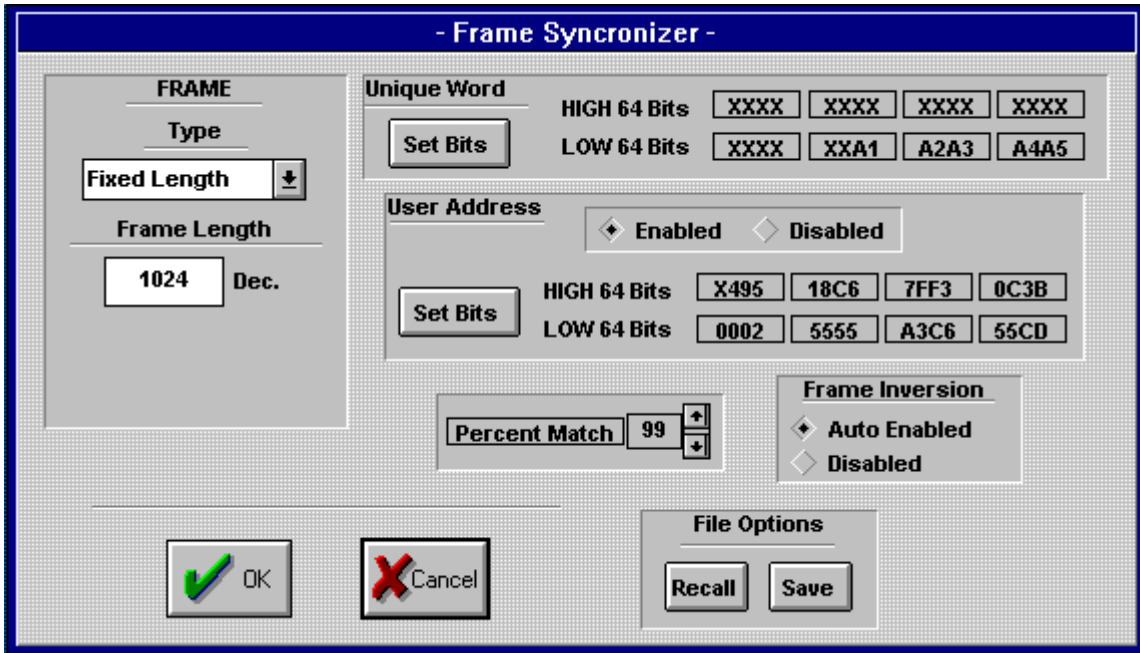


Figure 7–21 Fixed Length Frame Synchronizer Setup Menu

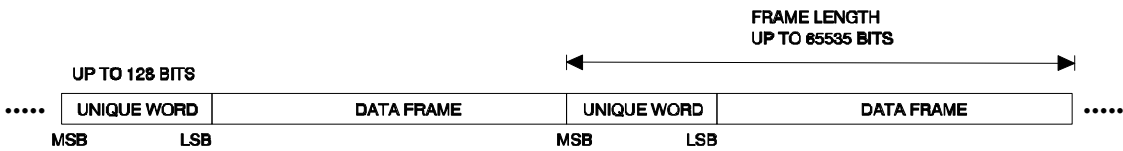


Figure 7–22 Fixed Length Frame Example

7.2.3.4.3 Variable Length Frame Sync

The Variable Length Frame Synchronizer setup menu is shown in Figure 7–23. Operation of the Variable Length frame sync is illustrated in Figure 7–24. The occurrence of the unique word indicates the start of a frame. The synchronizer circuit extracts a 16 bit frame length value from a location in the data frame given by the Length Location parameter. The maximum frame length is 65535 bits. Entering the unique word is identical to the method described for the Idle Based and Fixed Length frame sync. Operation of the Percent Match, Frame Inversion and User Address Search is also the same.

