927 Design Guide 5/2/06

Description

The 927 is a dual spectroscopy ADC with histogramming memory. The communication path in the 927 is implemented with a Cypress FX2LP USB + microprocessor chip.

USB Interface

Pipes

The USB interface consists of 4 pipes as follows:

Pipe	Direction	Endpoint	Description
0	IN	1	Receives responses to microprocessor commands
1	OUT	1	Sends commands to microprocessor
2	IN	2	Receives responses to High-Speed Commands
3	OUT	8	Sends High-Speed Commands to FPGA in the 927

NOTE: IN = transfer from 927 to computer
OUT=transfer from computer to 927

Pipes 0 and 1 communicate directly with the 8051 microprocessor. This communication path is primarily used to configure the FPGA. Every command has a response, so when a command is sent on pipe 1, a response should be returned on pipe 0. The length of the response varies from command to command.

Pipes 2 and 3 implement a “High-Speed” communication path. Commands are sent on Pipe 3 and responses occur on Pipe 2. See High Speed USB Communication for more information. Attempted communication on pipes 2 and 3 will hang unless the FPGA has been configured.

High Speed USB Communication

The high speed communication port uses the slave fifo interface on the USB chip. The chip is configured by the firmware such that endpoint 8 is used to transfer commands from the PC to the unit, and data is returned on endpoint 2.

The commands are variable length, but it is vital that no more or less data is sent than is expected.

The commands are composed of 32-bit words. At least 2 words must always be sent. Write and Modify commands will have data words that follow the command.

The commands are formatted as follows:

Offset (DWORDS)	Name	Description
0	ADDRESS	Start address and OPCODE for the operation
1	LENGTH	Number of Operations to perform
2	DATA0	Optional Data Word
…	…	…
1+LENGTH	DATAN	Final Optional Data Word

The ADDRESS register is memory mapped as follows

ADDRESS		**0**
Bit Number	**Bit Mnemonic**	**Function**
31-24	OPCODE	Operation to be performed.
18	InternalRegisters	Requests that operation be performed on the internal control registers. Note only the low 8 address bits are decoded for internal control registers and the only valid opcodes are READ and WRITE. If this bit is clear the operation is performed on the Battery Backed-up RAM
17-0	StartAddress	Starting Address in DWORDS of Data to read/write/modify

The ONLY valid OPCODE values are as follows:

Value (H)	Name	Description
0	READ	Reads a block of Data
1	WRITE	Writes a block of data to the range of addresses – One Data word must be sent for every Value that is written.
2	WRITESINGLE	Writes a block of with the same data value – Only One Data word should be transmitted
3	READLIST	Reads list mode Data – not used
4	AND	Ands data with the current contents of memory (non-internal only). LENGTH Data words must be sent after the command.
5	ANDSINGLE	Ands data with the current contents of memory (non-internal only). The same data word is used for all locations. ONLY 1 data word should be transmitted
6	OR	ORs data with the current contents of memory (non-internal only). LENGTH Data words must be sent after the command.
7	ORSINGLE	ORs data with the current contents of memory (non-internal only). The same data mask is used for all locations. ONLY 1 data

		word should be transmitted

All commands will have a response, even the write commands. Commands that have no data to return (WRITE, WRITESINGLE, AND, ANDSINGLE, OR, ORSINGLE) return a "zero" length response. This response has no data payload, but it must be read to clear the transaction.

The READ command returns a block of data that consists of the number of DOUBLE WORDS requested with the length parameter. For example, if Length is 100, 100 DWORDS (or 400 bytes) are returned.

NOTE: When reading a response it is VERY important that the I/O request be for at least one byte more than what is actually expected! If this rule isn't followed, potential "hang" conditions exist when the response is a multiple of 512 bytes.

Battery Backed-up RAM Memory Map

This is the memory map for the Battery Backed-up RAM which is used for histogram and parameter storage

ADDRESS (Hex) (32-bit words)	Name	Description
0-3FFF	SPECTRUM1	Primary Spectrum for Input 1 (Normal/Error)
4000-7FFF	ZDTSPEC1	ZDT Spectrum for Input 1 (Unused/ZDT)
8000-BFFF	SPECTRUM2	Primary Spectrum for Input 2 (Normal/Error)
C000-FFFF	ZDTSPEC2	ZDT Spectrum for Input 2 (Unused/ZDT
10000-1FFFF	PARAMETER	Parameter memory (Settings and App Data)

The battery backed-up RAM is accessible via the High-Speed USB path or from the microcontroller.

