

Digital X-ray Processor User's Manual

\$Rev: 12017 \$

Model DXP-XMAP

Revision: D

With
xManager Software
version 1.0.x

XIA, LLC

31057 Genstar Road
Hayward, CA 94544 USA
Tel: (510) 401-5760; Fax: (510) 401-5761
<http://www.xia.com/>

Information furnished by XIA LLC is believed to be accurate and reliable. However, no responsibility is assumed by XIA LLC for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of XIA LLC. XIA LLC reserves the right to change specifications at any time without notice. Patents have been applied for to cover various aspects of the design of the DXP Digital X-ray Processor (DXP).

Safety	vi
Specific Precautions	vi
Do Not Hot-Swap!	vi
Servicing and Cleaning	vi
Manual Conventions	vii
End Users Agreement	viii
Contact Information:	viii
1 Introduction.....	1
1.1 XMAP Features.....	1
1.2 Data Acquisition Modes	2
1.2.1 MCA Mode	2
1.2.1.1 SCA Feature in MCA Mode	2
1.2.2 MCA Mapping Mode	2
1.2.3 SCA Mapping Mode.....	3
1.3 System Requirements:.....	4
1.3.1 Host Computer	4
1.3.2 PXI/CompactPCI Crate	4
1.3.3 Detector/Preamplifier	4
1.3.4 Power Supplies	5
1.3.5 Cabling.....	5
1.3.5.1 Analog Inputs	5
1.3.5.2 TTL/CMOS Logic Inputs	5
1.4 Software and Firmware Overview.....	5
1.4.1 User Interface: xManager	5
1.4.2 Device Driver: Handel.....	6
1.4.3 Firmware and FDD Files.....	6
1.4.4 Initialization File	6
1.5 Support	6
1.5.1 Software and Firmware Updates	7
1.5.2 Related Documentation	7
1.5.3 Technical Support	7
1.5.3.1 Submitting a problem report:.....	8
1.5.4 Feedback	8
1.5.4.1 Export File Formats.....	8
1.5.4.2 Calibration.....	8
2 Installation.....	10
2.1 PXI Interface Installation	10
2.1.1 PXI System Configuration File: PXISYS.INI	10
2.1.2 PXI Bus Segments and Slots.....	10
2.2 Software Installation.....	10
2.2.1 Running the Installer	11
2.2.2 File Locations.....	11
2.2.3 Support	11
2.3 Configuring the Analog Signal Conditioner.....	12
2.3.1 Input Attenuation: JP100, JP200, JP300, JP400.....	12
2.4 Installing DXP-XMAP Cards	12
2.5 Making Connections	13
2.5.1 Signal Connections	13
2.5.2 GATE/SYNC/LBUS Connection	13
2.6 Starting the System.....	13
2.6.1 DXP-XMAP Driver Selection.....	14

3	System Configuration	15
3.1	Initialization Files.....	15
3.1.1	Starting xManager Without an INI File.....	15
3.2	The Configuration Wizard	16
3.2.1	General Settings	16
3.2.2	Hardware Synchronization Settings.....	19
3.2.3	Mapping Mode Settings	21
3.2.4	Completing the Configuration	22
3.3	Loading and Saving Initialization Files.....	23
3.3.1	Loading an INI file	23
3.3.2	Saving an INI file.....	23
4	Using xManager with the DXP-XMAP	24
4.1	A Quick Tour of xManager	24
4.1.1	Channel Selection.....	24
4.1.2	Settings Sidebar.....	25
4.1.3	Main Window	25
4.2	Detector and Preamplifier Settings	25
4.2.1	Pre-Amplifier Polarity	26
4.2.2	Reset Delay	27
4.2.3	Preamp Gain.....	27
4.2.4	Preamp Risetime	27
4.3	Normal Spectrum Mode Data Acquisition.....	27
4.3.1	Starting a Run	27
4.3.2	Skipping Channels	28
4.3.3	Spectrometer Settings	29
4.3.3.1	Peaking Time (Energy Filter)	29
4.3.3.2	Trigger Threshold.....	30
4.3.3.3	Baseline Threshold	30
4.3.3.4	Energy Threshold.....	30
4.3.3.5	Dynamic Range	31
4.3.3.6	MCA Number of Bins and MCA Bin Width.....	31
4.3.3.7	Baseline Average.....	31
4.3.4	Setting Regions of Interest (ROIs).....	32
4.3.4.1	Adding ROIs.....	32
4.3.4.2	Auto ROI	32
4.3.5	Single Channel Gain Calibration.....	33
4.3.5.1	Calibrating the Gain	33
4.3.5.2	Propagating the Nominal Gain.....	34
4.3.6	System-Wide Gain Calibration.....	35
4.3.6.1	Skipping Channels.....	35
4.3.6.2	Running the Calibration Macro	35
4.3.7	Saving and Loading INI Files.....	37
4.3.7.1	Saving an INI File.....	37
4.3.7.2	Loading an INI File.....	37
4.3.7.3	Creating an INI File	37
4.3.8	Output Statistics.....	37
4.3.8.1	Real Time.....	37
4.3.8.2	Trigger Live Time	37
4.3.8.3	Energy Live Time	37
4.3.8.4	Input Count Rate (ICR)	38
4.3.8.5	Output Count Rate (OCR).....	38
4.3.8.6	Dead Time %	38
4.3.8.7	ROI Statistics	38
4.3.9	Single Channel Analyzer (SCA).....	38

4.3.9.1	Creating SCAs	38
4.3.9.2	Running with SCAs	39
4.3.10	Saving and Loading Data	39
4.3.10.1	MCA Data.....	39
4.3.10.2	Baseline Histogram Data	39
4.3.10.3	Trace Data	40
4.3.10.4	DSP Parameters	40
4.4	Run Control.....	40
4.4.1	Run Presets (Automatic Run Termination).....	40
4.4.1.1	Run To Preset [choice1] Preset Value = choice2	41
4.4.2	LBUS (Run Start Synchronization)	41
4.4.3	The GATE Function	41
4.4.4	Resume Run: Clear or Retain MCA Data	41
4.5	Display Controls.....	42
4.5.1	MCA Auto Update / Refresh Rate.....	42
4.5.2	Graphical Display Tools.....	42
4.5.2.1	Basic Tools and Options	42
4.5.2.2	Cursor Tools and Options	43
4.5.2.3	Axis Tools and Options	43
4.6	Optimizations	44
4.6.1	Throughput (OCR)	44
4.6.1.1	Peaking Time (Energy Filter)	44
4.6.1.2	Gap Time (Energy Filter)	45
4.6.2	Pileup Rejection	47
4.6.2.1	Maximum Width Constraint.....	47
4.6.2.2	Peak Interval	47
4.6.2.3	Reducing the Fast Peaking Time.....	48
4.6.3	Energy Resolution.....	48
4.6.3.1	Proper Peaking Time Selection	48
4.6.3.2	Baseline Acquisition.....	48
4.6.3.3	Eliminate Noise Pickup	48
4.6.3.4	Sufficient Gain to Sample Noise	49
4.6.3.5	Sufficient Gap Time	49
4.6.3.6	Peak Sampling Time.....	49
4.7	Diagnostics	49
4.7.1	The Traces Panel (Oscilloscope).....	49
4.7.1.1	Determining the Preamplifier Polarity and Gain.....	50
4.7.1.2	Measuring the Preamplifier Risetime	52
4.7.1.3	Measuring the RC Decay Time τ (RC-Feedback Preamplifiers only)	53
4.7.1.4	Optimizing the Baseline Average Length.....	53
4.7.2	The Baseline Panel.....	56
4.7.2.1	The Baseline Threshold.....	57
4.7.3	DSP Parameters.....	58
4.7.3.1	Generating a Diagnostic DSP Parameters File	59
4.7.3.2	Modifying DSP Parameters.....	59
4.7.4	Submitting a problem report:	59
4.7.4.1	Generating a Full Error Report	60
4.7.4.2	Saving MCA Data	60
4.7.4.3	Saving Baseline Data.....	60
4.7.4.4	Saving Trace Data	60
5	Mapping Mode	61
5.1	Pixel Advance Settings	61
5.1.1	LBUS PCI Extension (Run Start Synchronization)	61

5.1.2	Pixel Advance on GATE Edge.....	61
5.1.2.1	GATE Polarity	62
5.1.2.2	GATE Ignore Setting.....	62
5.1.3	Pixel Advance using SYNC Clock	63
5.1.4	Pixel Advance under Host Control.....	64
5.2	Mapping Mode Data Acquisition	65
5.2.1	The Mapping Panel.....	65
5.2.2	Mapping Mode: MCA or SCA	65
5.2.3	Total Number of Pixels	65
5.2.4	Buffer Control.....	66
5.2.5	Mapping Mode Data Acquisition	66
5.3	Mapping Mode Data.....	66
5.3.1	Mapping Data Options	67
5.3.2	Mapping Data Format	67
5.3.3	Single Buffer Format.....	67
5.3.3.1	Statistics Units	68
5.3.3.2	Buffer Header.....	68
5.3.3.3	Mapping Mode 1: Full Spectrum Mapping.....	69
5.3.3.4	Mapping Mode 2: Multiple SCA Mapping	71
5.3.3.5	Mapping Mode 3: List Mode Mapping.....	73
6	Digital Filtering: Theory of Operation and Implementation Methods	75
6.1	X-ray Detection and Preamplifier Operation:.....	75
6.1.1	Reset-Type Preamplifiers	75
6.1.2	RC-Type Preamplifiers.....	76
6.2	X-ray Energy Measurement & Noise Filtering:	77
6.2.1	Digital Filtering Theory	77
6.2.2	Trapezoidal Filtering	79
6.3	Trapezoidal Filtering in the DXP:	80
6.3.1	Comparing DXP Performance	80
6.3.2	Decimation and Peaking Time Ranges	80
6.3.3	Time Domain Benefits of Trapezoids.....	81
6.4	Baseline Issues:.....	82
6.4.1	The Need for Baseline Averaging.....	82
6.4.2	Raw Baseline Measurement.....	84
6.4.3	Baseline Average Settings and Recommendations	84
6.4.4	Why Use a Finite Averaging Length?	85
6.5	X-ray Detection & Threshold Setting:	85
6.6	Peak Capture Methods	86
6.6.1	Setting the Gap Length.....	87
6.6.2	Peak Sampling vs. Peak Finding.....	87
6.7	Energy Measurement with Resistive Feedback Preamplifiers	89
6.8	Pile-up Inspection:	92
6.9	Input Count Rate (ICR) and Output Count Rate (OCR):	94
6.10	Throughput:	95
6.11	Dead Time Corrections:	97
7	DXP-XMAP Hardware Description	98
7.1	DXP-XMAP Overview	98
7.1.1	Four Independent DXP Channels.....	98
7.1.2	Rapid Data Readout	99
7.2	Timing and Synchronization Logic.....	99
7.2.1	Connections Overview.....	99
7.2.2	LBUS (Run Start) Synchronization	100

7.2.3	GATE Function: MCA Mode	101
7.2.3.1	GATE Polarity	101
7.2.3.2	Ignore GATE	102
7.2.3.3	GATE Real Time.....	102
7.2.4	GATE Function: Mapping Mode	102
7.2.4.1	Pixel Advance on GATE Edge.....	102
7.2.4.2	GATE Polarity	102
7.2.4.3	GATE Ignore Setting.....	103
7.2.5	SYNC Function: Mapping Mode	103
7.2.5.1	Pixel Advance using SYNC Clock	104
7.2.5.2	Synchronous Starts with SYNC	104
7.3	The Analog Signal Conditioner (ASC):	105
7.4	Analog to Digital Converter	106
7.5	The Filter, Pulse Detector, & Pile-up Inspector (FiPPI):	107
7.5.1	FiPPI Configuration.....	107
7.5.2	FiPPI Version and Variants.....	107
7.5.3	FiPPI Decimation	107
7.5.4	Digital Trapezoidal Filtering	108
7.5.4.1	Noise and Pileup.....	108
7.5.4.2	Fast (Trigger) Filter	108
7.5.4.3	Slow (Energy) Filter	108
7.5.4.4	Intermediate (Baseline) Filter.....	109
7.5.5	Statistics.....	109
7.6	The Digital Signal Processor (DSP):.....	109
7.6.1	Event Processing	110
7.6.2	Statistics.....	110
7.7	System FPGA	110
7.7.1	Basic 32-bit MCA Data Acquisition	111
7.7.2	Full Spectrum 16-bit MCA Mapping/Scanning Mode.....	111
7.7.3	Other Data Acquisition Modes.....	Error! Bookmark not defined.
7.8	3 rd Party PCI Interface	112
Appendices.....		113
Appendix A. DPP-XMAP Revision C Circuit Board Description.....		113
A.1.	Jumper Settings.....	114
A.2.	LED Indicators	114
A.3.	Connectors	115
A.4.	Power Consumption:	115

Safety

Please take a moment to review these safety precautions. They are provided both for your protection and to prevent damage to the digital x-ray processor (DXP) and connected equipment. This safety information applies to all operators and service personnel.

Specific Precautions

Observe all of these precautions to ensure your personal safety and to prevent damage to either the DXP-XMAP or equipment connected to it.

Do Not Hot-Swap!

To avoid personal injury, and/or damage to the DXP-XMAP, always turn off crate power before removing the DXP-XMAP from the crate!

Servicing and Cleaning

To avoid personal injury, and/or damage to the DXP-XMAP, do not attempt to repair or clean the unit. The DXP hardware is warranted against all defects for 1 year. Please contact the factory or your distributor before returning items for service. To avoid personal injury, and/or damage to the DXP-XMAP, do not attempt to repair or clean the unit.

Manual Conventions

Through out this manual we will use the following conventions:

Convention	Description	Example
»	The » symbol leads you through nested menu items and dialog box options.	The sequence File»Page Setup»Options directs you to pull down the File menu, select the Page Setup item, and choose Options from the sub menu.
Bold	Bold text denotes items that you must select or click on in the software, such as menu items, and dialog box options.	...click on the MCA tab.
[Bold]	Bold text within [] denotes a command button.	[Start Run] indicates the command button labeled Start Run.
monospace	Items in this font denote text or characters that you enter from the keyboard, sections of code, file contents, and syntax examples.	Setup.exe refers to a file called "setup.exe" on the host computer.
"window"	Text in quotation refers to window titles, and quotations from other sources	"Options" indicates the window accessed via Tools»Options .
<i>Italics</i>	Italic text denotes a new term being introduced, or simply emphasis	<i>peaking time</i> refers to the length of the slow filter. ...it is important first to set the energy filter Gap so that SLOWGAP to <i>at least one unit greater than the preamplifier risetime</i> ...
<Key> <Shift-Alt-Delete> or <Ctrl+D>	Angle brackets denote a key on the keyboard (not case sensitive). A hyphen or plus between two or more key names denotes that the keys should be pressed simultaneously (not case sensitive).	<W> indicates the W key <Ctrl+W> represents holding the control key while pressing the W key on the keyboard
<i>Bold italic</i>	Warnings and cautionary text.	<i>CAUTION: Improper connections or settings can result in damage to system components.</i>
CAPITALS	CAPITALS denote DSP parameter names	SLOWLEN is the length of the slow energy filter

End Users Agreement

XIA LLC warrants that this product will be free from defects in materials and workmanship for a period of one (1) year from the date of shipment. If any such product proves defective during this warranty period, XIA LLC, at its option, will either repair the defective products without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, Customer must notify XIA LLC of the defect before the expiration of the warranty period and make suitable arrangements for the performance of the service.

This warranty shall not apply to any defect, failure or damage caused by improper uses or inadequate care. XIA LLC shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than XIA LLC representatives to repair or service the product; or b) to repair damage resulting from improper use or connection to incompatible equipment.

THIS WARRANTY IS GIVEN BY XIA LLC WITH RESPECT TO THIS PRODUCT IN LIEU OF ANY OTHER WARRANTIES, EXPRESSED OR IMPLIED. XIA LLC AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. XIA'S RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. XIA LLC AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER XIA LLC OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Contact Information:

XIA LLC
31057 Genstar Rd.
Hayward, CA 94544 USA

Telephone:	(510) 401-5760
Downloads:	http://xia.com/DXP-XMAP_Download.html
Hardware Support:	support@xia.com
Software Support:	software_support@xia.com

1 Introduction

The DXP-XMAP includes four Digital X-ray Processor (DXP) channels on a single 3U PXI/CompactPCI card. Each DXP channel is a high rate, digitally-based, multi-channel analysis spectrometer designed for energy dispersive x-ray or γ -ray measurements. Amplifier and spectrometer controls including gain, filter peaking time, and pileup inspection criteria are under computer control. Given the very high data transfer rates of the PCI bus, the DXP-XMAP is uniquely suited for high-speed x-ray timing and scanning applications.

1.1 XMAP Features

- Single PXI/CompactPCI module contains 4 channels of pulse processing electronics with full MCA per channel.
- 4 MB of high-speed memory allows ample storage for timing applications such as mapping with full spectra or multiple ROI's. Memory can be read at the full PCI speed.
- Peak PCI transfer rates exceed 100 MB/sec.
- Peaking time range: 0.1 to 164 microseconds.
- Maximum throughput up to 1,000,000 counts/sec/channel.
- Digitization: 14 bits at 50 MHz
- Low noise front end offers excellent resolution, and provides excellent performance in the soft x-ray region (150 - 1500 eV).
- Operates with virtually any x-ray detector. Preamplifier type is computer controlled.
- 16 bit gain DAC and input offset are computer controlled.
- Pileup inspection criteria are computer selectable.
- Accurate ICR and livetime for precise deadtime correction and count rate linearity.
- Multi-channel analysis for each channel allows optimal use of data.
- Facilitates automated gain setting and calibration to simplify tuning array detectors.
- External GATE input allows data acquisition on all channels to be synchronized.
- All runs can be synchronized between modules using the LBUS signal connecting all the modules together.
- Single spectrum mode provides for the acquisition of a single spectrum per DXP channel.
- Mapping modes offers time-resolved data acquisition, i.e. one spectrum or set of SCA windows per pixel or scan point.

1.2 Data Acquisition Modes

The DXP-XMAP currently supports two data acquisition modes: static single-spectrum 'Normal' acquisition and time-resolved multi-spectrum 'Mapping' acquisition. Note: The two data acquisition modes use different memory architectures and thus require different firmware code to be downloaded.