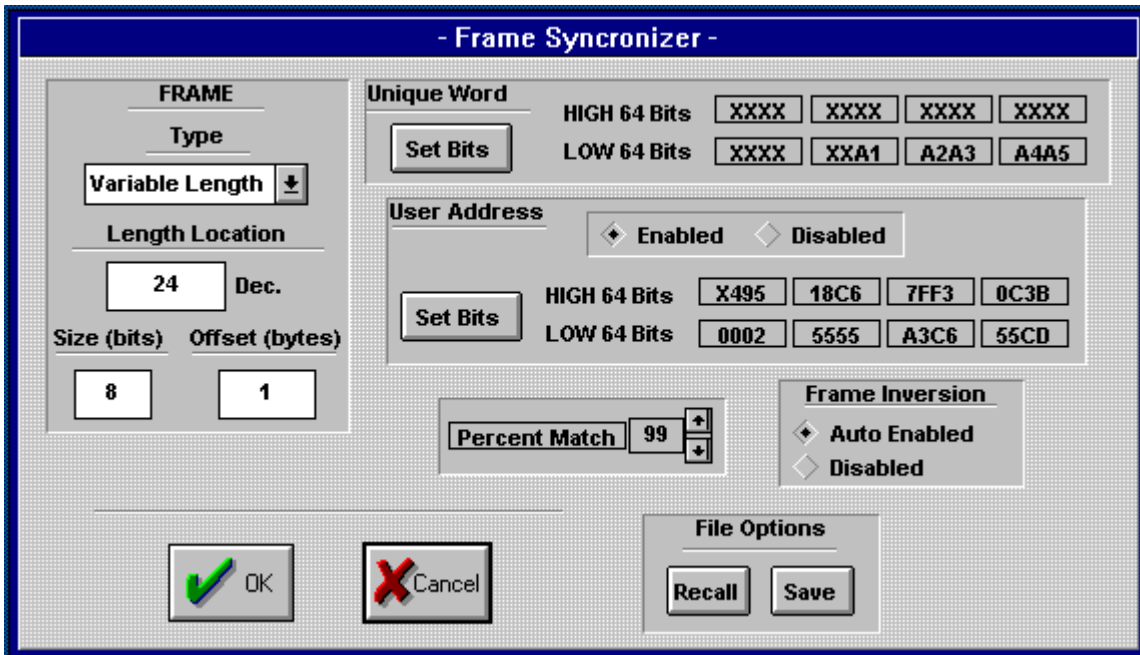


Figure 7–23 Variable Length Frame Synchronizer Setup Menu

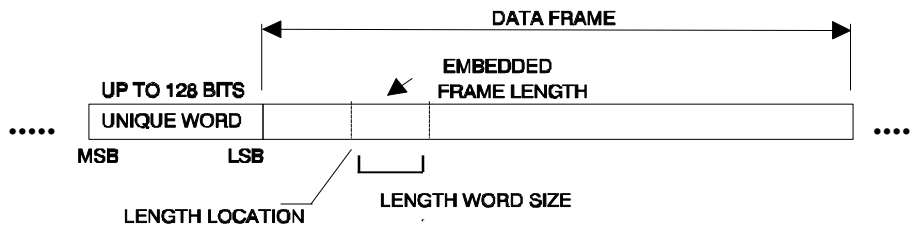


Figure 7–24 Variable Length Frame Example

7.2.3.5 Take & Skip Function

The Take & Skip menu (previously called Decimation) is shown in Figure 7–25. The Take & Skip function removes selected bits from the data stream. The function will continuously remove one bit out of a selected number of bits. For example, “Skip 1 of 2 Bits” removes every other bit from the data stream; “Skip 1 of 3 Bits” removes every third bit and so on. The take & skip can be manually “stepped” causing the start bit of the skip to be shifted by 1 each time the STEP button is pressed. (see Figure 7–26).