Internal Control Register Memory Map

The internal control registers are accessible both from the uController and the High Speed USB interface. Although both sources can read and write all registers, by convention, the uController will only write registers marked with a * below. The High Speed Interface will ONLY write the registers that are not marked with a *. All registers may be read by either source.

The 927 consists of 2 ADCs. Two complete banks of internal control registers exist for each ADC. ADC 1 consumes addresses 0x0 – 0x1F. ADC 2 consumes address 0x20-0x3F. Add the address offset in the following table to 0 (for ADC 1) or 0x20 (for ADC2) to determine the complete address.

Name	**Address**	**Description**

	Offset(H)	
CTLREG	0	Control Register
AUXREG	1	Register to control Auxillary I/O
CONVGAIN	3	Conversion Gain
ZDTMODE	4	ZDT speed and setup
PRESETCTL	5	Preset control bits
LTPRESET	6	LiveTime preset
RTPRESET	7	RealTime preset
RPPRESET	8	Roi Peak preset
RIPRESET	9	Roi Integral preset
ACQSTATUS	A	Acquisition Status Register
LIVETIME	B	Livetime in 20ms units
REALTIME	C	Realtime in 20ms units – Value captured by reading LIVETIME
AUX0COUNTER	D	Auxillary counter – Counts number of edges on TTL Input 0
AUX1COUNTER	E	Auxillary counter – Counts number of edges on TTL Input 1
CTLREG	F	Mirror of CTLREG – Allows Block XFERS to stop, setup and restart
VERSION	11	Returns FPGA Version
STOPACQ*	1F	Stops acquisition from uC (for MDA Preset)

Detailed Internal Control Register Description

The following describes each control register in detail.

ACQSTATUS

The Register contains an assortment of bits that control loading of preset values and starting and stopping acquisitions

ACQSTATUS		
Bit Number	Bit Mnemonic	Function
0	Active	High when acquisition is active. Goes low on preset expiration. Acquisitions are activated by taking Start bit high, or enabling trigger followed by the application of a trigger signal.
1	TriggerArmed	High when EnableTrigger has been activated, but trigger hasn't occurred.
31-2		Unused

AUX0COUNTER
AUX1COUNTER

Each ADC in the 927 has 2 TTL inputs associated with it (See AUXREG, sample ready and Spare Input). Each of these inputs is connected to a 32-bit counter

that increments on a rising edge on the input. The counter only increments when acquisition is active. Writing to the register loads the counter with a preset value.

AUXREG

The Register contains bits that control the Auxiliary I/O on the 9-pin D connector on the rear panel.

AUXREG		
Bit Number	Bit Mnemonic	Function
0	Change Sample	Change Sample Output
1	SpareOut	Spare Output
2	SampleRdy	Sample Ready Input (Read only)
3	SpareIn	Spare Input (Read only)
31-4		Unused

CONVGAIN

Conversion gain of system.

CONVGAIN		
Bit Number	Bit Mnemonic	Function
31-5		Unused
4-0	ConvGain	Conversion gain of system. 0=16k 1=8k 2=4k 3=2k 4=1k 5=512 6=256

CTLREG

The Control Register contains an assortment of bits which control function in the FPGA. There is also a “Mirror” to ctlreg at the end of the block of registers. The mirror allows single block transfers to ensure the unit is stopped, download settings and then START.

The bit map is below:

CTLREG		
Bit Number	Bit Mnemonic	Function
31-30	-	Unused
29	GateCoin	Coincidence mode (1= Coincident Gate, 0=Anticoincident)
28	GateEnable	Enable Gate (1 = Enables Gate signal, 0 = Gate signal does

		nothing). This signal is only valid when EnableInh- is low.
27-8	-	Unused
7	TestSliderDAC	Puts Slider circuit into test mode (need backdoor test command)
6	DisableSlider	Disables Slider (Need backdoor test command for this)
5	DisableULD	Disables upper-level discriminator
4	EnableTrigger	Enables External Trigger
3-1	-	Unused
0	Start	Setting the signal high starts acquisition. Signal should remain high until acquisition is to be stopped.

LTPRESET

Livetime preset register. Preset is in 20ms ticks. Preset value should be loaded into this register just before acquisition starts.