1.2.1 MCA Mode

In Multi-Channel Analyzer (MCA) mode a data acquisition run produces a single energy spectrum and associated run statistics for each DXP processing channel. Data acquisition runs can be started and stopped manually, or can be stopped automatically according to a preset real time, live time or number of input or output events.

Spectrum size ranges from 256 bins to 16384 bins. Each spectral bin is stored as a 32-bit value, allowing for up to 4,294,967,295 events per bin per run. Data is stored in on-board memory, and can be read by the host at any time during or after the run. The memory is normally cleared at the beginning of a run, but can instead be preserved, allowing for 'pause and resume' functionality. Data acquisition can be halted system-wide according to a user provided TTL/CMOS GATE signal, e.g. to achieve a synchronous run start.

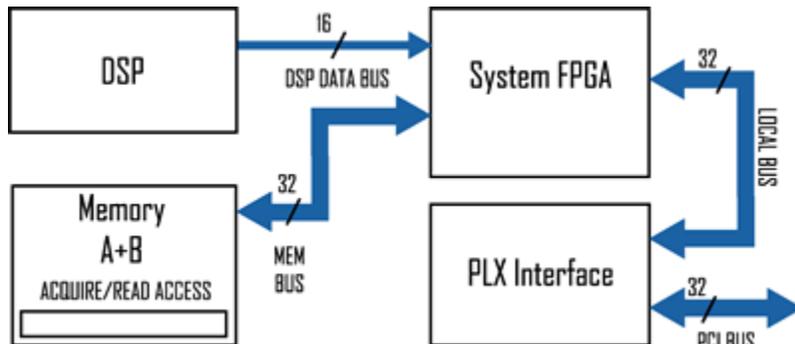


Figure 1.1: Data flow diagram for MCA mode.

1.2.1.1 SCA Feature in MCA Mode

The Single-Channel Analyzer (SCA) feature allows for up to 32 regions of the spectrum (SCA windows) to be defined and for which output counts are individually summed. The sums are organized into a table stored in memory, in addition to the MCA data and statistics. The SCA table can be accessed directly for fast readout of critical data.

1.2.2 MCA Mapping Mode

This mode supports x-ray scanning applications where multiple spectra are generated as an x-ray beam is scanned across a sample; each spectrum corresponds to a scan point, or pixel. This mode also supports XAFS spectroscopy, where each spectrum corresponds to the beam energy, or monochromator setting.

A data acquisition run produces multiple energy spectra, each with associated run statistics, for each DXP processing channel. Typically a user-provided TTL/CMOS timing signal is used to advance from one spectrum to the next during the run. Data acquisition runs can be started and stopped manually, or can be stopped automatically according to a preset number of spectra.

Spectrum size ranges from 256 bins to 16384 bins. Each spectral bin is stored as a 16-bit value, allowing for up to 65,535 events per bin. On-board memory is configured as two devices, memory A and memory B, each accessible to *either* the host *or* the on-board DSP. Continuous operation is achieved by reading memory A while the DSP writes memory B, and vice-versa. The data readout speed, spectrum size and total number of system processing channels place a limit on the minimum pixel, or dwell, time.

The external logic (LEMO) input can be configured to control the pixel advance function, which creates a new spectrum corresponding to a new pixel.

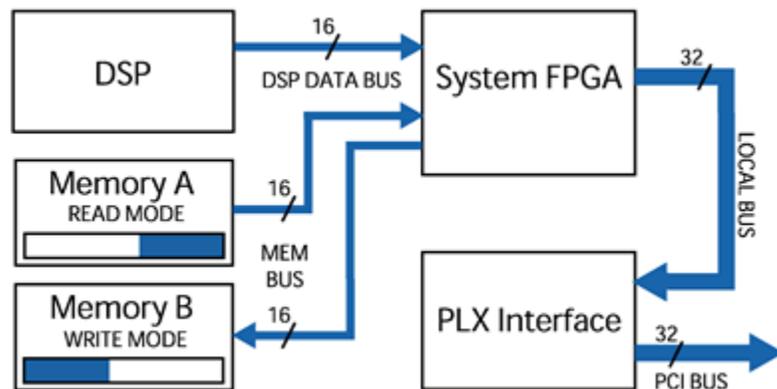


Figure 1.2: Data flow diagram for multiple spectrum mode.

1.2.3 SCA Mapping Mode

The Single-Channel Analyzer (SCA) mapping mode allows for up to 64 regions of the spectrum (SCA windows) to be defined and for which output counts are individually summed. Instead of entire spectra, only the tables of SCA sums are stored in memory. Compressing the data in this way allows for faster readout times, or, conversely shorter dwell times.

1.3 System Requirements:

The digital spectroscopy system considered here consists of a remote host computer, a PXI/CompactPCI crate, an optical link from host to crate, one or more DXP-XMAP cards, and an x-ray detector/preamplifier with appropriate power supplies. An alternative solution is to replace the remote computer and PXI controller with a native PXI/CompactPCI computer.

*Windows XP/2000
CompactPCI computer or a
desktop fitted with a
CompactPCI optical
interface card.*

1.3.1 Host Computer

The DXP-XMAP communicates with a host computer via the CompactPCI bus. The CompactPCI interface is described further in section 2.1. The host computer that runs XIA's Handel and/or xManager software must have the following minimum capabilities:

- ✓ 300 MHz or greater processor speed running most Microsoft Windows Operating systems (2000, XP).
- ✓ Native PCI computer, or desktop/laptop equipped with PCI interface card installed, e.g. National Instruments MXI-3 or MXI-4.

1.3.2 PXI/CompactPCI Crate

A wide variety of PXI Crates are available. We strongly recommend using standard crates from trusted vendors, e.g. National Instruments. Such vendors provide software that automatically generates the PXI System Configuration File: "C:\Windows\System\pxisys.ini". This file ensures proper communication with instruments in the crate.

- ✓ Make sure the crate provides a sufficient number of slots to accommodate your system.
- ✓ Native PCI computer or PCI interface card installed, e.g. National Instruments MXI-3 or MXI-4.

*Preamplifier signal
specifications must be
verified.*

1.3.3 Detector/Preamplifier

The DXP-XMAP accommodates nearly all preamplifier signals. The two primary capacitor-discharge topologies, pulsed-reset and resistive-feedback, are both supported. The input voltage range of the DXP analog circuitry results in the following constraints:

<i>Parameter</i>	<i>Minimum</i>	<i>Maximum</i>	<i>Typical</i>
X-ray pulse-height (w/ input attenuator)	250 μ V (1 mV)	375 mV (1.50 V)	25 mV -
Input voltage range (w/ input attenuator)	- -	+/-5 V (+/-20 V)	+/-3 V -

Table 1.1: Analog input signal constraints for pulsed-reset preamplifiers.

<i>Parameter</i>	<i>Minimum</i>	<i>Maximum</i>	<i>Typical</i>
X-ray pulse-height (w/ input attenuator)	250 μ V (1 mV)	625 mV (2.50 V)	100 mV -
Input voltage range (w/ input attenuator)	- -	+/-5 V (+/-20 V)	+/-3 V -
Decay time τ	100 ns	infinity	50 μ s

Table 1.2: Analog input signal constraints for resistive-feedback preamplifiers.

1.3.4 Power Supplies

If possible, we recommend using local power to generate DC voltages for the preamplifier and HV bias voltage for the detector.

- ✓ The XPPS, manufactured by XIA, provides linear power for up to 20 NIM-standard preamplifiers.
- ✓ The EHQxx line of CompactPCI HV bias supply modules, manufacture by ISEG, is available for resale from XIA.

If you decide to use your own supplies, expect to spend some time experimenting with ground connections. Switching currents in the CompactPCI crate tend to generate voltage spikes between local and external grounds that can show up in the signal path.

1.3.5 Cabling

1.3.5.1 Analog Inputs

The DXP-XMAP uses high-quality SMA connectors to accept the preamplifier signals. XIA provides a BNC-to-SMA converter cable harness for every input (4 per XMAP module).

1.3.5.2 TTL/CMOS Logic Inputs

The DXP-XMAP uses a LEMO connector for timing and synchronization logic. Depending on the application LEMO cables and 'T' adapters may be required. This hardware is available from a wide variety of vendors. Please contact XIA if you need assistance with LEMO hardware.

1.4 Software and Firmware Overview

Two levels of software are employed to operate the DXP-XMAP: a user interface for data acquisition and control, and a driver layer that communicates between the host software and the PCI bus. In addition, separate firmware code is downloaded to and runs on the DXP-XMAP itself.

1.4.1 User Interface: xManager

The user interface communicates with and directs the DXP-XMAP via the driver layer, and displays and analyzes data as it is received. As such XIA provides xManager as a general-purpose data acquisition application. xManager features full control over the DXP-XMAP, intuitive data visualization, unlimited ROI's (regions of interest) Gaussian fitting algorithms and the exporting of collected spectra for additional analysis. Please refer to Chapter 3 of this

manual for instructions on using xManager with the DXP-XMAP. Many users will employ xManager for configuration and system optimization, but will want to develop their own software to acquire data.

1.4.2 Device Driver: Handel

XIA provides source code and documentation for the Handel driver layer to advanced users who wish to develop their own software interface. XIA recommends using Handel for almost all advanced applications. Handel is a high-level device driver that provides an interface to the DXP hardware in spectroscopic units (eV, microseconds, etc...) while still allowing for safe, direct-access to the DSP. xManager uses the Handel driver, and thus also serves as a development example. Installation files and user manuals for Handel are available online at http://www.xia.com/DXP_Software.html.

1.4.3 Firmware and FDD Files

Firmware refers to the DSP (digital signal processor) and FPGA (Field Programmable Gate Array) configuration code that is downloaded to the DXP-XMAP itself. Typically two System FPGA files (one each for normal and mapping acquisition modes), one DSP file and up to four FiPPI (Filter-Pulse-Pileup-Inspector FPGA) files are necessary to acquire spectra across the full range of peaking times with a given detector/preamplifier. For simplicity XIA provides complete firmware sets in files of the form "firmware_name.fdd". This file format is supported by Handel, XIA's digital spectrometer device driver, and is the standard firmware format used in xManager. Two standard firmware files are available, one for pulsed-reset type preamplifiers and one for RC-feedback type preamplifiers. Updates to the firmware are available online at:

www.xia.com/DXP_Resources.html.

The System FGPA, DSP and FiPPI are discussed in Chapter 7. Firmware file formats are further described in **Error! Reference source not found.**

1.4.4 Initialization File

Handel (and thus xManager) uses an initialization (INI) file to store all necessary configuration information, including the path and filename of the firmware file on the host computer, number and slot location of xMAP modules in the system, detector characteristics and spectrometer settings, and timing and synchronization logic functions used.

1.5 Support

A unique benefit of dealing with a small company like XIA is that the technical support for our sophisticated instruments is often provided by the same people who designed them. Our customers are thus able to get in-depth technical advice on how to fully utilize our products within the context of their particular applications. Please read through this brief chapter before contacting us.

XIA LLC
31057 Genstar Rd.
Hayward, CA 94544 USA

Telephone: (510) 401-5760
Downloads: http://xia.com/DXP-XMAP_Download.html
Hardware Support: support@xia.com
Software Support: software_support@xia.com

*Check for firmware and software updates at:
http://www.xia.com/DXP_Resources.html*

1.5.1 Software and Firmware Updates

It is important that your DXP unit is using the most recent software/firmware combination, since most problems are actually solved at the software level. Please check http://xia.com/DXP-XMAP_Download.html for the most up to date standard versions of the DXP software and firmware. Please contact XIA at support@xia.com if you are running semi-custom or proprietary firmware code. (*Note:* It a good practice to make backup copies of your existing software and firmware before you update).

1.5.2 Related Documentation

As a first step in diagnosing a problem, it is helpful to consult most recent data sheets and user manuals for a given DXP product, available in PDF format from the XIA web site. Since these documents may have been updated since the DXP unit has been purchased, they may contain information that may actually help solving your particular problem. All manuals, datasheets, and application notes, as well as software and firmware downloads can be found at http://xia.com/DXP-XMAP_Download.html. In order to request printed copies, please send an e-mail to support@xia.com, or call the company directly. In particular, we recommend that you download the following user manuals:

- ✓ xManager User Manual – All users
- ✓ Handel User Manual – Users who wish to develop their own user interface

1.5.3 Technical Support

The DXP-XMAP comes with one year of e-mail and phone support. Support can be renewed for a nominal fee. Please call XIA if your support agreement has expired.

The XIA Digital Processors (DGF & DXP) are digitally controlled, high performance products for X-ray and gamma-ray spectroscopy. All settings can be changed under computer control, including gains, peaking times, pileup inspection criteria, and ADC conversion gain. The hardware itself is very reliable. Most problems are not related to hardware failures, but rather to setup procedures and to parameter settings. XIA's DXP software includes several consistency checks to help select the best parameter values. However, due to the large number of possible combinations, the user may occasionally request parameter values which conflict among themselves. This can cause the DXP unit to report data which apparently make no sense (such as bad peak resolution or

even empty spectra). Each time a problem is reported to us, we diagnose it and include necessary modifications in the new versions of our DXP control programs, as well as adding the problem description to the FAQ list on our web site.

1.5.3.1 Submitting a problem report:

XIA encourages customers to report any problems encountered using any of our software via email. In most cases, the XIA engineering team will need to review bug information and run tests on local hardware before being able to respond.

All software-related bug reports should be e-mailed to software_support@xia.com and should contain the following information, which will be used by our technical support personnel to diagnose and solve the problem:

- ✓ Your name and organization
- ✓ Brief description of the application (type of detector, relevant experimental conditions...etc.)
- ✓ XIA hardware name and serial number
- ✓ Version of the library (if applicable)
- ✓ OS
- ✓ Description of the problem; steps taken to re-create the bug
- ✓ Full Error Report (see section 4.7.4.1) plus additional data:
 - Saved MCA data, if relevant (see section 4.7.4.2)
 - Saved Baseline data, if relevant (see section 4.7.4.3)
 - Saved Trace data, if relevant (see section 4.7.4.4)

Please compress the Error Report into a ZIP archive and attach the the support request email.

1.5.4 Feedback

XIA strives to keep up with the needs of our users. Please send us your feedback regarding the functionality and usability of the DXP-XMAP and xManager software. We are also interested in hearing about improvements to the hardware and software. In particular, we are considering the following development issues:

1.5.4.1 Export File Formats

We would like to directly support as many spectrum file formats as possible. If we do not yet support it, please send your specification to software_support@xia.com.

1.5.4.2 Calibration

Currently the hardware gain of the DXP-XMAP is modified during energy calibration to produce a spectrum with a user defined bin scale, i.e. an integer electron-volts-per-bin value. The drawback is that the calibration

process often takes several iterations. Another approach to calibration is re-interpreting the bins. This is not difficult to do, but may produce confusion for the novice user. We are considering supporting this feature in future xManager releases.

2 Installation

CAUTION: *Improper connections or settings can result in damage to system components. Such damage is not covered under the DXP-XMAP warranty.*

Please carefully follow these instructions. It is important that you follow the steps in order: Install the PXI interface and identify the crate, install xManager and drivers, power off computer and crate, install DXP-XMAP hardware, power on crate, power on computer, identify hardware drivers, run xManager software.

2.1 PXI Interface Installation

XIA recommends using National Instruments PXI instrument crates and interface cards, and this document assumes this recommendation has been followed. In any case, please follow the instructions provided with your crate and/or controller.

2.1.1 PXI System Configuration File: PXISYS.INI

Make sure that your host computer properly communicates with the CompactPCI crate before continuing. Specifically, use the National Instruments Automation Explorer (NIMAX.EXE), to identify your crate. Doing so generates the PXI system initialization file "PXISYS.INI" in your windows directory. Please verify that the file exists in "C:\WINNT" directory for Windows 2000, or "C:\Windows" directory for Windows XP. This file is needed to properly start xManager the first time.

2.1.2 PXI Bus Segments and Slots

Each PCI bus segment holds up to 8 slots. Larger PXI crates employ bus extenders to provide host connectivity to two or more internal PCI bus segments. Such crates have front-panel markings—vertical hash marks between slot identifiers—to indicate where the bus segment boundaries are, e.g. for an 18-slot crate, slots 1-6 are in bus segment 1; slots 7-12 are in bus segment 2; slots 13-18 are in bus segment 3.

Note that the PXI bus connects slots in each bus segment, but does not extend across segment boundaries. The DXP-XMAP uses PXI bus trigger lines to synchronize multiple modules for timing applications. If there are multiple bus segments in your systems, timing signals, i.e. GATE and SYNC, must be physically connected to one module in each bus segment. That module will drive other modules in the segment via the backplane PXI trigger line.

2.2 Software Installation

Note: *Check for firmware and software updates at: http://www.xia.com/DXP_Resources.html*

Do not attempt to install the XMAP hardware until after the software and drivers have been installed. xManager operates on Windows 2000, XP and Vista machines. Updates to xManager are available online at:

www.xia.com/DXP-XMAP_Download.html

The update installation file is a executable, or .EXE file.

2.2.1 Running the Installer

- 1) Please close all applications that are currently running.
- 2) Insert the CD into the CD-ROM drive or, if your copy was delivered electronically, double-click the `setup.exe` program. If the CD installation does not start immediately, follow the instructions in steps (3) and (4).
- 3) Click the Start button and select the Run command.
- 4) Type `X:\Setup.exe` and click **[OK]**, where X is the letter of your CD-ROM drive.
- 5) After setup has completed, shut down your computer and complete the hardware configuration described in sections 2.3 through 2.5 before restarting.
 - The xManager 0.x installation will create a new directory: "C:\Program Files\xia\xManager 0.x".
 - A new Start Menu > Program group will be created.
 - A shortcut to the xManager executable is created on your desktop.
 - The hardware driver file "xmap9054.inf" is installed in C:\WINNT\inf" for Windows 2000, or "C:\Windows\inf" for Windows XP or Vista.

2.2.2 File Locations

The xManager default installation folder is:

```
C:\Program Files\XIA\xManager 0.x
```

This directory contains program files, libraries, log files and configuration, or INI, files. The "firmware" folder is:

```
~\xManager 0.x\firmware
```

This directory contains the normal and mapping firmware, or FDD, files (see section 1.4.3). Updates to the firmware are available online at:

www.xia.com/DXP-XMAP_Download.html

2.2.3 Support

For the latest documentation, please refer to XIA's website at

www.xia.com/DXP-XMAP_Download.html

XIA values all of the feedback it receives from customers. This feedback is an important component of the development cycle and XIA looks to use this feedback to improve the software. All bug fixes and feature suggestions should be directed to software_support@xia.com. Please be sure to include as much information as possible when submitting a bug report. For further instructions please refer to section 1.5.