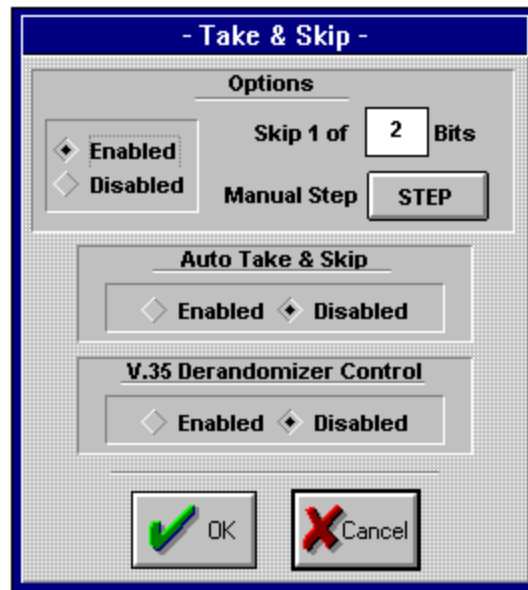


Figure 7–25 Take & Skip Function Menu

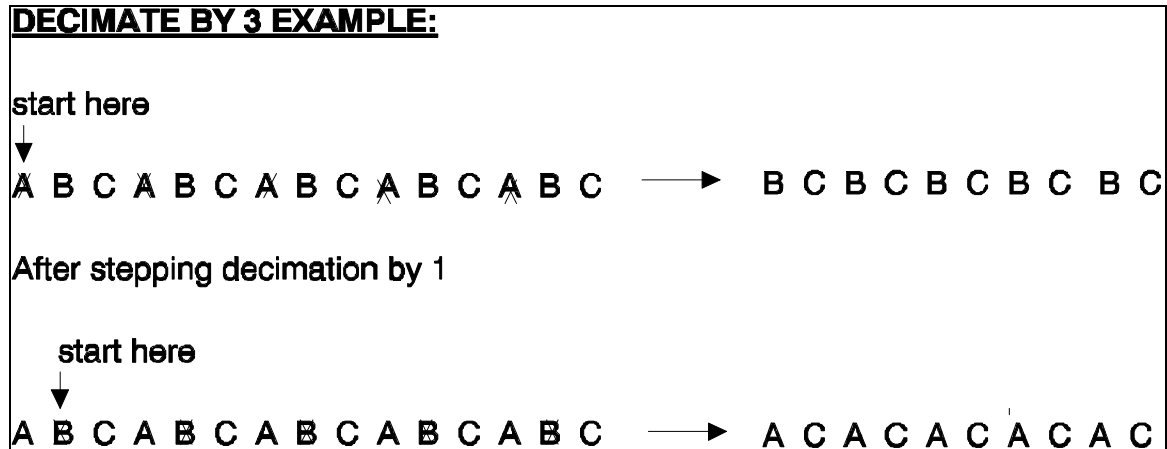


Figure 7–26 "Take and Skip" Example

The *Auto Take & Skip* function is planned for future versions and is not currently enabled. It would automatically step the skip until frame sync lock is achieved. The V.35 Derandomizer control allows a V.35 feedthru randomizer (0,17,20) to follow the Take & Skip without requiring occupying a separate processing block on the main menu.

The take & skip function is useful for removing the forward error correction bits from 1/2 rate systematic convolutional codes. With 1/2 rate systematic convolutional codes one information bit is transmitted followed by a parity bit. Decimation by 2 can be used to extract the data and discard the FEC. The Manual Step button may have to be used if the decimation starts on the wrong bit (i.e., starts extracting the FEC parity instead of the data).

7.2.4 Other Decoder Controls and Displays

7.2.4.1 SEND Button

The *SEND* button is used to affect any changes made to system setup parameters.

7.2.4.2 Cancel Button

The *Cancel* button is used to abort any changes made to system setup parameters before issuing a *SEND* command.

7.2.4.3 Saving and Recalling Decoder Configurations

Complete decoder configurations can be saved to disk by pressing the "Save" button under the File Options in the main menu. A standard Windows file access dialog shown in Figure 7-27 will appear and allows the name and destination of the configuration file to be entered. Pressing the "Recall" button allows saved configurations to be recalled from disk. Decoder files are always saved with a .dcd file extension.