LIVETIME

The LTREMAIN registers report the number of Live time ticks. The counter is in 20ms ticks. The register may be written to in order to establish an exiting Livetime in a spectrum.

PRESETCTL

Preset Control Register

PRESETCTL		
Bit Number	Bit Mnemonic	Function
7	Unused	
6	Unused	
5	ENABROIPEAK	Enables ROI peak preset
4	ENABROIINT	Enables ROI Integral Preset
3	ENABLT	Enables Livetime preset
2	ENABRT	Enables RealTime preset
1	ENABOF	Enables Overflow Preset
0	-	Unused

RTPRESET

Realtime preset register. Preset is in 20ms ticks. Preset value should be loaded into this register just before acquisition starts.

RPPRESET

ROI Peak preset register. Preset value should be loaded into this register just before acquisition starts.

RIPRESET

ROI Integral preset register. Preset value should be loaded into this register just before acquisition starts. If acquisition is just a restart of a previous acquisition, the register should be loaded with the number of ROI counts remaining.

REALTIME

The REALTIME register reports the number of real time ticks remaining. The real time should be "captured" by Reading the LIVETIME register. The counter is in 20ms ticks. The register may be written to in order to establish an exiting Realtime in a spectrum.

STOPACQ

Writing a 1 to this register will stop acquisition. This enables the microcontroller to stop acquisition without disturbing the CTLREG which is reserved for writes by the High Speed USB. This register should be immediately set back to 0 and must be cleared on powerup.

ZDTMODE

ZDT Setup

ZDTMode		
Bit Number	Bit Mnemonic	Function
5	ZDTMode	1=Error Spectrum 0=Normal
4	EnabZDT	1=Enable zero deadtime counting, 0=Normal
3-0	ZDTSpeed	Sets how often the ZDT value is updated. Legal Settings are: 0 = 137us 1 = 273us 3 = 546us 7 = 1092us 15=2184us

Microprocessor USB Communication (Low-Speed)

USB endpoint 1 is used to communicate directly with the microprocessor in the 927. This communication path is primarily used to configure the FPGA. The format of a command on this communication path is as follows:

The commands are formatted as follows:

Offset (bytes)	Name	Description
0	Command0	Low byte of command
1	Command1	High byte of command
2	Length0	Low byte of Response Length (bytes)
3	Length1	High byte of Response Length
4	DATA0	Optional Data
…	…	…
N+4	DATAN	Final Optional Data Word

The microprocessor will always return "Length" bytes even if the command normally would respond with fewer.

The 1st 2 bytes of the response are the error flags (Buf[0]=ErrMac, Buf[1]=ErrMic)

The remaining bytes in the response are the returned data.

Command Dictionary

Name	Value (Hex)	Description
CMD_INITFPGA	4	Initializes FPGA in preparation for a Configuration.
CMD_LOADFPGA	5	Sends a block of configuration data to the FPGA
CMD_TESTFPGA	6	Returns 1 if FPGA has been successfully configured, 0 if not
CMD_SETSDAC	7	Loads LLD/ULD DACs. The command takes 2 32-bit byte-swapped parameters and is formatted as follows: BUF[0] = CMD_SETSDAC (0x7) BUF[1] = 0 BUF[2] = response length LSB BUF[3] = response length MSB BUF[4] = 0 (32-bit MSB of Serial DAC ID) BUF[5] = 0 BUF[6] = 0 BUF[7] = 32-bit LSB of Serial DAC ID: ID=0: Input A ULD (14-bits) ID=1: Input A LLD (14-bits) ID=2: Input B ULD (14-bits) ID=3: Input B LLD (14-bits) BUF[8] = 0 (32-bit MSB DAC value) BUF[9] = 0 BUF[10] = Most Significant Byte of 14-bit DAC value BUF[11] = Least Significant Byte of 14-bit DAC value
CMD_DISAFIRM	8	Turns off the background firmware loop – this command should be sent just before INITFPGA if the FPGA is already initialized
CMD_ENABFIRM	9	Enables the background firmware loop – this command should be sent shortly after the FPGA has been configured.
CMD_TESTPRESETS	A	Tests the MDA and Uncertainty presets for completion, if they are enabled.
CMD_RESETMDA	10	Restarts MDA preset tests. Send before Start or after Clear.
CMD_RESETFPGA	11	Resets the FPGA. This command should be sent shortly after the FPGA is configured, but before ENABFIRM.
CMD_VERSION	12	Returns the firmware version (Big Endian)