2.3 Configuring the Analog Signal Conditioner

The term ‘jumper’ is used in this section. Jumpers are placed on 3-pin headers, connecting the center pin to one or the other peripheral pin, similar to a single-pole-double-throw (SPDT) switch. *Note:* Four equivalent DXP channels are present on a single DXP-XMAP card. Component references for each channel begin with a unique number. All components in channel 0 are numbered 1xx, in channel 1 are numbered 2xx, etc. For example the input attenuator for channel 0 is JP100. The equivalent attenuator for channel 1 is JP200, etc.

2.3.1 Input Attenuation: JP100, JP200, JP300, JP400

Attenuation may be necessary if the preamplifier gain or output voltage range is excessive and/or high-energy x-rays are to be processed. Pulses up to several hundred millivolts in size and a voltage range of +/- 5 Volts can be accommodated without attenuation. The default position for jumpers JPx00, labeled ‘0dB’ (see **Figure 2.1**), passes the signal directly. If larger signals must be processed, set JPx00 to the ‘-12dB’ position to reduce the input signal by a factor of four.

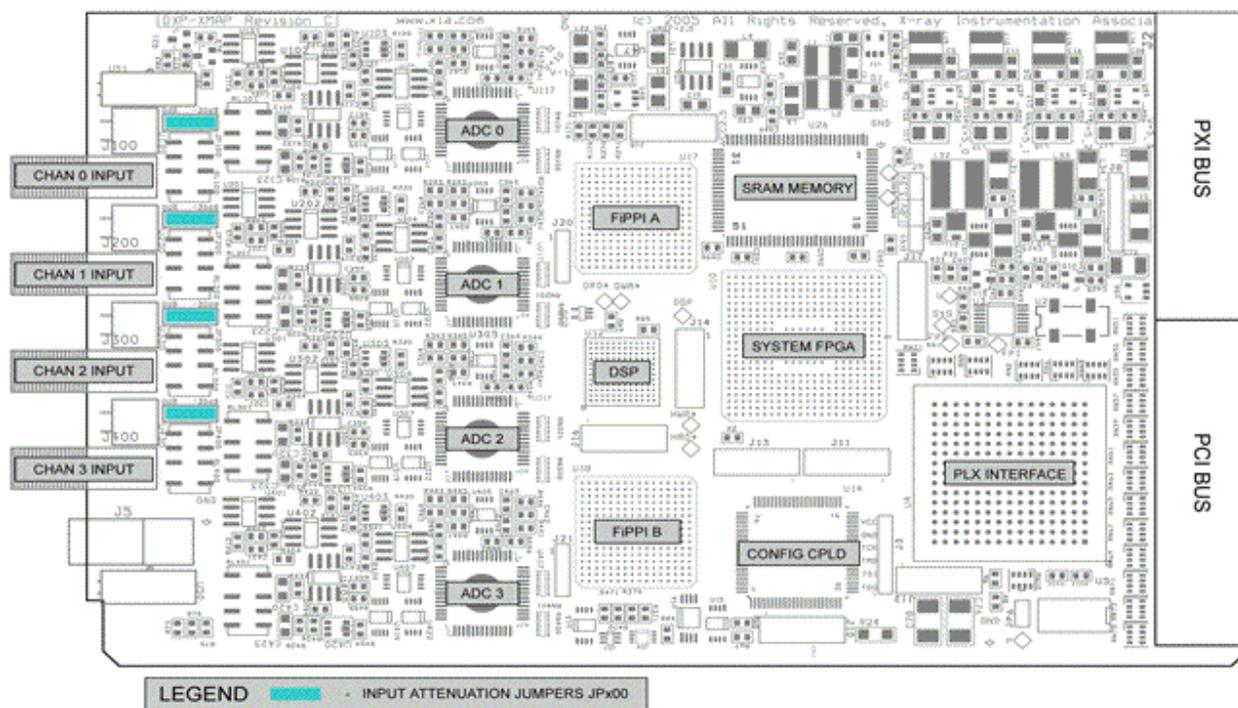


Figure 2.1: The DXP-XMAP printed circuit board. Input attenuation jumpers JPx00 are highlighted in blue.

2.4 Installing DXP-XMAP Cards

Note: XIA PXI modules are NOT hot-pluggable. Never attempt to install or remove an XMAP Digital Spectrometer module while the instrument crate is powered on. Such action can irreparably harm the module.

Turn off the host computer and PXI crate power. Install DXP-MAP modules in the PXI crate, making sure that the cards are fully inserted and each card's handle is secured such that it locks in the horizontal position. Connect your detector outputs to the xMAP SMA input connectors.

2.5 Making Connections

It is possible to damage the DXP-XMAP and/or connected equipment if the instructions below are not followed. All electronic connections are made at the front panel of the DXP-XMAP. We recommend using cables under three meters in length for signal connections to the preamplifier.

2.5.1 Signal Connections

The DXP-XMAP employs SMA connectors for size, reliability and signal quality. XIA provides four (4) BNC-to-LEMO cables with each DXP-XMAP. Fasten these cables to each DXP-XMAP input, and connect the other end to each detector/preamplifier output. Use BNC extension cables if necessary. The default ordering starts at channel 0 (upper input) of the left-most module in the crate and proceeds down that module, then up to channel 0 of the next module to the right, etc.

2.5.2 GATE/SYNC/LBUS Connection

Each DXP-XMAP includes a programmable TTL/CMOS level LEMO connection. This connection can be configured as a GATE or SYNC timing input for time-resolved spectroscopy applications. Typically one DXP-XMAP module of each PCI bus segment is designated the GATE or SYNC 'master' module which accepts the LEMO connection, and connects to other modules in the bus segment via a dedicated backplane PXI trigger line.

The LEMO connection can also be configured as an LBUS (READY) output to synchronize the start of data acquisition runs across multiple crates or bus segments. The LBUS function utilizes a wired-OR backplane PXI line to simultaneously start a run on all modules contained in a PCI bus segment—the run is started when all connected modules release this line, indicating they are ready. This wired-OR line can be extended between PCI bus segments when the front-panel LEMO connector is configured as an LBUS extension.

For now, do not make any LEMO connections. The Configuration Wizard utility will help you make decisions about which function(s) to use, and how to make the proper LEMO connections. See sections 3.2 and 7.2 for more details.

2.6 Starting the System

Make sure of the following before proceeding:

- ✓ Your system satisfies the requirements outlined in section 1.3 above.
- ✓ A PCI-to-optical interface card, e.g. National Instruments MXI-3 or MXI-4, has been installed and communicates properly with the host computer.

- ✓ The xManager software and drivers have been installed.
- ✓ DXP-XMAP modules have been installed.
- ✓ Detectors and preamplifiers are connected.
- ✓ A low-to-moderate intensity x-ray source is available for calibration and system verification.

Turn on the crate, detector HV and preamplifier power supplies, and restart the host computer.

2.6.1 DXP-XMAP Driver Selection

Windows should automatically find the new hardware and start the Found New Hardware Wizard.

- 1) The welcome screen prompts, "Can Windows connect to Windows Update to search for software?" Select "No, not this time" and press **[Next]**.
- 2) If the hardware is identified as an "XIA xMAP Digital Spectrometer", select "Install the software automatically (Recommended)" and press **[Next]** and proceed to step 6. If not, select "Search for a suitable driver for my device (recommended)" option and press **[Next]** to proceed to the "Locate Driver Files" page.
- 3) Select "Specify a location" and press **[Next]**.
- 4) Browse to, or type "C:\WINNT\inf" for Windows 2000, or "C:\Windows\inf" for Windows XP or Vista, and press "OK"
- 5) Windows should find the file "xmap9054.inf" in the specified directory. Press **[Next]**.
- 6) A warning may appear regarding Windows Logo testing. Press the **[Continue Anyway]** button to complete the driver installation.
- 7) Repeat steps 1-6 for each xMAP module installed.

Note: Driver selection can be changed at any time via the Windows Device Manager. To open the Device Manager, right-click on the "My Computer" icon and select "Manage". Now click on "Device Manager" in the left-pane of the Computer Management window. DXP-XMAP cards can be found under "Other Devices".

3 System Configuration

At this point the xManager software and drivers should have been installed, and the DXP-XMAP hardware should be powered on and identified by Windows. This chapter will guide you in using the xManager Configuration Wizard utility.

3.1 Initialization Files

After power up the DXP-XMAP's DSP and programmable logic are in an unknown state. Program code, or firmware, for these devices must first be downloaded via the PCI bus before data can be acquired. After the devices are operational, user settings are downloaded.

Handel (and thus xManager) uses an initialization (INI) file to store all necessary configuration information, including the path and filename of the firmware file on the host computer, number and slot location of xMAP modules in the system, detector characteristics and spectrometer settings, and timing and synchronization logic functions used. In order to start properly, xManager needs to have the following information:

- ✓ The location of the xMAP FDD firmware file (DSP and FPGA code that runs on the board, included in the installation package).
- ✓ The number and location of xMAP modules in the system.
- ✓ Various properties of the detector preamplifier including type, polarity and gain.
- ✓ Which timing and synchronization functions are to be used. Master and slave modules will be designated automatically.
Note: section 7.2 describes timing and synchronization logic.

INI files can be updated at any time, i.e. after the spectrometer settings have been optimized, and existing INI files can be loaded at any time. If you have previously run with xManager, your registry settings will point to the most recently used INI file, and xManager will automatically run with these settings upon startup.

3.1.1 Starting xManager Without an INI File

Start xManager via the Start menu: Start > Programs > xManager 0.x > xManager. The first time xManager 0.x starts up, the xManager Configuration File Error panel will appear, as shown in below.



To open the Configuration Wizard: Select "Configuration Wizard" from the "Tools" menu

Figure 3.1: The Configuration File Error appears the first time you run xManager because a valid INI file has not been selected.

Press the "Generate New File" button to launch the Configuration Wizard, which guides the user step-by-step to create an INI file.

3.2 The Configuration Wizard

The Configuration Wizard utility can be launched at any time from the "Tools" menu in xManager.

3.2.1 General Settings

These basic settings are the bare minimum necessary to run the DXP-XMAP in normal single-spectrum mode.

- 1) **Welcome to the xMAP Configuration Wizard**
The first panel of the Configuration Wizard is simply a welcome screen with some information about the utility. Press [Next] in the xMAP Configuration Panel.
- 2) **Starting Template**
For now, select <default blank template> and press [Next]. Note that if you have an existing INI file you can select it now to use as a template such that some data entry steps below are skipped. This is quite useful when creating multiple similar configurations.
- 3) **Firmware**
The firmware file contains all program code for the programmable devices on the DXP-XMAP. Press the [FDD File...] button to browse, or type "C:\Program Files\xia\xManager 0.x\firmware\xmap_revb.fdd" and press [Next]. If you have updated your firmware since xManager was installed, be sure to select the new file. Note that different firmware files are required for pulsed-reset and RC-feedback type preamplifiers. Updates to the firmware are available online at:

www.xia.com/DXP_Resources.html.

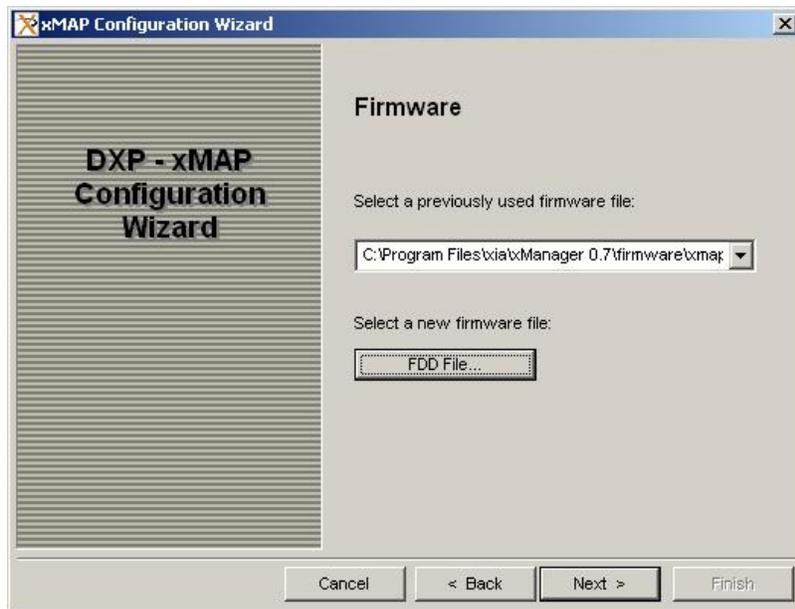


Figure 3.2: The firmware file contains program code for the DXP-XMAP's programmable devices.

4) **Detector Configuration**

Select the appropriate detector type. For **Reset** type, enter the **Reset Interval**. This is the time in microseconds that the preamplifier takes to reset and settle, and should be set conservatively to prevent associated voltage transients from entering the spectrum. If you don't know the reset time, enter 10 (microseconds). For **RC Feedback** enter the **RC Decay Time** in microseconds. Press **[Next]**.

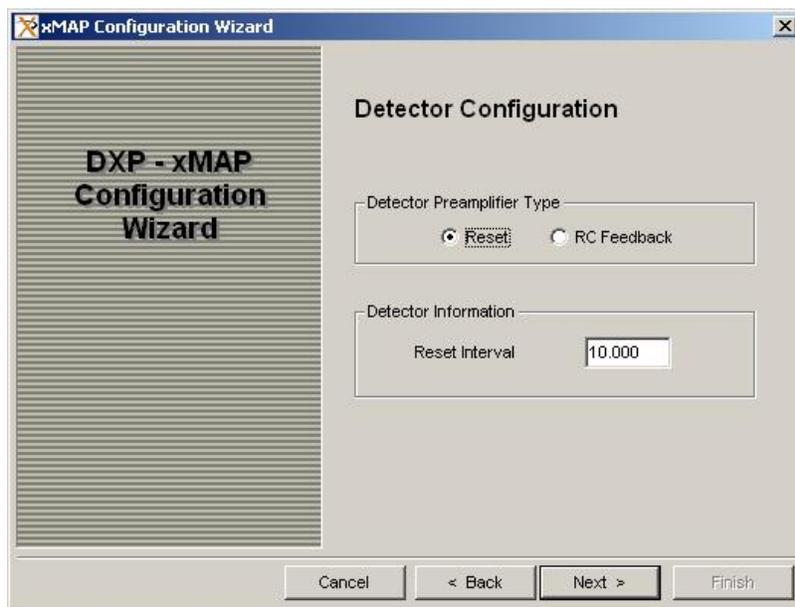


Figure 3.3: The Detector Configuration settings.

5) **Hardware Configuration**

If you selected **<default blank template>** in step 1) above, or you have

changed the slot configuration or number of modules in your system, select **Use current system information**. This option will result in the parsing of the PXI system configuration file "pxisys.ini" described in section 2.1.1 above, such that xMAP modules located in the crate are organized from left to right. If you selected an existing INI file in step 1) above, and would like to use the slot configuration or number of modules stored therein, select **Use the values from template file**. Press [Next].

6) **XMAP PCI Configuration**

This panel displays all located DXP-XMAP modules, ordered by default from left to right. At this point it is possible to re-order the modules and/or disable modules or individual processing channels if, e.g. your system has more XMAP channels than detector elements. If you are happy with the default configuration, make note of whether more than one PCI bus is listed. Press [Next] and skip to 8) below.

- a) *To re-order modules:*
Simply edit the **PCI Bus** and **PCI Device ID** fields manually such that the slot order displays as desired. Press [Next] and skip to 8) below.
- b) *To disable module(s):*
Simply edit the **Number of Modules** field. By default the right-most slot is disabled. Press [Next] and skip to 8) below.
- c) *To disable individual channels:*
Edit the **Number of Active Channels** field, check the **Configure Individual Channels** checkbox. Press [Next] and proceed to 7) below.

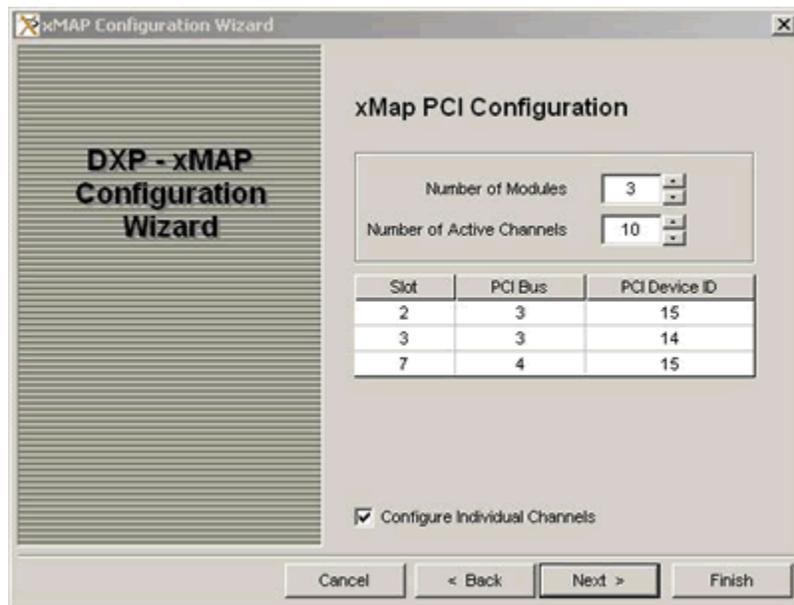


Figure 3.4: This three-module system occupies slots 2, 3 and 7. Two channels are to be disabled.

7) **XMAP Channels Configuration**

This panel displays only if you checked the **Configure Individual**

Channels checkbox in the previous panel. By default the right-most, highest-numbered channels are disabled first. To change which channels are disabled, simply click in the **Enable** column.