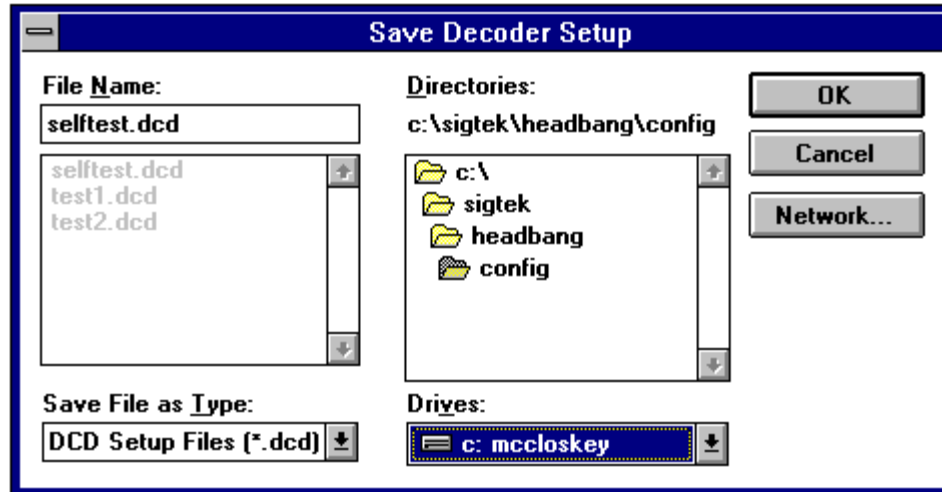


Figure 7–27 Decoder Configuration File Save

7.2.4.4 Saving Data to Disk

Data at any point in the decoding chain can be stored to disk. Data storage is initiated by pressing the *Data Output* button on the ST-106 main menu which brings up the *Output File Control* menu shown in Figure 7–28. Up to three files can be open at one time storing data from different points in the processing chain to disk. The number of files which can be opened is determined the board resources remaining after defining the processing in the decoding chain. Some processing functions (such as take&skip and derandomization) use board resources which are normally used to collect and transfer bit stream data to the computer. *Bit Collect* status is marked as *UNAVAILABLE* if there are not enough remaining board resources for bit collection. (This concept of board resources is discussed in more detail in Section 6).

A destination file name is entered by pressing the *Change Output File* button which presents a standard Windows file dialog box. The point in the processing chain from which data is collected is set by the *Data Capture Point* selection. Data from the decoder input or the output of any of the processing blocks defined in the main menu can be selected for storage to disk. Data storage is initiated by pressing the *Start Collect* button and terminated by pressing the *Stop Collect* button. A status indicator under *Collection Control* tells whether the data storage is active or inactive. Collection is also terminated whenever the decoder configuration is changed from the main menu.

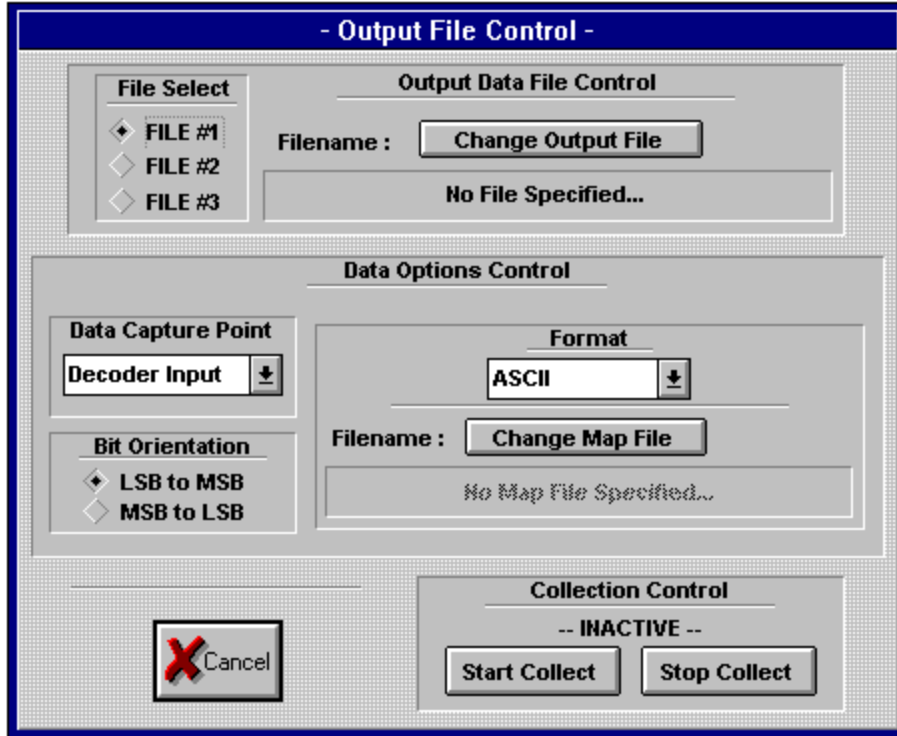


Figure 7–28 Output Data File Control Menu

Data can be stored in one of four formats: ASCII, HEX, Binary or Character Mapped ASCII. There is also the option to change the bit orientation of data bytes stored to disk. The best way to illustrate the bit ordering of the ST-106 data storage to disk is with an example. For this example, it is assumed that the following 120 bit (15 byte) data stream is input to the ST-106 decoder:

A1 A2 A3 A4 A5 41 53 43 49 49 20 54 45 53 54

The data stream is shown in hex representation. Byte A1 is transmitted first with bit order 10100001 followed by the remainder of the frame. We have adopted the convention of calling the first bit of a received 8-bit byte the MSB (most significant bit). If the first 40 bits of the data stream shown above are considered a sync pattern (A1 A2 A3 A4 A5), the fixed length frame sync setup (see Section 7.2.3.4.2) shown in Figure 7–29 can be used to synchronize to the start of frame.


```

SCII TEST; c f r ASCII TEST; c f r ASCII TEST; c f r ASCII
      TEST; c f r ASCII TEST; c f r ASCII T
EST; c f r ASCII TEST; c f r ASCII TEST; c f r ASCII TEST; c f r ASCII
      TEST; c f r ASCII TEST; c f
r ASCII TEST; c f r ASCII TEST; c f r ASCII TEST; c f r ASCII
      TEST; c f r ASCII TEST; c f r ASCII
I TEST; c f r ASCII TEST; c f r ASCII TEST; c f r ASCII TEST; c f r ASCII
      TEST; c f r ASCII TEST

```

Figure 7–30 ASCII Formatted Collect (“MSB to LSB”)

The collected data file can be viewed with a standard text editor. Note however that some text editor programs such as Windows Notepad can only be used to view relatively small text files. As can be seen in Figure 7–30, the data frame contains some ASCII printable characters (0x41 = “A”, 0x53 = “S”, 0x43=“C”, and so on). The 5 byte sync pattern translate to ASCII characters ; c f r .

7.2.4.4.2 Hex Data Storage Format

The HEX format saves collected bytes to file as their equivalent hex character pair (two characters per byte). Figure 7–31 is an example of the data frame given above saved as Hex with bit orientation **MSB to LSB**. Note that the ST-106 software puts an additional space between hex bytes so they may be distinguished from one another. Since in this mode the data is stored as text characters, it can also be viewed with a text editor.

```

53 54 A1 A2 A3 A4 A5 41 53 43 49 49 20 54 45 53 54 A1 A2 A3 A4
      A5 41 53 43 49
49 20 54 45 53 54 A1 A2 A3 A4 A5 41 53 43 49 49 20 54 45 53 54
      A1 A2 A3 A4 A5
41 53 43 49 49 20 54 45 53 54 A1 A2 A3 A4 A5 41 53 43 49 49 20
      54 45 53 54 A1
A2 A3 A4 A5 41 53 43 49 49 20 54 45 53 54 A1 A2 A3 A4 A5 41 53
      43 49 49 20 54
45 53 54 A1 A2 A3 A4 A5 41 53 43 49 49 20 54 45 53 54 A1 A2 A3
      A4 A5 41 53 43
49 49 20 54 45 53 54 A1 A2 A3 A4 A5 41 53 43 49 49 20 54 45 53
      54 A1 A2 A3 A4
A5 41 53 43 49 49 20 54 45 53 54 A1 A2 A3 A4 A5 41 53 43 49 49
      20 54 45 53 54

```

Figure 7–31 HEX Formatted Collect (“MSB to LSB”)

7.2.4.4.3 Binary Data Storage Format

The Binary format saves collected bytes to without any formatting. Because it is not formatted as characters, data stored in binary format cannot be viewed with a text editor. The binary format consumes the least disk space of all the storage methods. Typically, a separate software program is used to read and interpret the binary data. The binary data can be read from the file using (in C programming) the *fread* instruction.