Error Responses:

The 1st 2 bytes of all responses are the error flags. The 1st byte is the macro response. The 2nd byte is the micro response. Each macro code has several micro codes that help determine what went wrong. The micro error bits may be or'd together if more than one problem occurred. These are the standard ORTEC MCA error response codes that can be found in most any ORTEC MCA manual. The most common ones used in the microprocessor USB communication are

ErrMac=0 ErrMic=0	Success
ErrMac=0 ErrMic=2	A preset is already exceeded (CMD_TESTPRESETS)
ErrMac=4 ErrMic=128	FPGA failed/needs configuration (CMD_TESTFPGA)
ErrMac=129 ErrMic=132	Invalid Command

FPGA Configuration

The FPGA (Field Programmable Gate Array) inside the 927 is an SRAM device, so it loses its programming when the power is turned off. The program that is loaded into the FPGA resides in the computer, so the program must be loaded into the FPGA via the USB connection. As outlined in the previous section, there are several commands that are used to perform this configuration. The general flow is as follows:

Send CMD_DISAFIRM to turn off the firmware background loop

Send CMD_INITFPGA to initialize the FPGA

Open the RBF file and read a block of binary data. The driver requires that < 65536 bytes be sent in any command, so read the data in blocks less than this limit.

Send the block of data with CMD_LOADFPGA

Continue reading blocks of data and sending them over with the CMD_LOADFPGA command.

Once data is exhausted, Send the CMD_TESTFPGA command to determine if the FPGA has been successfully loaded. If not, start over.

Send the CMD_RESETFPGA command to prepare the FPGA for operation.

Send the CMD_ENABFIRM command to turn the background loop back on.

Firmware Programming Guide

The following sections define the memory maps, register definitions and techniques required to implement the firmware in the 927.

Data Space Memory Map

The microprocessor on the 927 has a 64k data space that is used to store variables and hold registers. The memory is mapped as follows:

ADDRESS (Hex) (Bytes)	Name	Description
0-7F	ISRVEC	Interrupt Service Routine Vectors
80-3FFF	INTERNAL	INTERNAL program/data space (Overlapped)
4000-402C	CONTROL	Control Registers
8000-DFFF	Ext. RAM	Off-chip RAM (not batt. backed) used for XDATA. NOTE that C000-CFFF is consumed by the debugger if it is used.
E000-FFFF	USB	USB Registers – Internal to EZUSB

Program Space Memory Map

The Internal microprocessor has a program space that is 64k bytes. The 927 only uses the low 16K which is internal to the microprocessor. Note that this memory overlaps the Data Space, so care must be taken to avoid mapping both data and program to the same location.

ADDRESS (Hex) (Bytes)	**Name**	**Description**
0-3FFF	INTERNAL	Internal Program/Data Space (Overlapped)

Access to Internal Control Registers from MicroController

The internal registers are accessible both from the uController and the High Speed USB interface. In order to multiplex the registers to the two locations with a minimum of conflict, indirect addressing of the registers is employed. The following registers are located in the uController XDATA memory space.

Name	**Address (H)**	**Description**
DATAREG0	4003	Low Byte of Output Data Register
DATAREG1	4002	2nd Byte of Output Data Register
DATAREG2	4001	3rd Byte of Output Data Register
DATAREG3	4000	High Byte of In Data Register
DATAINREG0A	4007	Low Byte of In Data Register for ADC A
DATAINREG1A	4006	2nd Byte of In Data Register
DATAINREG2A	4005	3rd Byte of In Data Register
DATAINREG3A	4004	High Byte of In Data Register
DATAINREG0B	400B	Low Byte of In Data Register for ADC B
DATAINREG1B	400A	2nd Byte of In Data Register
DATAINREG2B	4009	3rd Byte of In Data Register
DATAINREG3B	4008	High Byte of In Data Register
ADDRREG	400C	Address Register
STATREG	400D	Status Register –

		Bit 0: State Machine is BUSY. Don't write any registers if 1. Bit 1: MemoryDataRdy. High when a memory word read operation is complete Bit 2: MemoryFunctionBusy. High when memory processor is busy.

To write to an internal register, place the value to write in the 4 DATAREG bytes. Write the Address of the internal register (See Internal Register Memory Map) to ADDRREG.

To read an internal register, write the address of the internal register with bit 7 set high to ADDRREG. Check the STATREG, if the low bit is not set, the data can be read from DATAINREG.