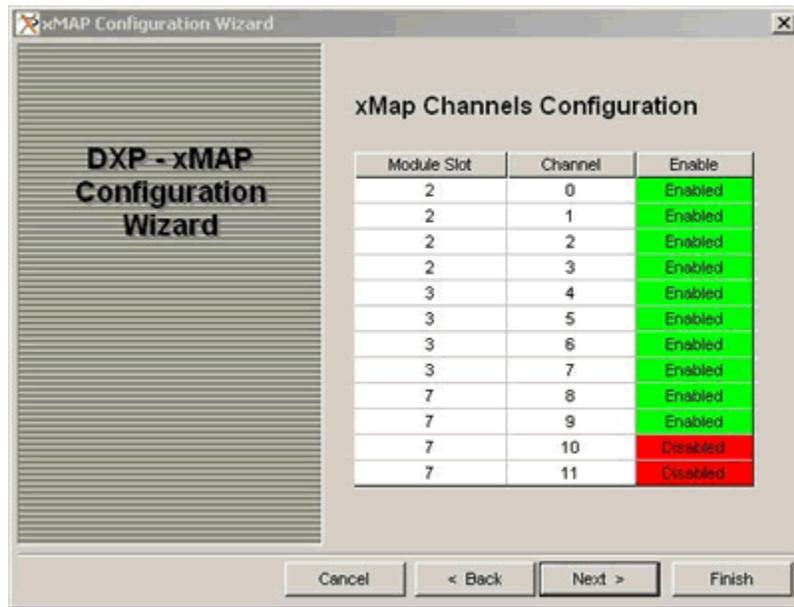


Figure 3.5: This panel allows you to choose which channels are disabled if, for example, your system has more XMAP channels than detector elements.

8) **Detector Element Property**

Each DXP-XMAP processing channel (4 per module) includes a programmable gain analog stage to compensate for the detector gain. Initially the same polarity and gain should be used for all channels. During the calibration process the gain can be fine-tuned for each channel and the INI file updated.

Check the **Apply to All Detector Elements** checkbox. Click in the **Polarity** column: "-" if xray steps generate a negative voltage step; "+" if xray steps generate a positive voltage step. If you don't know the polarity, keep the default (negative) setting. Enter the gain in [mV/keV]. If you don't know the gain, keep the default of (3 mV/keV). Press **[Next]**.

3.2.2 **Hardware Synchronization Settings**

These settings are used to configure the front panel LEMO connection and backplane trigger lines for normal single-spectrum mode. If you intend only to use the DXP-XMAP in single-spectrum mode without the synchronization features, you can skip ahead to save the generated INI file.

9) **Hardware Timing Synchronization**

If you want to use the synchronization features and/or use the DXP-XMAP in mapping mode, select **Configure hardware for synchronized run**, press **[Next]**, and proceed to 10) below. Otherwise select **Skip to save current configuration**, press **[Next]** and skip to 17).

10) Multi-Bus Synchronous Run Start

The LBUS function is used to synchronize the start time of modules in different PCI bus segments. The **Configuration Wizard** detects whether your XMAP modules populate multiple bus segments, and if not, 'grays out' the LBUS option. Please review section 7.2.2 for a complete description of this feature.

If more than one bus segment *is* populated we recommend using the LBUS function, but it is not necessary. If used, the left-most module in each PCI bus segment will be designated as the LBUS master, i.e. it accepts the front-panel LBUS connection. Make your selection and press **[Next]**.

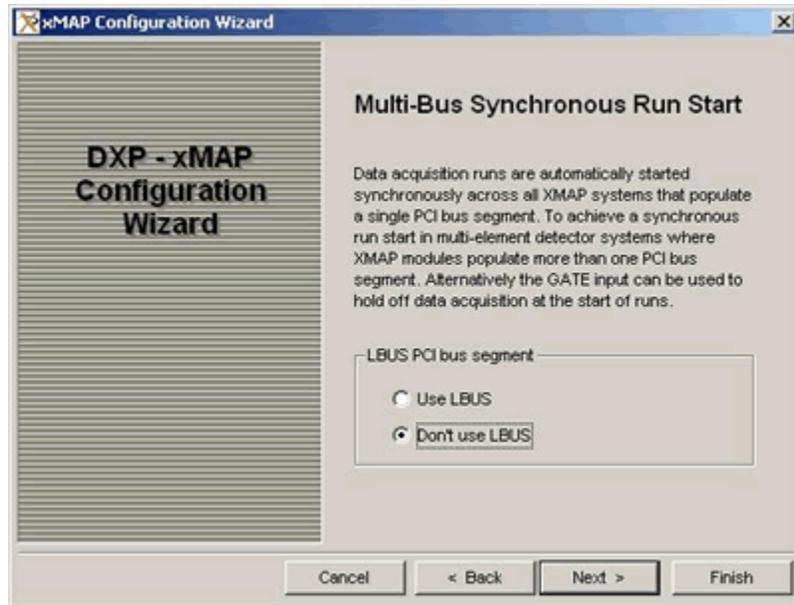


Figure 3.6: In this case LBUS will not be used.

11) GATE Function

The GATE function is used to selectively halt data acquisition during a run according to a user-provided TTL/CMOS logic signal. Please review section 7.2.3 for a complete description of this feature.

The **Enable GATE** setting reserves the left-most available module in each PCI bus segment as the GATE master, i.e. it accepts the front-panel GATE connection. The "GATE Polarity" setting determines whether data is halted when the GATE logic signal is LO or HI. Make your selections and press **[Next]**.

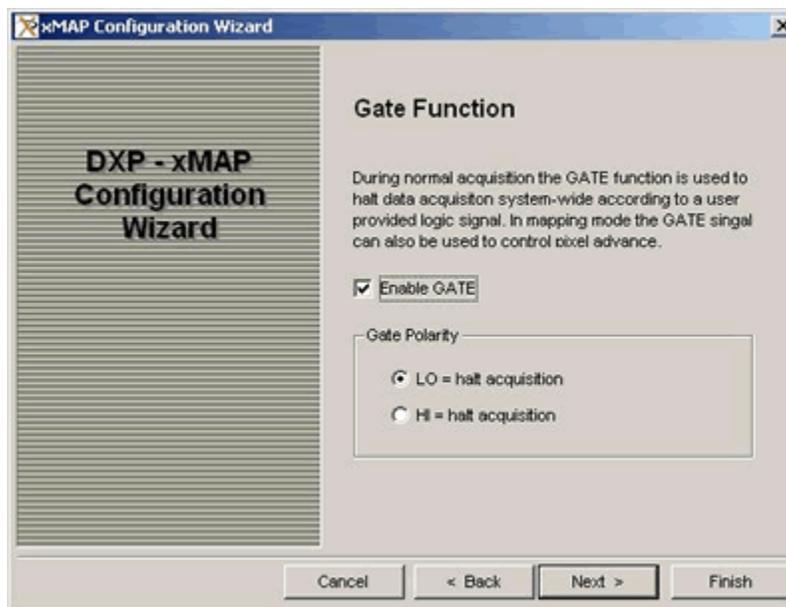


Figure 3.7: For this system GATE is enabled, set to halt acquisition when LO.

3.2.3 Mapping Mode Settings

The remaining settings relate to the mapping mode wherein multiple spectra are acquired for each processing channel in a single run, e.g. to produce an elemental map of the sample in x-ray scanning applications. Please review section 1.2.1.1 for a description of the mapping modes.

12) Mapping Configuration

If you want to use the DXP-XMAP in mapping mode, select "Continue with mapping configuration" and press [**Next**]. Otherwise select "Don't use mapping mode" and press [**Next**].

13) Pixel Advance Mode

The Pixel Advance triggers the change to a new spectrum in multiple-spectrum data acquisition. Typically the Pixel Advance is controlled by a user-provided logic signal. In GATE mode each leading edge transition generates a Pixel Advance instruction. See section 7.2.4 for a description of this mode. In SYNC mode a Pixel Advance instruction is generated every N LO-to-HI transitions. See section 7.2.5 for a description of this mode. In User mode the Pixel Advance is triggered by a command from the host computer.

14) The next panel depends on the Pixel Advance Mode selection:

a) GATE Pixel Advance Options

As described in section 7.2.4.3, in GATE Pixel Advance mode the GATE signal by default also halts data acquisition. If "Pixel advance only" is selected on this panel, data will be written to the new spectrum immediately after each leading edge transition regardless of the pulse-width.

Note: The polarity selection made in step 11) above is used for both the normal and mapping modes, e.g. if "LO = halt acquisition" was selected, the pixel advance occurs on the HI-to-LO transition.

- b) **SYNC Pixel Advance Options**
 The SYNC pixel advance occurs after the selected "Number of cycles" is detected, on the desired "Trigger Edge".
Note: Selecting SYNC mode reserves the left-most available module in each PCI bus segment as the SYNC master, i.e. it accepts the front-panel SYNC connection.
- c) **User Pixel Advance Options**
 There are no options for the this mode; the utility skips to next panel.
- 15) **Timing Mode Run Control Options**
 The DXP-XMAP can automatically stop the data acquisition run after a prescribed number of pixels. The "Number of Pixels Per Run" setting can easily be modified later in xManager. If it is set to minus one (-1) the run continues until the user stops the run.
 A dual memory architecture is used to achieve continuous operation in mapping mode. Each memory device is 1,048,576 words in size. The "Number of Pixels Per Readout" is slightly less than the total device size divided by the individual spectrum size. If zero, or a number greater than acceptable, is selected the largest number that can be used is automatically calculated. This setting can easily be modified later in xManager.

3.2.4 Completing the Configuration

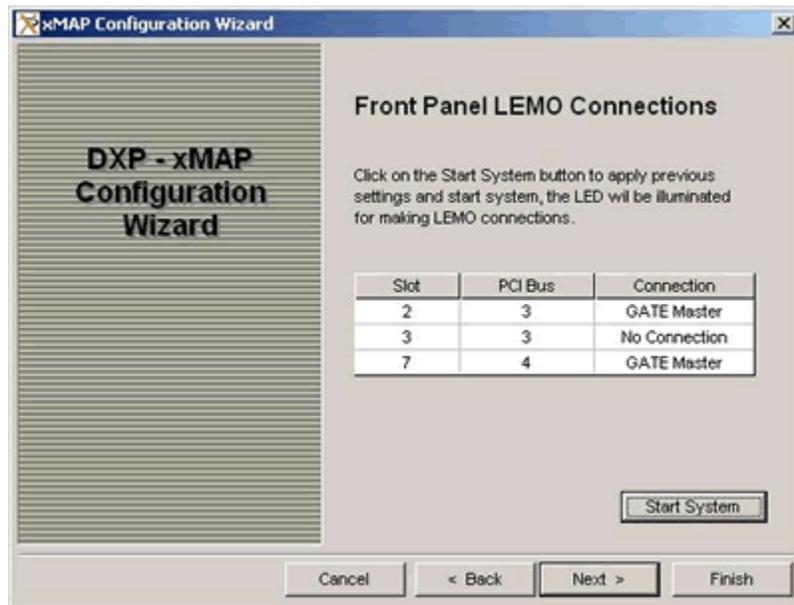


Figure 3.8: As expected one module in each PCI bus segment (3 and 4) has been designated as a GATE Master.

- 16) **Front Panel LEMO Connections**
 This panel displays the resulting front panel LEMO assignments. This is a good time to make those connections, if applicable. The [**Start System**] button immediately configures the DXP-XMAPs. Note that after configuration the front panel LEDs also indicate the front panel LEMO connections:

- a) LBUS Master Modules
These modules must all be connected together using LEMO cables and T-adaptors. LBUS can also be used as a 'READY' output as described in section 7.2.2.
 - b) GATE Master Modules
These modules must be connected to the users GATE signal generator.
 - c) SYNC Master Modules
These modules must be connected to the users SYNC clock generator.
- 17) **Save Completed Configuration**
The INI file you have created can now be saved. Select a unique name for the file, e.g. "C:\Program Files\xia\xManager 0.x\ixmap_32module.ini". Press **[Finish]** to save the INI file and exit the **Configuration Wizard**. Note that if you did not start the system in 16) above, you must load the INI file to enact your changes.

3.3 Loading and Saving Initialization Files

INI files can be updated at any time, i.e. after the spectrometer settings have been optimized, and existing INI files can be loaded at any time. If you have previously run with xManager, your registry settings will point to the most recently used INI file, and xManager will automatically run with these settings upon startup.

3.3.1 Loading an INI file

Select "Load Configuration" from the File menu. Browse to and select an INI file that you just created and press "Open". xManager will download firmware and initialize the xMAP modules in your system.

3.3.2 Saving an INI file

INI files can be updated at any time, i.e. after the spectrometer settings have been optimized, by selecting "Save Configuration" or "Save Configuration As..." from the "File menu. You may find it useful to maintain several INI files, e.g. for operating with different detectors, or with different spectrometer settings.

4 Using xManager with the DXP-XMAP

At this point the xManager software and drivers should be installed, the DXP-XMAP hardware should be powered on and identified by Windows, and a valid initialization file should have been created and loaded. This chapter will guide you in using xManager with the DXP-XMAP module.

4.1 A Quick Tour of xManager

xManager is a PC-based application that provides for the setup, optimization and failure diagnosis of the instrument, and allows for the reading out, displaying, analyzing and exporting of acquired energy spectra. When you start the program and an initialization file has been loaded, the xManager main window should be displayed as in Figure 4.1.

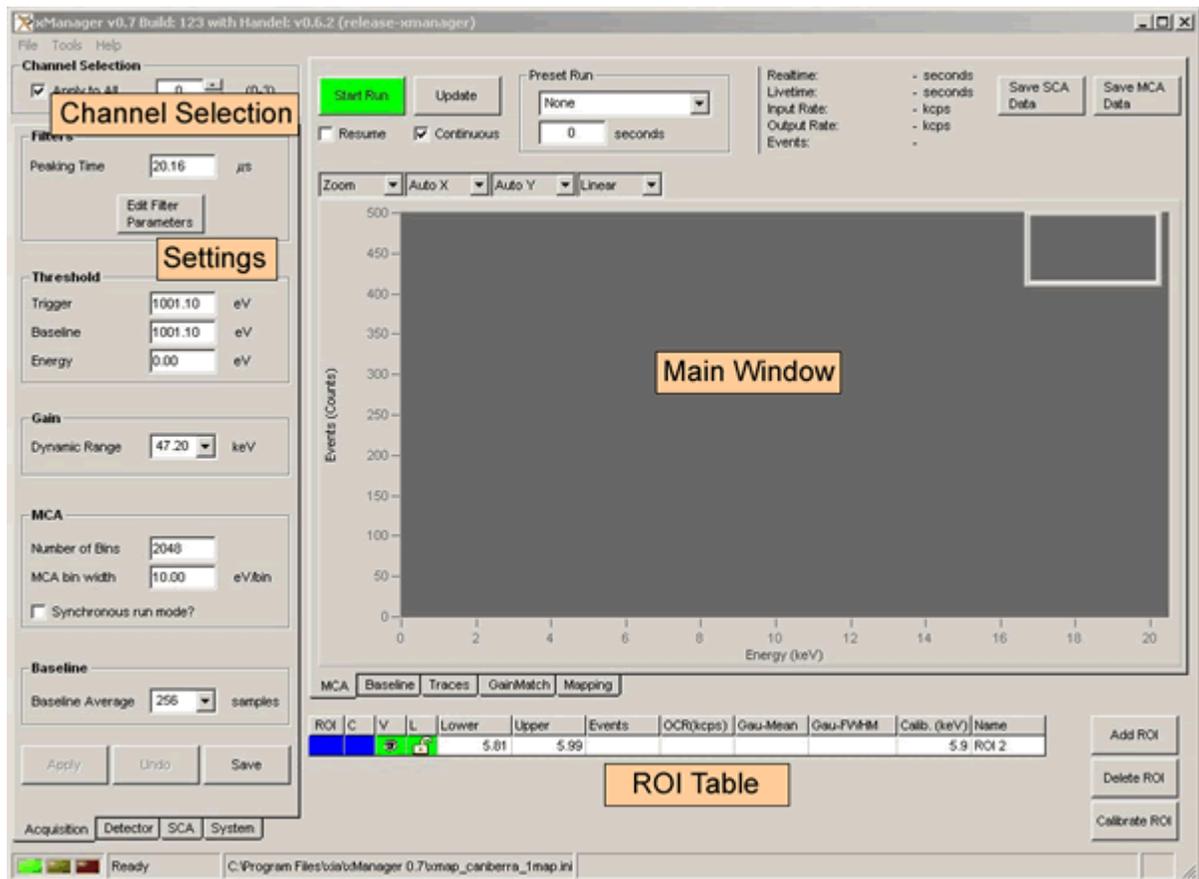


Figure 4.1: The xManager main window upon startup, after hardware initialization.

4.1.1 Channel Selection

Each DXP-XMAP module provides four (4) digital x-ray processing channels. The **Channel Selection** global control sets the channel for which

settings and data are displayed in xManager. The "Apply to All" checkbox applies to settings only: If the checkbox is checked, any change to settings will be applied to all channels simultaneously.

Note: Individual channels can be *disabled* (see section 3.2.1 #6)) so that they are invisible to xManager. This is appropriate when the number of detector elements is not a multiple of 4, i.e. some XMAP channels are not connected to the detector. Alternatively individual channels can be *skipped* (see sections 4.3.2 and 4.3.6.1) within xManager. This is useful for selective data acquisition and calibration.

4.1.2 Settings Sidebar

The tabbed **Settings** sidebar provides easy access to all hardware and firmware settings. It is intended to be the primary interface for setup and optimization. The **Configuration** tab contains spectrometer settings such as peaking time and thresholds. The **Detector** tab contains detector and preamplifier settings such as polarity and gain. The **SCA** tab displays Single Channel Analyzer data, if applicable, based on user-entered regions of interest. The **System** tab displays front panel logic assignments for each DXP-XMAP module, and provides for skipping individual processing channels.

4.1.3 Main Window

The tabbed **Main Window** contains the MCA, Baseline, oscilloscope, and system calibration. The **MCA** tab is used for normal mode spectrum acquisition. The **Baseline** tab displays the baseline histogram. The **Trace** tab contains the oscilloscope tool for displaying ADC, filter output, and baseline data. The **GainMatch** tab contains the system-wide gain calibration tool. The **Mapping** panel is used for time-resolved multi-spectrum and multi-SCA data acquisition.

4.2 Detector and Preamplifier Settings

If the Configuration Wizard was followed correctly as described in section 3.2, the system should be nearly ready for data acquisition. Before taking a spectrum, however, we recommend verifying the Detector and Preamplifier settings.