7.2.4.4.4 Character Mapped Data Storage Format

.The Character Mapped format saves data as ASCII characters which have a non-standard character mapping. The character mapping is defined in a file with a .map extension. Figure 7–32 gives a partial example of a character map file contents.

```

SIGTEK 1.0 MAP --- Test Data Map file.
NUM  STD ASCII  OUTOUT
00    NUL      ""
01    SOH      ""
02    STX      ""
03    ETX      ""
04    EOT      ""
05    ENQ      ""
06    ACK      ""
07    BEL      ""
08    BS       ""

LATER IN THE FILE. . . .

2F    /        ""
30    0        ""
31    1        ""
32    2        ""
33    3        ""
34    4        ""
35    5        ""
36    6        ""
37    7        ""
38    8        ""
39    9        ""
3A    :        ""
3B    ;        ""
3C    <        ""
3D    =        ""
3E    >        ""
3F    ?        "19"
40    @        "20"
41    A        E
42    B        F
43    C        G
44    D        H
45    E        I

```

Figure 7–32 Character Map File Example

The .map files are contained in directory C:\SIGTEK\ST106\OUTPUT and can be edited with a standard text editor. The first two lines in the file are for comments but are necessary for proper operation. The file is formatted in 3 columns. The first column is hex numbers from 00 to FF and serves as an index. The second column is the standard ASCII mapping of the first columns hex byte values to ASCII characters. **The first two columns must always remain unchanged.** The third column is the mapping of standard ASCII characters to a set of non-standard ASCII characters. Three options are available to set the character mapping:

Option 1: No Change in ASCII Mapping

To leave the ASCII mapping in plane for a particular character, include a set of double quotes (“”) in the thirds column. There should be no spaces between the quotes. A file called *default.map* has been provided in the OUTPUT directory which has all double quotes in the third column and can be used as a template for creating new character mapping files.

Option 2: Map character to character

To map one ASCII character to another, simply place the new character in the third column across from the standard ASCII character you wish to map. No quotes are required. (See the last 5 entries in the example above).

Option 3: Map hex value to hex value

To map one 8 bit hex value to another, place the new two character hex value in quotes in the third column across from the hex value you wish to map. (See the entries for hex values 3F and 40 in the example above).

7.2.4.5 Displaying a Data Bit Raster

In addition to being able to store data to disk, data can be collected and rastered on the screen. This is initiated by pressing the *Bit Raster* button which results in the *Bit Raster Control* window shown in Figure 7–33.

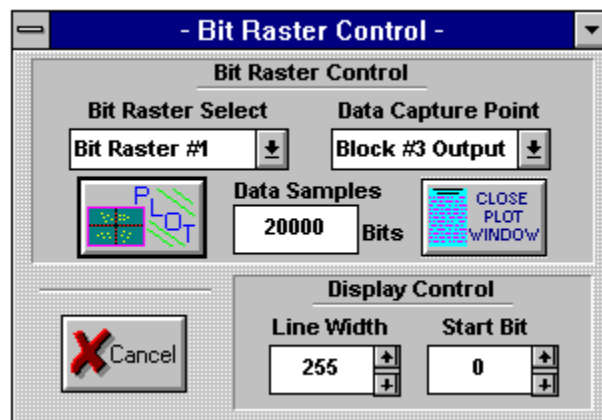


Figure 7–33 Bit Raster Control Window

As with the data storage to disk, the point in the processing chain from which data is collected and rastered is set by the *Data Capture Point* selection. Data from the decoder input or the output of any of the processing functions defined in the main menu can be selected for rastering. Data collection and rastering is initiated by pressing the *PLOT* button. A typical bit raster display is shown in Figure 7–34. The number of bits collected is set by the *Data Samples* parameter. Up to 65535 bits can be collected for display by the bit raster. The *Line Width* parameter controls the raster width and *Start Bit* sets which bit is displayed first in the raster. The raster window can be closed by pressing the *Close Plot Window* button.