Access to Data memory from MicroController

The Data Memory is accessible both from the uController and the High Speed USB interface. In order to multiplex the registers to the two locations with a minimum of conflict, indirect addressing of the memory is employed. The following registers, located in the uController XDATA memory space, are used to read, modify and write the memory.

Name	Address (H)	Description
MEMDATAWRITE3	4010	High Byte of Output Data Register
MEMDATAWRITE2	4011	3rd Byte of Output Data Register
MEMDATAWRITE1	4012	2nd Byte of Output Data Register
MEMDATAWRITE0	4013	Low Byte of Output Data Register
MEMADDR3	4014	High byte of address register (always 0)
MEMADDR2	4015	3rd Byte of Address Register
MEMADDR1	4016	2nd Byte of Address Register
MEMADDR0	4017	Low Byte of Address Register
MEMCOUNT1	4018	2nd Byte of Number of Channels
MEMCOUNT0	4019	Low Byte of Number of Channels
MEMFUNCTION	401A	Function to perform
READNEXT	401B	Writing to this register tells Block Read State machine that it is OK to read the next memory value
MEMDATAREAD3	401C	High Byte of In Data Register
MEMDATAREAD2	401D	3rd Byte of In Data Register
MEMDATAREAD1	401E	2nd Byte of In Data Register
MEMDATAREAD0	401F	Low Byte of In Data Register
INTEGRAL5	4020	High Byte of Integral Register
INTEGRAL4	4021	5th Byte of Integral Register
INTEGRAL3	4022	4th Byte of Integral Register

INTEGRAL2	4023	3rd Byte of Integral Register
INTEGRAL1	4024	2nd Byte of Integral Register
INTEGRAL0	4025	Low Byte of Integral Register
MAX3	4026	High Byte of Max Register
MAX2	4027	3rd Byte of Max Register
MAX1	4028	2nd Byte of Max Register
MAX0	4029	Low Byte of Max Register
MAXCHAN3	402C	4th Byte of Max Channel Register (always 0)
MAXCHAN2	402D	3rd Byte of Max Channel Register
MAXCHAN1	402E	2nd Byte of Max Channel Register
MAXCHAN0	402F	Low Byte of Max Channel Register

In addition the STATREG contains 2 bits that give the status of the memory processor.

Bit 1: MemoryDataRdy. High when a memory address has been read and the result is waiting in MEMDATAREAD.

Bit 2: MemoryFunctionBusy. High when memory processor is busy.

MEMDATAWRITE

Register loaded by uController with value used to Write or Modify memory.

MEMDATAREAD

Result of last read operation performed by the uController Memory Access Function.

MEMADDR

Starting address used by uController Memory Access Function. (32-bit Address)

MEMCOUNT

Number of operations uController Memory Access Function is to perform.

MEMFUNCTION

Control register for uController Memory Access Function. When this register is written with a value, it initiates a Memory Access operation. The functions are defined as follows:

Value (H)	Name	Description
0	RESET	Resets the system
1	READ	Reads a block of Data. After each word is read, it is stored in MEMDATAREAD and MemoryDataRdy is set. Writing anything to READNEXT clears the MemoryDataRdy flag and initiates another read operation.
2	WRITE	Fills a block of memory with the value in MEMDATAWRITE. MemoryFunctionBusy goes low when the operation completes.
3	OR	Ors MEMDATAWRITE with the current contents of

		memory. MemoryFunctionBusy goes low when the operation completes.
4	AND	Ands MEMDATAWRITE with the current contents of memory. MemoryFunctionBusy goes low when the operation completes.
5	EVAL	Computes the Integral and finds the MAX and MAXCHAN. Note if High bit of MEMDATAWRITE is set, only data with the ROI bit set gets evaluated. MemoryFunctionBusy goes low when the operation completes.

READNEXT

Writing anything to this address initiates the read of the next memory address in a block read function.

INTEGRAL

INTEGRAL holds the sum of the specified range of channels. If the MSB of MEMDATAWRITE is set, then the sum only consists of channels with the ROI bit set.

MAX

Returns maximum value found during EVAL operation. If the MSB of MEMDATAWRITE is set, then only channels with the ROI bit set are evaluated.

MAXCHAN

Returns Channel that holds the maximum value found during an EVAL Operation. If the MSB of MEMDATAWRITE is set, then only channels with the ROI bit set are evaluated.