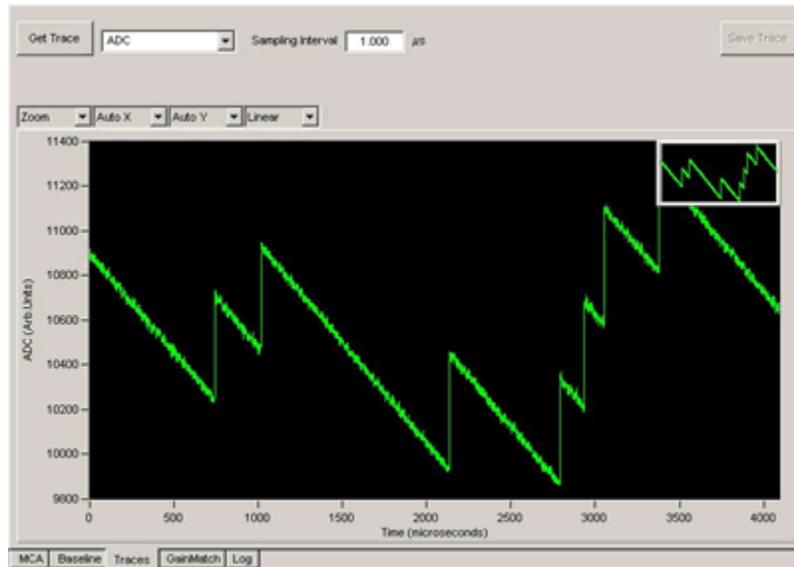


Figure 4.2: An ADC trace displayed in the **Trace** panel oscilloscope tool. Notice that the displayed x-ray events (7 total) are voltage steps with *rising* edges, thus the polarity is set correctly.

Select the **Trace** tab in the **Main Window** to display the oscilloscope tool (see **Figure 4.2**). Select **ADC** from the drop-down list, set the **Sampling Interval** to "1.000" μ s and press the **Get Trace** button to display a 4096-point raw ADC data set.

Select the **Detector** tab of the **Settings** panel. The **Polarity** setting enables or disables a digital inverter depending on the signal polarity of the preamplifier. The **Reset Interval** is the settling time, in microseconds, of the preamplifier reset. The **Preamp Gain** is the gain, in milli-Volts per kilo-electron-Volt of the charge sensitive preamplifier. The **Apply** button downloads the adjusted setting(s) to the DXP-XMAP hardware. The **Save** button saves adjusted setting(s) to the currently selected INI file. For a thorough discussion of oscilloscope diagnostic tool, please review section 4.7.1.

Note: Do not confuse detector bias polarity with the polarity of the preamplifier signal; they are not necessarily related.

4.2.1 Pre-Amplifier Polarity

Preamplifier polarity denotes the polarity of the raw preamplifier signal, NOT the detector bias voltage polarity. A positive polarity preamplifier produces a positive step, defined as a voltage step with a rising edge, response to an incident x-ray. The digital filters in the DXP-XMAP expect an input signal with positive steps. An optional input inverter is employed to correct the signal polarity for negative polarity preamplifiers. If the polarity has been set correctly, the ADC oscilloscope trace should display positive steps.

If the ADC trace displays positive steps (as in **Figure 4.2**), the polarity has been set correctly. If not, change the **Polarity** setting and press the **Apply** button. Acquire a new trace to verify that the polarity setting is correct.

Please read through section 4.7.1 for a thorough description and figures relating to the preamplifier signal polarity.

4.2.2 Reset Delay

The **Reset Delay** is the period of time after each preamplifier reset that the DXP-XMAP waits before re-enabling data acquisition. The delay is set based on the settling time of the preamplifier reset transient waveform, typically ranging from hundreds of nanoseconds to hundreds of microseconds. If you are unsure, enter "10" μ s. Setting the delay shorter than the transient settling time may introduce 'reset artifact' events into the spectrum. Setting the delay longer than necessary introduces additional processor dead time, which will reduce the data throughput at high count rates.

4.2.3 Preamp Gain

The **Preamp Gain** setting, in combination with the dynamic range setting, controls the DXP-XMAPs variable gain amplifier such that the input requirements of the ADC are satisfied, given the gain of the preamplifier. If you know the gain of your preamplifier, enter that value. Otherwise we recommend using the default value of 3mV/keV. This setting is normally adjusted automatically during energy calibration. In cases of extremely low or high preamplifier gain, it may be necessary to adjust the nominal gain before taking a spectrum. If the displayed x-ray steps are less than 50 ADC units in height, reduce the **Preamp Gain** setting. If the displayed x-ray steps are greater than 2,000 ADC units in height, increase the **Preamp Gain** setting.

4.2.4 Preamp Risetime

This is an advanced setting, accessible by pressing the [**Edit Filter Parameters**] in the **Acquisition** settings tab. The preamplifier risetime should be measured and the **Minimum Gap Time** set accordingly. This setting is described in detail in section 4.6.1.2. See section 4.7.1.2 for details on using xManager to measure the risetime for your system and section 6.3 for a theoretical discussion of the issues involved in trapezoidal filtering.

4.3 Normal Spectrum Mode Data Acquisition

*To begin data collection:
Select the MCA tab and
press the [Start Run]
button.*

4.3.1 Starting a Run

Once the detector/preamplifier settings have been verified you are ready to collect a sample spectrum. Place a known X-ray source, for example an ^{55}Fe source that produces Mn K_{α} line at 5899 eV, such that x-rays strike the detector's active area at a moderate to low rate, i.e. less than 10,000 x-rays absorbed per second.

Select the **MCA** tab and press the [**Start Run**] button in the data display panel to begin data collection. An uncalibrated energy spectrum should appear. Figure 4.3 shows a sample uncalibrated ^{55}Fe spectrum. Proceed to section 4.3.2 if a spectrum is displayed.

No spectrum?

Check your hardware setup, e.g. x-rays present?

- ✓ Check your initialization settings, e.g. preamplifier type and polarity correct?
- ✓ Troubleshoot the signal using the Oscilloscope tool, as described in section 4.7.1.

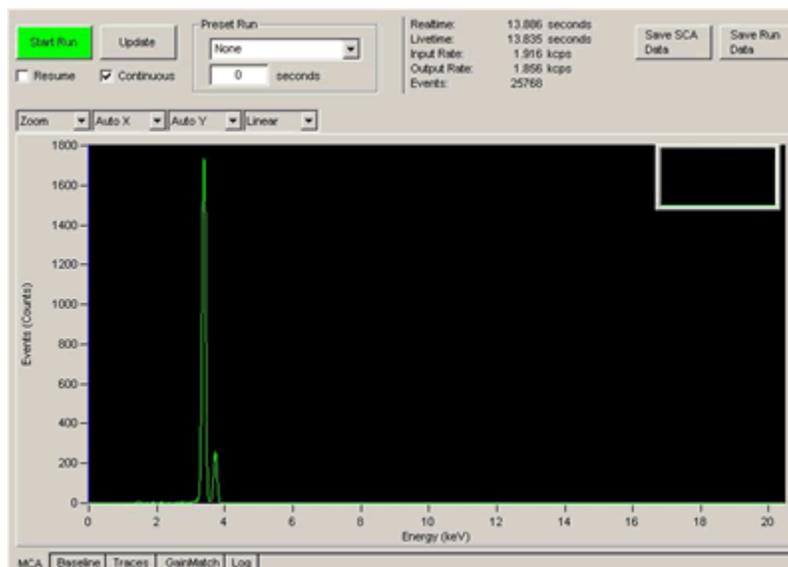


Figure 4.3: An uncalibrated ^{55}Fe spectrum.

Press the [**Refresh**] button to manually read out the MCA data, or check the **Continuous** checkbox to automatically refresh the spectrum. A horizontal line at zero on the y-axis indicates that no output events have been acquired, although the run is active. This can result from a hardware setup problem, e.g. x-rays not hitting detector; detector not powered, etc. Or it can result from incorrect configuration settings. The most common problem is incorrect detector/preamplifier settings. To troubleshoot these settings please refer to the Diagnostics section 4.7.

4.3.2 Skipping Channels

The manual or automatic MCA data readout operates on all *active* processing channels, i.e. though only one spectrum is displayed, data from all channels is locally accessible via the **Channel Selection** control.

Processing channels can be disabled, or skipped, from the readout operation. Select the **System** settings tab to display the **Channel Selection Detail**. Click in the **Skip Session** column to de-select individual channels. *Note:* Channel skipping also applies to system-wide gain matching as described in section 4.3.6.

Channel	C	Skip Session	Run Status
0			Ready
1			Ready
2			Ready
3			Ready
4			Ready
5			Ready
6			Ready
7			Ready
8			Ready
9			Ready
10		Skipped	
11		Skipped	

Figure 4.4: The **Channel Selection Detail** control in the **System** settings tab. In the system shown, MCA data readout and system-wide gain matching would be skipped for channels 10 and 11.

To change common acquisition settings select the **Configuration** tab of the settings panel.

4.3.3 Spectrometer Settings

The primary spectrometer settings are immediately accessible via the **Configuration** tab in the settings panel.

4.3.3.1 Peaking Time (Energy Filter)

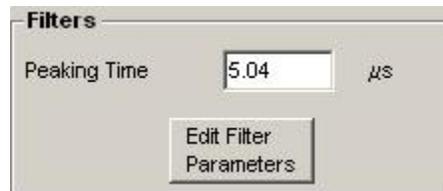


Figure 4.5: The **Peaking Time** is displayed in the **Configuration** tab of the **Settings** panel.

*Note: The energy filter peaking time is widely referred to as “**peaking time**”, whereas the fast filter peaking time is referred to as “**fast peaking time**”*

The energy filter peaking time is one of the primary user controls. Generally speaking, a longer peaking time produces better energy resolution at the cost of increased dead time and thus lower output count rate. In practice, the user may set the peaking time to a shorter than optimal value in order to increase data throughput, making up for degraded energy resolution with improved statistics. Most detectors also have an upper limit above which the energy resolution gets worse. HPGe detectors typically have optimal peaking times between 16μs and 32μs. Silicon drift detectors often produce the best resolution at 10μs or less. Some SiLi detectors show resolution improvements out to 80μs or longer.

Making a plot of energy resolution versus peaking time provides a useful future reference.

The **[Edit Filter Parameters]** button accesses additional filter parameters, including the energy gap time, the fast, or trigger, filter settings and pileup rejection parameters. The default filter settings reflect a compromise between robustness and performance and typically do not need to be changed. In some cases energy resolution for a given peaking time can be improved significantly if these settings are optimized as described in section 4.6.

4.3.3.2 Trigger Threshold

The trigger, or fast, filter threshold sets the low-energy limit for the fast filter, which is used primarily for pileup inspection. If the baseline threshold is employed, the *detection* of x-rays actually extends to energies significantly below the trigger threshold (see section 4.3.3.3). For this reason it is not necessary to set the trigger threshold aggressively, i.e. setting the threshold *as low as possible* will derive little benefit. If set too low, the trigger threshold will introduce a zero energy noise peak into the spectrum. In extreme cases it will halt data throughput entirely.

To optimize the fast filter threshold, set the **Baseline Threshold** to zero (so that output events are generated by fast filter triggers only), edit the **Trigger Threshold** value and press **[Apply]**. Typical values range from 600eV to 1500eV. A good procedure is to initially set the value too high, reduce it until the zero energy noise peak starts to become significant, and then raise it again until the noise peak is eliminated.

The fast filter length is independent of the energy filter length, or peaking time, thus the trigger threshold does NOT need to be optimized every time the peaking time is changed. All thresholds must be readjusted if the gain changes significantly.

4.3.3.3 Baseline Threshold

Note: The baseline threshold is not available for decimation 0, i.e. peaking times less than or equal to 500 ns.

The baseline threshold sets the low-energy limit for the intermediate, or baseline, filter, which is used for both baseline acquisition and low-energy x-ray detection. To optimize the baseline filter threshold, first optimize the trigger threshold as described above, then edit the **Baseline Threshold** value and press **[Apply]**. Typical values range from 150 eV to 1000 eV.

The baseline filter length is linked to the energy filter length, or peaking time, thus the baseline threshold should be optimized every time the peaking time is changed. All thresholds must be readjusted if the gain changes significantly.

Please review section 6.4 for a detailed description of baseline acquisition and averaging. Section 4.7.2.1 describes the empirical optimization of the baseline threshold

4.3.3.4 Energy Threshold

The energy threshold sets the low-energy limit for the slow, or energy, filter, which is used primarily for measuring the pulse-height, i.e. energy, of x-ray voltage steps. Triggering on the energy (slowest) filter can extend the detection range down to the lowest energies for a given detector, however, in most cases we recommend setting the **Energy Threshold** to zero. This is

CAUTION: *In almost all cases the Energy Threshold should be set to zero. An error term in the counting statistics is introduced when the Energy Threshold is enabled. For this reason it should only be enabled at low data rates.*

because the dead time associated with x-rays detected by the energy filter can not be directly measured. It remains available primarily for two special cases:

- A non-zero energy threshold is appropriate for ultra-soft x-ray detection at very low input count rates.
- A non-zero energy threshold may be used to extend the detection range for decimation 0, i.e. peaking times less than 500 ns. Dead time and count rate statistics will however be distorted.

4.3.3.5 *Dynamic Range*

The dynamic range setting combines with the detector gain setting to determine the variable analog gain of the DXP-XMAP. The variable gain is set such that an x-ray with energy equal to the dynamic range value produces a voltage step of the maximum allowable amplitude at the ADC input. X-rays with energies exceeding the dynamic range value cannot be processed correctly. The presence of such x-rays can result in a significant reduction in the output count rate. The **Dynamic Range** setting should be set above the largest x-ray energy present in the system. Typical values range from 40keV to 100keV. Edit the **Dynamic Range** value and press [Apply].

4.3.3.6 *MCA Number of Bins and MCA Bin Width*

The size and granularity of the spectrum can be adjusted. The number of spectrum bins sets the granularity of the acquired spectrum. The eV/Bin setting determines the size of each MCA bin in electron Volts. Together, these settings determine the energy span of the MCA: The spectrum ranges from zero to a maximum energy equal to the number of spectrum bins multiplied by the MCA bin width (e.g. a 40.96keV spectrum results from 2048 bins at 20eV/bin).

Note that these digital spectrum controls are independent of the **Preamp Gain** and **Dynamic Range** settings that control the variable analog gain. If the MCA energy range is less than the dynamic range, the entire spectrum will be free of distortion. If the MCA energy range exceeds the dynamic range setting, the spectrum will be distorted: Higher energy x-ray data will be attenuated or cut off. For this reason the product of **Number of Bins** and **MCA Bin Width**, i.e. the MCA energy range, should be less than the dynamic range. Edit the **Number of Bins** and **MCA Bin Width** values and press [Apply]. Start a new run.

4.3.3.7 *Baseline Average*

The baseline is the output of the energy filter in the absence of x-rays. A running average of baseline samples, acquired between x-ray events, is subtracted from the x-ray peak samples to arrive at the true energy of incident x-rays. A perfect detector and preamplifier would produce a constant baseline, however, in the real world the actual baseline varies. The number of **Baseline Average Samples** can strongly affect performance. More samples improve noise reduction but slow the reaction time to actual changes in the baseline. In most cases a value between 64 and 512 will produce the best results.

Please review section 6.4 for a detailed description of baseline acquisition and averaging. Section 4.7.1.4 describes the empirical optimization of the number of samples in the baseline average.

4.3.4 Setting Regions of Interest (ROIs)

A region of interest (ROI) is a user-defined energy range of the MCA spectrum for which separate statistics are displayed in the ROI Table. Typically an ROI corresponds to an energy peak. ROIs are used for energy calibration and SCA acquisition.

4.3.4.1 Adding ROIs

The **Region Of Interest** table is located below the spectrum. A single ROI is displayed by default. If you cannot see the ROI table, slide the panel separator up and/or press the “Add” button. The four leftmost columns in the table control the display of the ROIs. The first column indicates which ROI is active—only one ROI can be active at a time. Note that the displayed color and visibility can be set, and that each ROI can be locked.

The **Lower** and **Upper** bounds of the ROI can be entered manually in the table, or automatically created for a given peak using the **Auto ROI** function.

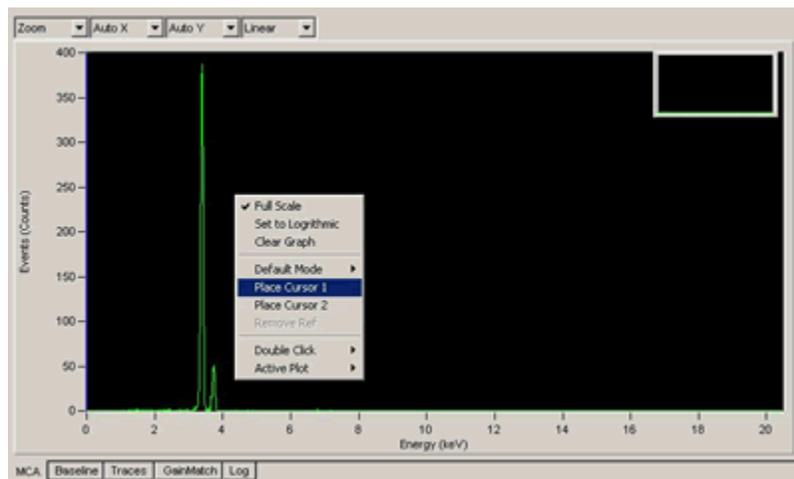


Figure 4.6: The full spectrum window ‘right-click’ menu is displayed. Context sensitive menus are available if the mouse pointer is located over the x-axis or y-axis, or over the cursor.

4.3.4.2 Auto ROI

The **Auto ROI** function generates lower and upper bounds for the active ROI about a selected energy peak. Place the mouse pointer over the spectrum peak of interest. Right-click and select “**Place Cursor 1**”. Drag the Cursor 1 to the center of the calibration peak, right click on Cursor 1 and select “**Auto ROI**”. A region of interest should automatically appear on the peak. In some cases, where few events have been collected, the Auto ROI feature will not properly enclose the peak. In these cases, the ROI can be adjusted directly in the Spectrum Window.