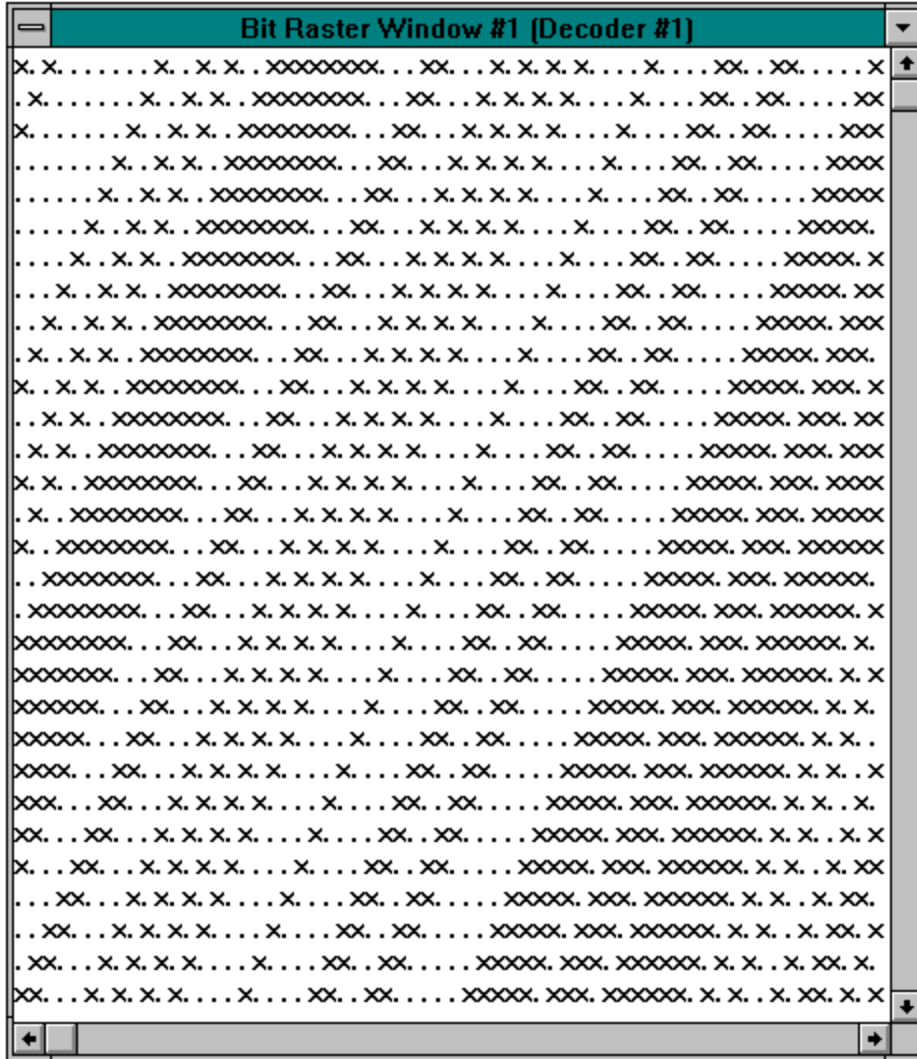


Figure 7-34 Bit Raster Display

8. Theory of Operation

The SIGTEK ST-106 Decoder/Derandomizer is designed to perform frame synchronization and forward error correction (FEC) decoding functions on serial data streams. The synchronization and decoding functions are performed by a combination of FEC decoding ASICs and XILINX field programmable gate arrays (FPGAs) interconnected via a high speed TDM bus. A block diagram of the SIGTEK ST-106 Decoder/Derandomizer is shown in Figure 8-1.

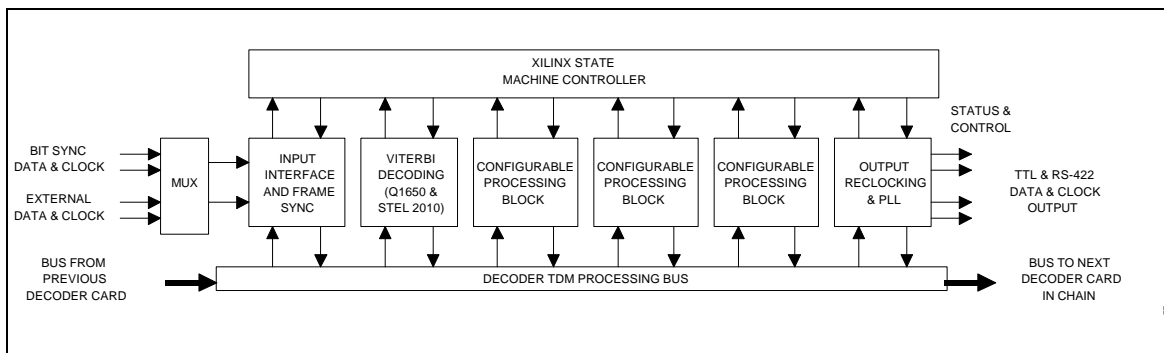


Figure 8-1 ST-106 Block Diagram

Data is input from either a TTL or RS-422 source and placed on the ST-106 tri-state, time division multiplex (TDM) bus so that any (or all) of the processing functions on the bus can access it. The TDM bus is divided into timeslots with each processing function on the bus assigned timeslots to read data from and write data to the bus. By simply changing the timeslots assignments, the order of the processing can be changed. The TDM bus can be chained between decoder boards to increase the overall processing capability.

Frame synchronization functions are performed by a XILINX field programmable gate array (FPGA) and a HARRIS correlator IC. The XILINX FPGA allows the frame sync circuits to be easily modified to suite new frame structures. The modified design files can then be downloaded via the system software. The correlator is configured for dual 128 bit x 1 operation and is used for detect occurrence of unique synchronization words for the frame synchronizers and the additive derandomizer.

Decoding of convolutional codes is performed by two Viterbi decoder ASICs: the Qualcomm Q1650 for constraint length 7 ($K=7$) codes and the STEL 2010 for constraint length 6 ($K=6$) codes. The Q1650 can decode signals with input symbol rates up to 2.1 Mbps while the STEL 2010 is limited to symbol rates of 512 kbps. Both decoders can perform either soft decision (3-bit quantization) or hard decision (1-bit quantization). The decoders ICs are connected the TDM bus via a XILINX FPGA.

Three of the processing blocks connected to the TDM bus consist of a XILINX FPGA and a 32k x 8 bit static ram. They can be configured for a variety of functions such as derandomizers, deinterleavers or deep FIFO memories for data collection. Normally, all three of these block are configured as FIFO memories for data collection. This permits data from three points in the processing chain to be stored to disk as discussed in Section 5.4. If one of these processing blocks is assigned to another function (such as derandomization) from the main menu, then one of the data collection resources will be lost and only two files can be open for data storage. The system software prevents the user from selecting output files when the processing resources are not available.

Data is read from the TDM bus and re-clocked to the output of the board using a phase locked loop circuit consisting of a XILINX FPGA, a Motorola dual VCO and a bank of selectable loop filters. The PLL synthesizes a symmetrical clock at the output data rate which typically is the input symbol rate times the FEC code rate. The XILINX FPGA implements the PLL division ratio counters and phase detector. The division ratios are set by the system software and are based on the input symbol rate and FEC code rate. A small 8 bit output FIFO is used in the XILINX chip to compensate for the mismatch in duty cycle between the clock used to read data from the TDM bus and the symmetrical output clock. This XILINX chip can also be used to implement a state machine controller for applications where complex sequencing is required to implement decoding functions.

9. Specifications

Table 9–1 lists specifications for the SIGTEK ST-106 Decoder/Derandomizer.

Table 9–1 ST-106 Decoder Specifications

<u>General</u>	
Size	Single Full Size PC-AT ISA Board
Power Requirements	+5V @ 1.0A, +/-12V @ 0.01A
<u>Input</u>	
Data Rate	2400 bps to 2.1 Mbps
Resolution	3-bit (soft decision) or 1-bit (hard decision)
Levels	TTL and RS-422
Connector Type	SMA (TTL), 10-pin IDC (RS-422)
Impedance	50 Ω (TTL), 100 Ω (RS-422)
<u>K=7 Convolutional Decoding</u>	
Data Rates	Up to 2.1 Mbps (input symbol rate)
Code Rates	1/3, 1/2, 3/4, 7/8
Polynomials	G0: 171, G1: 133, G2: 165 (octal)
<u>K=6 Convolutional Decoding</u>	
Data Rates	Up to 512 kbps (input symbol rate)
Code Rates	1/3, 1/2
Polynomials	Set 1: G0: 51, G1: 67, G2: 75 (octal) Set 2: G0: 73, G1: 61 (octal)
<u>Derandomization</u>	
Types	Feedthru and Additive
Taps	Programmable to 24 taps
Truncation	Up to 32767 bits (for additive type)
Initial Fill	Up to 24 bits (for additive type)
<u>Outputs</u>	
Signals	Decoded Data & Clock
Level	TTL and RS-422
Connector Type	SMA (TTL) 10-pin IDC and DB-15 (RS-422)