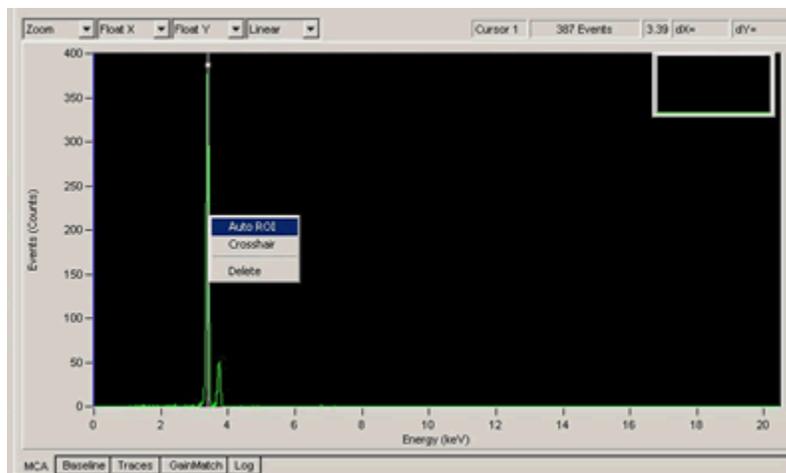


Figure 4.7: The Auto ROI function (found in the cursor context menu) automatically defines a region of interest around the peak selected with the active cursor. The cursor context menu is displayed by right-clicking on a cursor.

4.3.5 Single Channel Gain Calibration

Multi-element detectors often have significant channel-to-channel gain variations. Further, the nominal gain specification provided by the manufacturer can be off by 20% or more. This process modifies the variable gain stage of the currently selected processing channel, and refines its **Preamp Gain**. Due to the analog nature of the variable gain amplifier that is used, the precise analog gain following a hardware gain modification is unknown until it is measured. For this reason, calibration is an iterative process that must be executed any time acquisition values are changed that require a hardware gain modification, e.g. if the spectrum size is increased.

Section 4.3.6 describes how to use the **Gain Matching** tool to calibrate all processing channels simultaneously. Once calibration is completed we will save the modified configuration settings, each channel with its own unique **Preamplifier Gain**, to an INI file so that they will be retrieved the next time xManager is started.

4.3.5.1 Calibrating the Gain

At this point you should have an energy peak bounded by a region of interest. Please review section 4.3.4 if you have not created an ROI. The ROI table also displays the mean energy and width of the peak in the ROIs, as well as the ROI upper and lower limits. Either the arithmetic mean or the Gaussian mean can be displayed. Similarly the FWHM (full-width-half-maximum) is displayed either in kilo-electron-Volts or as a percentage of the mean energy. Simply click on the appropriate column heading to change the display.

Select the **Detector** settings tab and make note of the **Preamp Gain** value. To calibrate: First make sure the ROI containing the selected peak is “**Active**”, as set in the leftmost column and then enter the peak’s known energy into the “**Desired**” field of the ROI (i.e. for an ^{55}Fe K_{α} line enter 5.899 keV). .

Then press the **[Calibrate ROI]** button, then start a new run. The spectrum should now appear with the peak properly calibrated. For the best accuracy it is often necessary to run the calibration through a few iterations. If the initial spectrum was badly out of calibration, the resulting change in gain may cause the peak to jump partially or fully out of its ROI. In this case, readjust the ROI so that it centers on the peak before repeating the calibration.

4.3.5.2 Propagating the Nominal Gain

When you are satisfied with the calibration, note that the **Preamp Gain** value has been modified. Before we proceed with a system wide calibration it is good practice to propagate this nominal value to all the processing channels. Make sure the **Apply to All** checkbox is checked. Cut-and-paste, or simply re-type the **Preamp Gain** value and press **[Apply]**. Note that all channels have now been set to the new value.

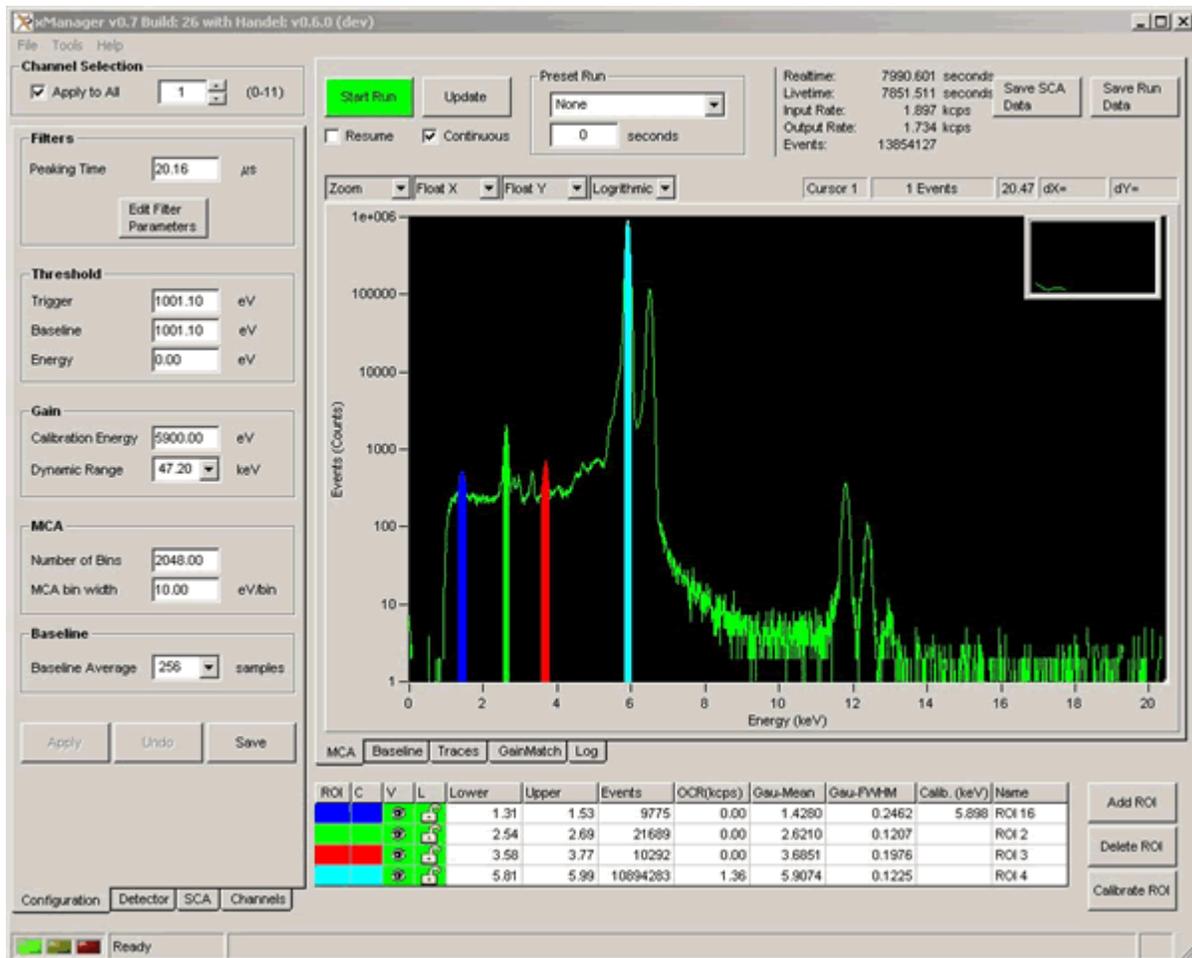


Figure 4.8: A calibrated ^{55}Fe spectrum with four regions of interest (ROIs).

4.3.6 System-Wide Gain Calibration

The **GainMatch** tool automates the gain calibration for all specified processing channels simultaneously, according to user constraints. Selected channels can be skipped. The iterative routine acquires data for the user-specified **Acquisition Time**, looks for a peak within the **Calibration Peak Range**, compares the measured peak centroid to the **Calibration Energy** and adjusts the gain as necessary. The process repeats until to the **Number of Iterations** has expired, or the **% of Calibration Energy** has been reached.

4.3.6.1 Skipping Channels

The **GainMatch** tool operates on all *active* processing channels. Individual processing channels can be disabled, or skipped, from the calibration macro. Select the **System** settings tab to display the **Channel Selection Detail**. Click in the **Skip Session** column to de-select individual channels.

Note: Channel skipping also applies to MCA data readout as described in section 4.3.2.

Channel	C	Skip Session	Run Status
0			Ready
1			Ready
2			Ready
3			Ready
4			Ready
5			Ready
6			Ready
7			Ready
8			Ready
9			Ready
10		Skipped	
11		Skipped	

Figure 4.9: The **Channel Selection Detail** control in the **System** settings tab. In the system shown, MCA data readout and system-wide gain matching would be skipped for channels 10 and 11.

4.3.6.2 Running the Calibration Macro

Select the **GainMatch** tab in the main window. Remember that this tool works best if the nominal gain has been verified as described in section 4.3.5 above: The algorithm is only stable if a peak is detected within the specified range. Depending on source intensity the **Acquisition Time** should typically range between 1 and 10 seconds. It is best to experiment with the settings to get a feel for the utility.

Note that if the calibration routine fails to find a peak in a given channel, that channel will automatically be disabled. You may want to re-enable channels afterwards in the **System** settings tab.

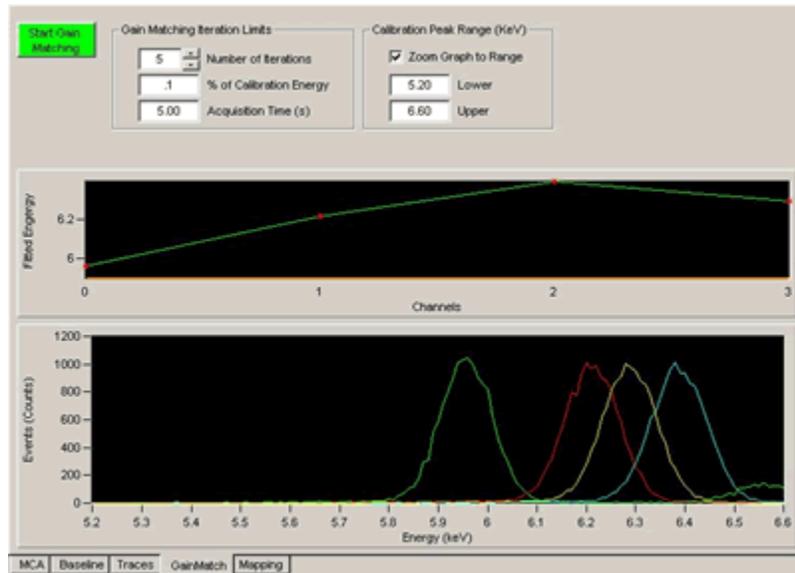


Figure 4.10: The GainMatch panel after one iteration. Note the significant channel-to-channel gain variation.

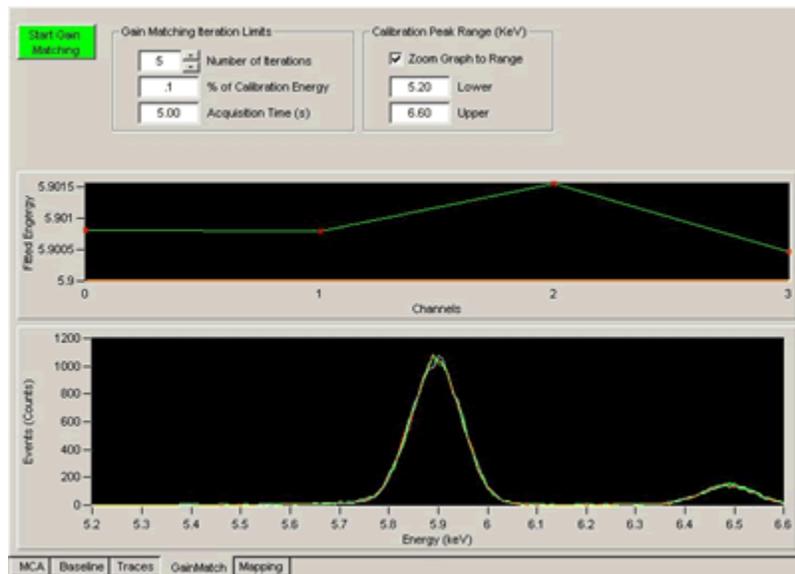


Figure 4.11: The GainMatch panel after five iterations. Note the near perfect channel-to-channel calibration.

Note that if energy calibration results in a significant change in gain, it may be necessary to adjust thresholds.

► Save your (modified) INI file to a unique filename: Select **Save Configuration As...** from the **File** menu.

4.3.7 Saving and Loading INI Files

Completion of the gain calibration is the final step in the verification of basic settings. The settings should now be saved to an INI file such that they will automatically reload whenever xManager is started. Because calibration is required any time spectrometer settings are changed, we recommend creating separate INI files for each commonly used peaking time. Once the settings for a peaking time have been optimized and the system calibrated, the entire setup can be restored by loading the INI file.

4.3.7.1 Saving an INI File

Select **File » Save Configuration As...** to open the **Save Configuration File** dialog. Enter a unique filename and press the **[Save]** button.

4.3.7.2 Loading an INI File

Select **File » Load Configuration...** to open the **Open Configuration File** dialog. Select a file and press the **[Open]** button.

4.3.7.3 Creating an INI File

The **Configuration Wizard** utility should be used to generate new INI files. Select **Tools » Configuration Wizard...** to open the **Save Configuration File** dialog. The utility allows for a previously used INI file to be used as a template. Please refer to section 3.2.

4.3.8 Output Statistics

Global statistics, such as ICR, OCR and deadtime fraction are displayed along the top of the main window. Statistics for defined regions of interest are displayed in the ROI table.

4.3.8.1 Real Time

This is simply the time elapsed between the Start Run and Stop Run operations, measured in the DSP itself every 8ms with 320 ns units. Intermediate values read out during the run will therefore have the lower accuracy, but the value reported at the end of the run will be accurate to 320ns.

4.3.8.2 Trigger Live Time

This is the measured live time of the Fast Trigger Filter, i.e. the time that the Fast Trigger Filter output stays below threshold. This measurement has units of 320ns, and is used to calculate the input count rate.

4.3.8.3 Energy Live Time

This DSP-level output is the *computed* live time of the Energy Filter. This value is also accurate to 320 ns at end of run but is only computed at 8 ms intervals during the run.

Note: The displayed Live Time does not express a relationship between the OCR and ICR. The Dead Time Percentage display does relate OCR to ICR.

4.3.8.4 Input Count Rate (ICR)

The measured input count rate (ICR) is displayed in units of thousands of counts per second [kcps]. The DSP applies internal correction procedures so that the measured ICR is very close to the true ICR, especially for longer peaking time settings. Please see section 0 for a discussion of this issue.

4.3.8.5 Output Count Rate (OCR)

The output count rate is also displayed in units of thousands of counts per second [kcps]. The OCR is simply the total number of detected events that did not pile up divided by the real time elapsed. Detected events that do not pile up, but whose measured energy falls outside the spectrum upper and lower limits, are called overflows and underflows, respectively. Both overflows and underflows are included in the OCR.

4.3.8.6 Dead Time %

The Dead Time Percentage (%) is calculated as the OCR divided by the ICR multiplied by the real time.

4.3.8.7 ROI Statistics

The peak centroid and width measurements are displayed in the ROI table. The centroid can be displayed as the arithmetic mean or the Gaussian mean. Similarly the peak width, or FWHM (Full Width Half Maximum), can be displayed in units of energy (eV), or as a fraction of the peak centroid energy. Click on the column heading to toggle the display. The number of events and the OCR are also displayed for each ROI.

4.3.9 Single Channel Analyzer (SCA)

An SCA window is similar to a region of interest (ROI): It is a user-defined range of the energy spectrum. While an ROI is a software-level construct used in analyzing a spectrum, an SCA window is processed at the hardware level. The DSP counts all events within the window and stores the result, the **SCA Counts**, in memory. SCA windowing thus allows for faster readout. Instead of reading the entire spectrum, it is only necessary to read a single word for each SCA. Select the **SCA** tab in the settings panel to view and modify SCAs.

4.3.9.1 Creating SCAs

SCA windows can be transferred from the ROI table. Simply select the desired ROI from the table (such that the ROI column reads "ACT"), then press the **[Insert Active ROI]** button. All defined ROIs can be transferred in one step by pressing the **[Insert All ROI]** button.

To manually define SCAs, first enter the number of SCA windows desired and press **[Apply]**. Notice that the corresponding number of rows is added to the SCA table. To define an SCA, all that is required is the **Low (keV)** and **High (keV)** bounds. These values can be entered manually.

4.3.9.2 Running with SCAs

Acquire data by starting and stopping an MCA run. Notice that when the run is complete, the SCA table is updated with the sum of events in the SCA window. In xManager the SCA function is included for demonstration purposes: The SCA data is redundant because xManager automatically reads out the entire spectrum in any case. SCAs are intended for users building applications on Handel, who desire very fast readout times. Please refer to the Handel User Guide for help reading out SCAs directly via the Handel device driver.

4.3.10 Saving and Loading Data

All data is stored in ASCII format and are easily readable using a text editor. The default format for MCA data includes bin scaling and other basic operating parameters and is date/time stamped. We would like to directly support as many formats as possible—please let us know if your format is not supported.

4.3.10.1 MCA Data

Spectra can be saved for later display or for analysis in another program. Acquire a spectrum, then select **Save MCA Data...** from the **File** menu. Saved MCA data can be displayed at any time. There are two methods of displaying saved MCA data:

- If you want to compare a saved spectrum to currently acquired spectrum, the saved data can be displayed as an overlay. First acquire a spectrum, then select **Open MCA Data...** from the **File** menu. Notice that the statistics and ROI table operate on the current data only. Note also that if new data is acquired the stored data is preserved. Zooming and panning operate by default on the current data. To zoom and pan the saved data right-click in the display area, select **Active Plot** and select the saved waveform. To clear the stored data right-click in the data display area and select **Remove Ref**.
- If you would like to analyze a stored spectrum, first right-click in the data display area and select **Clear Graph**. Now select **Open MCA Data...** from the **File** menu. Notice that the statistics, ROI table and graphical tools now operate on the stored data. Note also that if new data is acquired the stored data is automatically cleared.

To Save or Load a spectrum:

- Press the [**Save MCA Data**] button in the **MCA Window**, or
- Select **Save MCA Data...** or **Open MCA Data...** from the **File** menu.

4.3.10.2 Baseline Histogram Data

The baseline histogram can be saved for later display. Acquire a baseline histogram, then select **Save Baseline...** from the **File** menu. A saved baseline histogram can be displayed at any time. Again, there are two methods of displaying a saved baseline histogram:

- If you want to compare a saved baseline histogram to currently acquired data, the saved data can be displayed as an overlay. First acquire a baseline histogram, then select **Open Baseline...** from the **File** menu. Notice that the **Baseline Mean** and **Baseline FWHM** operate on the current data only. Note also that if new

data is acquired the stored data is preserved. Zooming and panning operate by default on the current data. To zoom and pan the saved data right-click in the display area, select **Active Plot** and select the saved waveform. To clear the stored data right-click in the data display area and select **Remove Ref**.

- If you would like to analyze a stored baseline histogram, first right-click in the data display area and select **Clear Graph**. Now select **Open Baseline...** from the **File** menu. Notice that the **Baseline Mean** and **Baseline FWHM** now operate on the stored data. Note also that if new data is acquired the stored data is automatically cleared.

4.3.10.3 Trace Data

Oscilloscope traces can be saved for later display. Acquire a trace, then select **Save Trace...** from the **File** menu. A saved trace can be displayed at any time. Again, there are two methods of displaying a saved baseline histogram:

- If you want to compare a saved trace to currently acquired data, the saved data can be displayed as an overlay. First acquire a trace, then select **Open Trace...** from the **File** menu. Note that if new data is acquired the stored data is preserved. Zooming and panning operate by default on the current data. To zoom and pan the saved data right-click in the display area, select **Active Plot** and select the saved waveform. To clear the stored data right-click in the data display area and select **Remove Ref**.
- If you would like to analyze a stored trace, first right-click in the data display area and select **Clear Graph**. Now select **Open Trace...** from the **File** menu. Notice that zooming, panning and cursor operations now operate on the stored data. Note also that if new data is acquired the stored data is automatically cleared.

4.3.10.4 DSP Parameters

The **DSP Parameters** panel allows for the export of displayed data to a file. Select **Tools»DSP Parameters...**, then press the [**Export to File...**] button.

4.4 Run Control

This section covers run synchronization. Automatic run termination is achieved by setting run Presets. Run start synchronization is achieved in multi-module systems using the LBUS hardware function. Runs can be disabled in realtime according to a user-provided logic signal using the GATE function. After stopping a run, the run can be resumed without first clearing the data.

Run Control settings are accessed in the MCA Panel – select a preset type from the drop-down list and enter a value to select the criteria for which the current run is stopped.

4.4.1 Run Presets (Automatic Run Termination)

Run presets determine the duration of the data acquisition run. The DXP-XMAP can end the run when a specified preset real or live time has elapsed, or when a specified number of events have been detected or processed.



Figure 4.12: The Run Preset settings in the MCA panel.

4.4.1.1 Run To Preset [choice1] Preset Value = choice2

Select the **MCA** tab in the **Main Window**, select a **Preset Run** type from the drop-down list select the criteria for which the current run is stopped:

- **None** - run ends when user presses “Stop” button.
- **Fixed Real Time** – run ends after specified real time elapses.
- **Fixed Energy Live Time** – run ends after specified live time elapses.
- **Fixed MCA Events** – run ends after specified number of valid events have been processed into the MCA.
- **Fixed Input Triggers** – run ends after specified number of input events have been detected.

Now enter a value in the box below, e.g. 30 seconds, or 1000 counts. Start a new data acquisition run and note that the run automatically stops when the run preset criteria have been met.

4.4.2 LBUS (Run Start Synchronization)

The LBUS function is used to synchronize the start time of modules in different PCI bus segments. Because a 'Start Run' command must be issued to each module, the actual start time would otherwise vary from module to module. A backplane PXI_TRIG line is used to automatically synchronize all modules in a PCI bus segment. The front panel LEMO connector, configured for LBUS functionality, extends this synchronization line between bus segments.

If more than one bus segment is populated we recommend using the LBUS function, otherwise it is unnecessary. Please review section 7.2.2 for a complete description of this feature, and run the Configuration Wizard again if you'd like to enable the LBUS function. If you decide to use the LBUS function the left-most module in each PCI bus segment will be designated as the LBUS master, i.e. it accepts the front-panel LBUS connection.

4.4.3 The GATE Function

The external logic (LEMO) input can be configured to halt data acquisition simultaneously on all DXP channels according to an external logic signal. The so-called GATE function supports TTL/CMOS levels. Please review section 7.2.3 for a complete description of this feature, and run the Configuration Wizard again if you'd like to enable the GATE function.

4.4.4 Resume Run: Clear or Retain MCA Data

When a new run is started the data from the previous run can either be cleared or retained. This setting is accessed in the **MCA** window via the

Resume checkbox. If the box is checked the **[Start/Stop]** button has start/pause functionality: new data accumulates with data from the previous run. If the box is unchecked, data from the previous run is first cleared.

4.5 Display Controls

This section briefly describes tools and options related to the display of acquired data.

4.5.1 MCA Auto Update / Refresh Rate

The **MCA** display can be set to refresh with a user specified interval. If the **Continuous** box is checked, xManager will automatically refresh the display. To set the refresh interval, select **Options...** from the **Tools** menu, and select the **Misc** tab. Enter the desired interval in milliseconds. If the **Continuous** box is unchecked the display is updated only when the **[Update]** button is pressed or when the run ends.

4.5.2 Graphical Display Tools

xManager features a wide array of display controls, some or all of which are accessible in each graphics window (i.e. **MCA**, **Baseline** and **Trace** panels). Most of these controls can be accessed by right-clicking, as shown in the figures below. These tools are intuitive and redundant.

4.5.2.1 Basic Tools and Options

Right-click in the Main window to view the basic display tools and options. Some of these items are also accessible in the drop-down list items above the plot area.

- **Full Scale** – Scales both axes such that all data is displayed.
- **Set to Logarithmic (Set to Linear)** – Sets the y-axis to logarithmic or linear scale.
- **Clear Graph** – Clears all data and the cursors.
- **Default Mode** – Sets the default left-click action. This can be set to Zoom, Create ROI or Pan-on-click
- **Place Cursor 1** – Select to place cursor #1. Notice that the data point selected by the active cursor is displayed above the plot area.
- **Place Cursor 2** – Select to place cursor #2. Notice that delta-x and delta-y values relative to cursor #1 are displayed above the plot.
- **Remove Ref** – Clears a reference waveform (typically loaded from a file) from the display.
- **Double-Click** – Sets the default double-left-click action.
- **Active Plot** – Selects either the current data or data loaded from file to be active for zooming and panning.

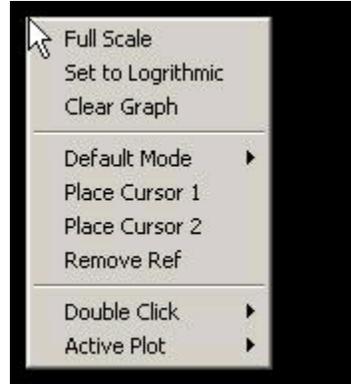


Figure 4.13: Right-click in the Main display window to view the basic display tools and options.

Alternatively, a default mode, e.g. zoom, ROI, or pan, can be selected using the drop-down list menus above the plot area. The default mode applies when you left-click in the display area. Left-click on either axis to pan. Most of the tools are intuitive and redundant.

4.5.2.2 Cursor Tools and Options

After placing a cursor, right-click on the cursor itself to view the cursor tools and options.

- **Auto ROI** (MCA panel only) – If the cursor was placed on an energy peak this will automatically set the bounds of the active ROI to enclose the peak.
- **Crosshair** – The default cursor is a vertical line. This option sets the cursor to an x-y crosshair.
- **Delete** – Deletes the cursor.

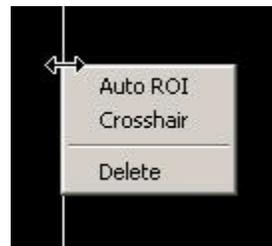


Figure 4.14: After placing a cursor, right-click on the cursor itself to view the cursor tools and options.

4.5.2.3 Axis Tools and Options

Left-click and drag on an axis to pan. Right-click on either the x or y axis to view the axis tools and options. Some of these items are also accessible in the drop-down list items above the plot area.

- **Y Axis Mode** – *Fixed* prevents the y-axis from re-scaling according to new data or user zoom or pan operations. *Floating* prevents the y-axis from re-scaling according to new data but allows user to zoom and pan. *Auto* sets the y-axis dynamically such that input data is fully displayed, but can be

overridden by user zoom or pan operations. *Fixed Auto* sets the y-axis dynamically such that input data is fully displayed, and disables user zoom or pan operations.

- **X Axis Mode** – *Fixed* prevents the x-axis from re-scaling according to new data or user zoom or pan operations. *Floating* prevents the x-axis from re-scaling according to new data but allows user to zoom and pan. *Auto* sets the x-axis dynamically such that input data is fully displayed, but can be overridden by user zoom or pan operations. *Fixed Auto* sets the x-axis dynamically such that input data is fully displayed, and disables user zoom or pan operations.
- **Log Scale** – Sets the y-axis to logarithmic or linear scale.

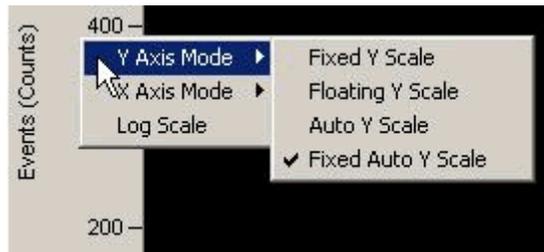


Figure 4.15: Right-click on either the x or y axis to view the axis tools and options.

Alternatively, axis modes can be selected using the drop-down list items above the plot area.

4.6 Optimizations

This section describes various parameter optimizations for the best performance in throughput, pileup rejection and energy resolution.

4.6.1 Throughput (OCR)

The OCR depends only on the ICR and the dead time per event τ_d :

$$\text{OCR} = \text{ICR}_t * \exp - (\text{ICR}_t \tau_d), \text{ where } \tau_d = 2*(t_p + t_g).$$

To increase the OCR at a given ICR, the dead time per event must be reduced. The obvious first step is to reduce the energy peaking time t_p . Further improvements can be made by reducing the gap time t_g .

4.6.1.1 Peaking Time (Energy Filter)

The peaking time is the energy filter length, or integration time, i.e. the ramping interval of the trapezoid. It is the primary setting in determining the balance between count rate performance and energy resolution.

Making a plot of energy resolution versus peaking time provides a useful future reference.

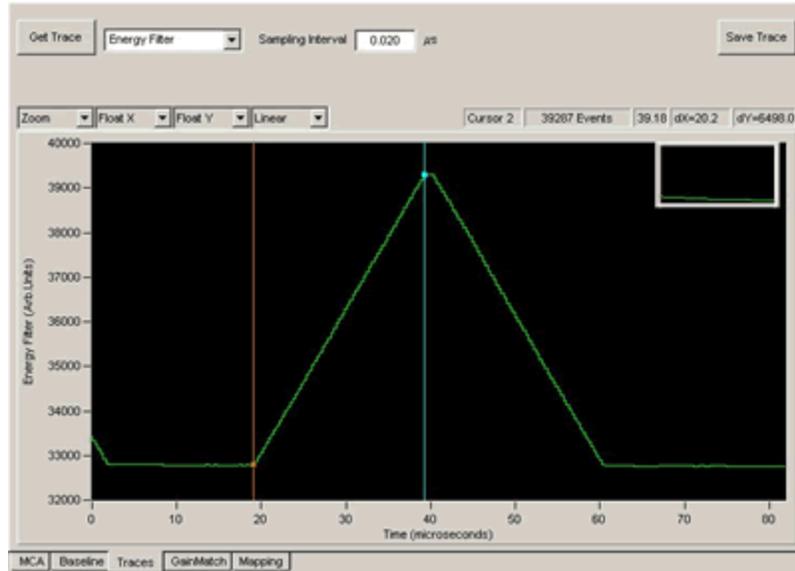


Figure 4.16: The output response of the Energy Filter with peaking time (ramping time) = $20.16\mu\text{s}$ and gap time (flattop time) = $0.96\mu\text{s}$. The trapezoid is the response to an x-ray.

You will generally find it useful, after making a first attempt to optimize settings, to capture a set of spectra over a wide range of peaking times, preferably over the full range that the DXP-XMAP supports and generate a plot of energy resolution versus peaking time. This will serve two purposes: first to serve as a standard of comparison, so that you can tell if further parameter adjustments are helping or not; and second, to provide you with some feedback about whether your spectroscopy system is behaving properly. Later, when everything is optimized and all the noise sources have been suppressed, you can go back and repeat these measurements to provide hard data for use in selecting the best peaking time for a given input count rate.

*To change advanced acquisition settings such as the **Minimum Gap Time**: Press the [Edit Filter Parameters] button in the*

*Reducing the **gap time** can significantly increase the data throughput at a given peaking time.*

4.6.1.2 Gap Time (Energy Filter)

The gap time of the energy filter sets the flattop length of the output trapezoid. Because the gap time directly affects the dead time, it is advantageous to set the gap time as short as possible. The gap time is subject to several constraints.

Generally the gap time should be set to a value that exceeds the 0 – 100 % preamplifier risetime in response to a detected x-ray. As long as this constraint is met, the trapezoid peak is tolerant of variations of the x-ray arrival time relative to the ADC clock. The digital filter architecture further constrains the gap time to an integer between 3 and 64 decimated clock intervals. In xManager, the user sets the **Minimum Gap Time** slightly larger than the measured preamplifier risetime, and xManager automatically maintains the gap time based on the decimation-dependent filter constraints. Please refer to section 4.7.1.2 for details on using xManager to measure the risetime for your system, and section 6.3.2 for a discussion of decimation and decimated clock periods.

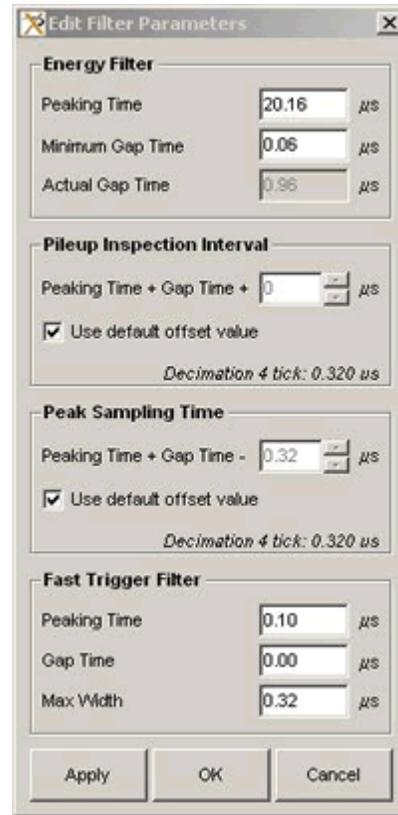


Figure 4.17: The **Edit Filter Parameters** panel.

To edit the **Minimum Gap Time** time, press the **Edit Filter Parameters** button in the **Configuration** settings panel to open a dialog. Enter the desired value for **Minimum Gap Time** and press **[OK]**.

Normally the **Minimum Gap Time** should be set to a value that exceeds the preamplifier risetime in response to a detected x-ray, however, there is one exception. At very high count rates, where resolution is less of a concern, it can be advantageous to set the **Minimum Gap Time** to a smaller value, even to zero. This setting will only have an effect for decimation 0, i.e. for peaking times less than 0.50µs. For other decimations the gap time will be set to the minimum value of 3 decimated clock cycles. Note that you may have to adjust other settings as a result:

1. **Peak Sample Offset** – Because you are reducing or even eliminating the flattop section of the trapezoid, performance becomes more sensitive to the energy sampling time. Refer to section 4.6.3.6.
2. **Gain Calibration** – A consequence of setting the gap time less than the preamplifier risetime is ballistic deficit: The peak value of the trapezoid is reduced. As a result you will almost certainly have to increase the gain after the gap time has been changed. See sections 4.3.5 and 4.3.6.

4.6.2 Pileup Rejection

Pileup inspection is described in detail in section 6.8. These settings should only be modified by users with a good understanding of the principles of pileup inspection.

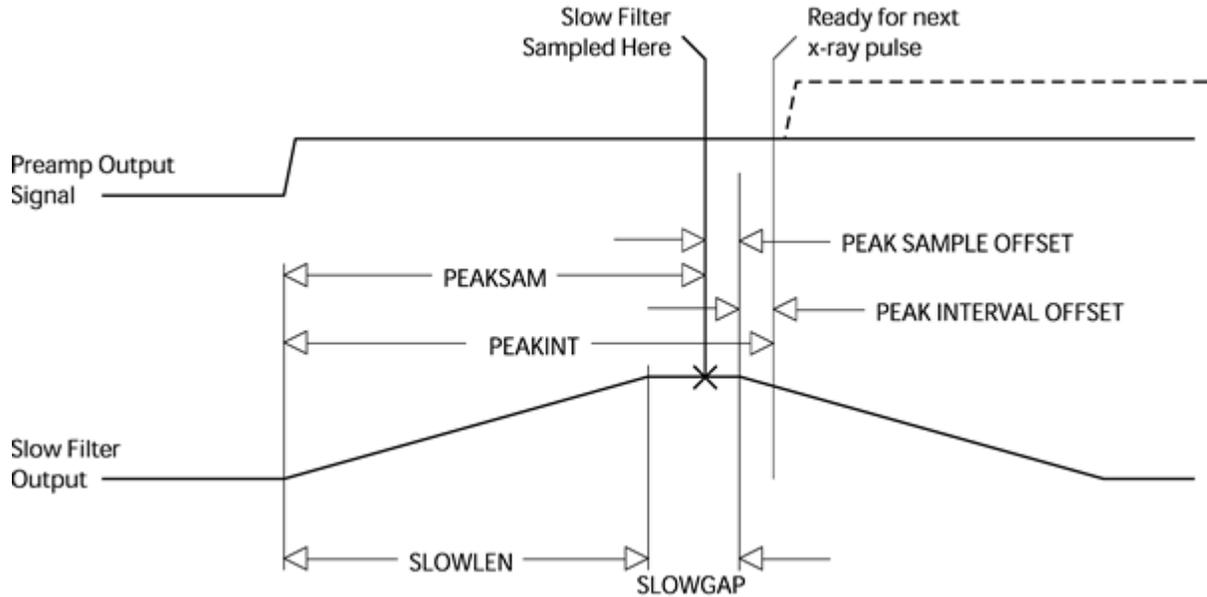


Figure 4.18: Slow, or Energy, filter output waveform diagram.

CAUTION: When set too low, the **MAXWIDTH** criterion can reject non-piled-up x-rays, resulting in attenuation at higher energies.

4.6.2.1 Maximum Width Constraint

The DSP parameter **MAXWIDTH** sets the maximum acceptable time that the fast trigger output can stay above threshold for a single event. Properly set, this constraint detects fast pileup (event separation on the order of 100ns). See section 6.8 for more information. The **Max Width** setting is accessible in the **Edit Filter Parameters** panel. By default it is set to 400ns, allowing for a preamplifier risetime up to 200ns. **MAXWIDTH** should be *at least* twice the fast peaking time plus the preamplifier's 1% settling time:

$$\text{Max Width} > 2 * \text{Fast Peaking Time} + \text{Preamp Risetime}$$

4.6.2.2 Peak Interval

The DSP parameter **PEAKINT** sets the minimum acceptable time that the slow energy filter needs to process a single event, i.e. the interval between peaks that can be properly sampled. This constraint detects slow pileup (event separation on the order of the energy peaking time). See section 6.8 for more information.

The optimum peak interval is usually fixed relative to the sum of the peaking time and gap time. The **Peak Interval Offset** is measured forwards in time from this sum, i.e. measured forwards from the end of the flattop period (see **Figure 4.18**):

$$\text{Peak Interval} = \text{Peaking Time} + \text{Gap Time} + \text{Peak Interval Offset}$$

The **Peak Interval** setting is accessible in the **Edit Filter Parameters** panel. In most cases it should be left at zero. Larger values will result in a more conservative pileup inspection at the cost of increased deadtime-per-event.

4.6.2.3 Reducing the Fast Peaking Time

At very high rates the fast filter peaking time may be reduced, to maintain good pileup inspection.

The default fast peaking time of 100 ns should be used in most cases. Generally speaking, a longer fast filter peaking time produces a lower pileup inspection threshold at the cost of a longer pileup inspection time interval. Little if any real benefit is derived from *increasing* the fast peaking time unless the preamplifier signal is extremely noisy. For good pileup rejection, the fast filter peaking time should be much shorter than the energy filter, which becomes a problem when the shortest energy filter peaking times are used. In these cases, some improvements in pileup rejection may be possible if the fast filter peaking time is *reduced*, e.g. to 60ns. We don't recommend using a fast gap time other than zero.

Open the **Edit Filter Parameters** panel and enter a new value for the **Fast Trigger Filter Peaking Time** and press [**Apply**]. Note that you may have to adjust other settings as a result:

1. **Trigger Threshold** – Because of the zero gap time, the **Fast Trigger Filter** normally produces some ballistic deficit. Reducing the trigger peaking time can heighten this effect. For best results the threshold be checked, as described in section 4.3.3.2.
2. **Max Width** – The time over threshold is directly related to the filter length. If you previously optimized **Max Width**, i.e. the maximum time over threshold, you may need to re-optimize. See section 4.6.2.1.

4.6.3 Energy Resolution

There are many possible reasons for poor energy resolution. This section points to the most common issues.

4.6.3.1 Proper Peaking Time Selection

The first step in improving energy resolution is, of course, to optimize the **Peaking Time**. Use your plot of energy resolution versus peaking time to select a peaking time where you get good energy resolution before making these adjustments.

4.6.3.2 Baseline Acquisition

Capturing good baseline values and proper averaging are vital to achieving good energy resolution. The **Baseline Threshold** and **Baseline Average Samples** settings must be set properly for a given peaking time. See section 4.7.2 for making adjustments in xManager, and section 6.4 for more detailed explanations of baseline issues.

4.6.3.3 Eliminate Noise Pickup

Noise pickup can destroy performance. It is very important to identify and eliminate excess noise in the hardware. Typically this involves eliminating ground loops, removing switching power supplies in close proximity and

improving shielding. Please refer to section 4.7.1 below for a brief introduction to using the **Trace** panel to identify noise issues.

4.6.3.4 Sufficient Gain to Sample Noise

If the signal gain is such that noise is not properly digitized at the ADC, energy resolution will not be optimal. This would result from a **Preamp Gain** setting that is too high (resulting in a DXP-XMAP variable gain setting that is too low). Set the gain so that the noise is sufficiently digitized – see section 4.7.1.1.

4.6.3.5 Sufficient Gap Time

If the gap time is too short, the trapezoid peak sample (the energy sample) becomes dependent on the arrival time of the x-ray relative to the ADC clock. Make sure that the **Minimum Gap Time** is longer than the preamplifier risetime as described above in section 4.6.1.2.

4.6.3.6 Peak Sampling Time

The optimum sampling time of the energy filter is usually fixed relative to the sum of the peaking time and gap time. The **Peak Sample Offset** is measured backwards in time from this sum, i.e. measured backwards from the end of the flattop period (see **Figure 4.18**):

$$\text{Sampling Time} = \text{Peaking Time} + \text{Gap Time} - \text{Peak Sample Offset}$$

The **Peak Sample Offset** setting is accessible in the **Edit Filter Parameters** panel. In most cases it should not be edited. An exception is when running with a very short gap time at decimation 0 (see section 4.6.1.2 above). In this case the **Peak Sample Offset** should be reduced empirically. Please see section 6.6.2 for a full discussion before attempting this procedure.

4.7 Diagnostics

The DXP-XMAP and xManager provide several diagnostic tools for identifying and resolving functional and performance issues.

*To open the **Traces** panel, click on the **Traces** tab in the main window.*

4.7.1 The Traces Panel (Oscilloscope)

The Traces Panel displays waveforms captured at various stages of the DXP's digital filter, such as the preamplifier signal as seen at the DXP-XMAP's analog-to-digital-converter (ADC). Each 16-bit output is plotted over 4096 sample points, with a user settable sampling interval. It can be a useful diagnostic tool for checking preamplifier polarity and gain, measuring the risetime, and for tracking down noise pickup and baseline irregularities. Note that signed waveforms are shifted such that zero is displayed at the midpoint value 32,768.

To acquire and view a waveform:

1. Click the drop down list to select the desired waveform:
 - **ADC** – The digitized (unsigned) ADC waveform, i.e. the input signal to the digital filters.

- **Baseline History** – The (signed) baseline running average. These are the actual values that are subtracted from the energy filter.
 - **Trigger Filter** – The (signed) raw fast filter output.
 - **Baseline Filter** – The (signed) raw intermediate filter output. For decimation 0 this is the same as the Energy filter.
 - **Energy Filter** – The (signed) raw slow filter output.
 - **Baseline Samples** – The (signed) intermediate filter output baseline samples selected for use in the running average.
 - **Energy Samples** – The (unsigned) slow filter output peak samples selected for the energy spectrum.
2. Enter a value ranging 0.020 - 1000 microseconds in the Sampling Interval field. This is the time between each displayed sample.
 3. Press **[Get Trace]**.

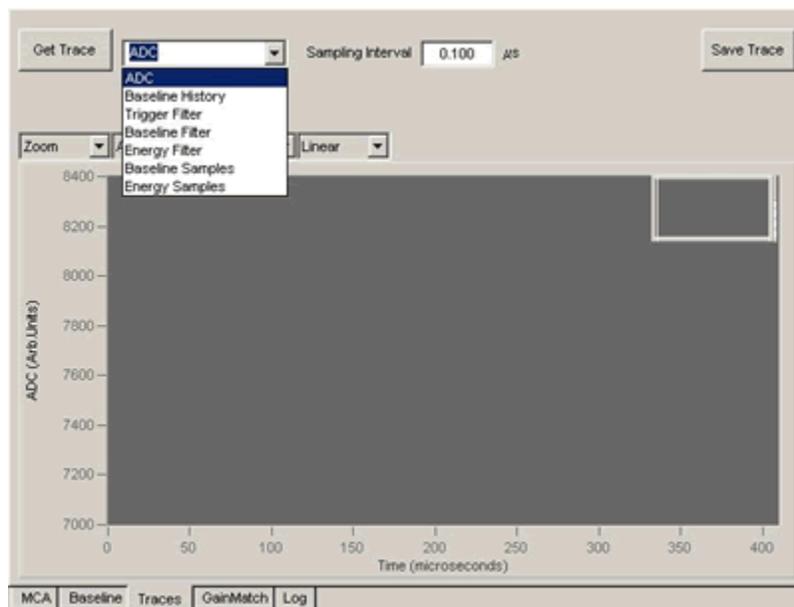


Figure 4.19: The Traces panel is a useful diagnostic tool.

*The preamplifier **Gain** and **Polarity** settings are accessed in the **Detector** tab of the **Settings** panel:*

4.7.1.1 Determining the Preamplifier Polarity and Gain

A common configuration error involves setting either the preamplifier signal polarity or gain incorrectly. Note: The preamplifier type, i.e. pulsed-reset or RC-feedback, is determined by the firmware file that is downloaded to the hardware (see section 3.2.1)

The Preamplifier Polarity configuration setting determines whether the ADC code is inverted prior to the digital filter pipeline, which expects x-ray pulses with a rising edge. The Trace panel displays the digital signal after this optional inversion. If the x-ray pulses are displayed with a falling edge, as shown in **Figure 4.20**, then the polarity setting is incorrect; if pulses are displayed with a rising edge, as in **Figure 4.21**, then the polarity setting is correct.

The **Preamp Gain** setting in the **Detector** panel, in combination with the **Dynamic Range** setting in the **Configuration** panel, determines the DXP-

XMAP analog variable gain setting. The variable gain is set such that an x-ray with energy equal to the dynamic range value produces a voltage step of the maximum allowable amplitude at the ADC input. X-rays with energies exceeding the dynamic range value cannot be processed correctly. The **Dynamic Range** setting should thus be set above the largest x-ray energy present in the system.

In order to get the best energy resolution the gain must be set such that electronic noise is digitized sufficiently that it can be properly filtered. In practice this means that the noise should span 20 or more vertical units in the display. In **Figure 4.20** the noise is contained in less than 10 displayed vertical units, indicating that the hardware gain setting is too low. This could be due either to a **Preamp Gain** configuration setting that is too high or to the **Dynamic Range** being set too large. The noise displayed in **Figure 4.21** spans approximately 40 vertical display units, indicating that the Preamplifier Gain is set correctly and the spectrum is properly sized.

For best results the noise should span 20 or more vertical units in the Oscilloscope Panel display.

To adjust your own system, first select the **Trace** tab in the main window and acquire a few traces, until you have recognizable x-ray events displayed. Compare the polarity and noise amplitude to the figures. If necessary, change the **Polarity** and **Preamp Gain**. You may also need to modify the **Dynamic Range**.

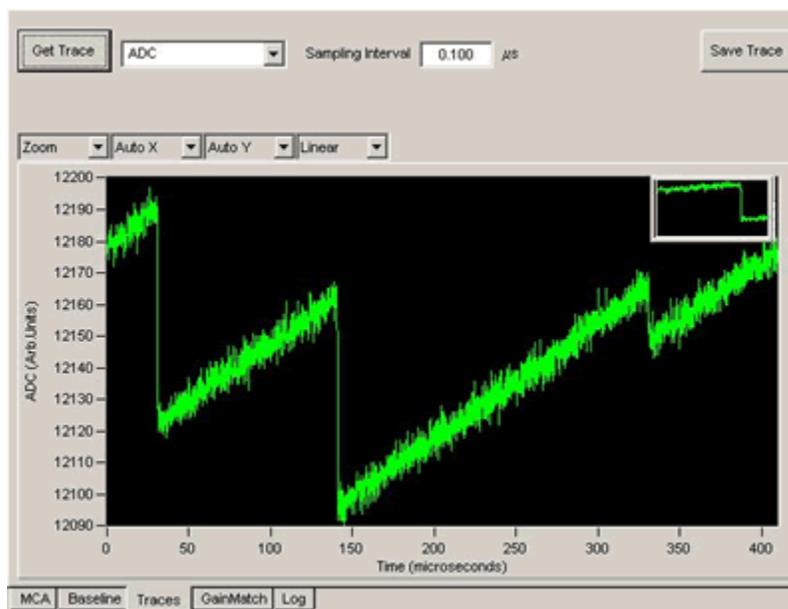


Figure 4.20: An ADC trace of a reset-type detector with the DXP-XMAP configured with the wrong polarity and a gain setting that is too low. X-ray steps displayed in this panel should have a rising edge, and noise should span 20 or more vertical units.

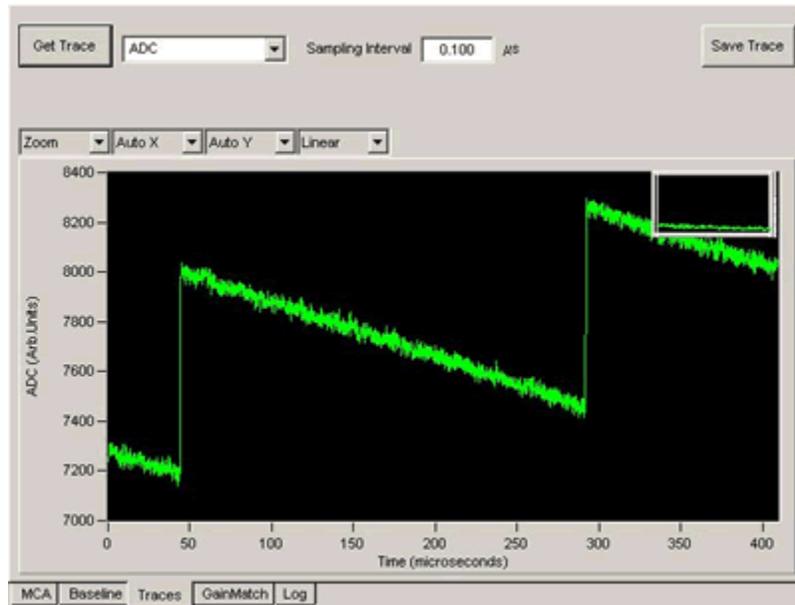


Figure 4.21: An ADC trace with correct polarity and a typical gain. Note that the noise is well digitized at roughly 40 vertical units.

4.7.1.2 Measuring the Preamplifier Risetime

The **Trace** panel is useful for measuring the preamplifier signal risetime, which should be done before modifying the **Minimum Gap Time** as described in section 4.6.1.2.

As mentioned earlier, the minimum sampling interval in the display is 20ns—the actual ADC sampling period. Acquire an ADC trace at the minimum sampling interval of 0.020μs that includes at least one well separated x-ray event. Use the zoom tool (accessed via the right click menu or through the display controls at the graph's upper left) to expand the horizontal axis about the selected event. Place **Cursor 1**, by right-clicking in the display area and selecting "**Place Cursor 1**", immediately before the x-ray pulse. Similarly, place **Cursor 2** immediately after the signal has settled following the pulse. The **dX** field of the cursor data area in the upper right hand corner should now display the 0 – 100% preamplifier risetime in μs.

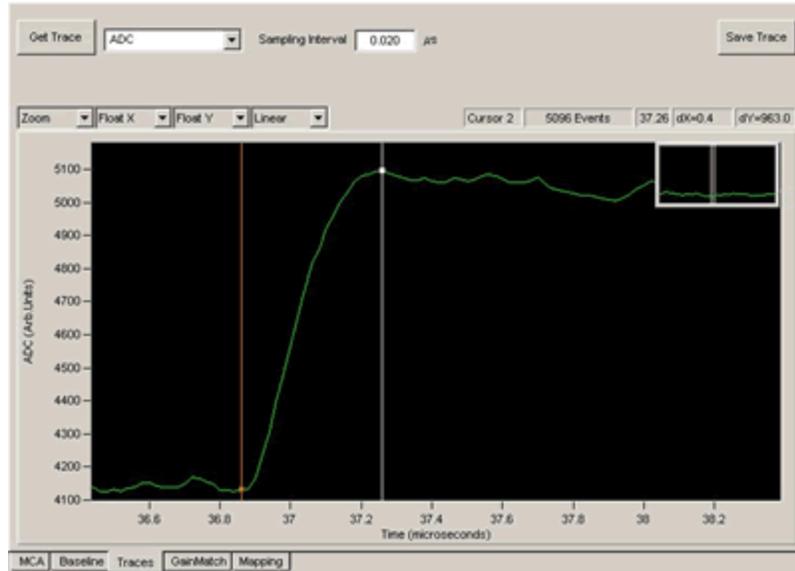


Figure 4.22: Use the zoom function and cursors to measure the preamplifier risetime. The risetime is approximately 400ns.

4.7.1.3 Measuring the RC Decay Time τ (RC-Feedback Preamplifiers only)

The **Trace** panel is also useful for measuring the decay time for RC-feedback preamplifiers.

Acquire an ADC trace that includes at least one well separated x-ray event. Use the zoom tool (accessed via the right click menu or through the display controls at the graph's upper left) if necessary to expand the horizontal axis about the selected event *such that the entire decay time is displayed*. Place **Cursor 1**, by right-clicking in the display area and selecting “**Place Cursor 1**”, at the peak value of the x-ray pulse. Similarly, place **Cursor 2** immediately before the x-ray pulse such that a baseline value is selected. Record the **dY** value from the cursor data display—this is the pulse height. Now move **Cursor 1** to the point on the decay curve that produces a new dY value that is 1/e times the measured pulse height:

$$dY' = (1/e) \cdot dY \sim 0.37 \cdot dY$$

The cursors should now be separated by the time constant τ , displayed in μs in the **dX** field.

4.7.1.4 Optimizing the Baseline Average Length

The **Trace** panel is also useful for optimizing the number of samples in the baseline average. Please first review section 6.4 for a thorough discussion of baseline acquisition.

The **Baseline Average Length** refers to the number of samples of the **Baseline Filter** raw output in the running average of baseline samples, which should be:

- Large enough to average out electronic noise at the higher frequencies.

- Small enough to track low frequency fluctuations in the front end, i.e. detector dark current.

The precise noise distribution and the low-frequency signal characteristics of the detector and preamplifier together yield an optimum number of samples in the average. Too small a value will not allow for proper filtering of electronic noise. Too large a value will not allow proper tracking of low frequency signals, e.g. due to EMI, that can be cancelled out with double correlation.

First select **Baseline Filter** from the drop-down list, set the **Sampling Interval** to 1.000 μ s and press [**Get Trace**] to view the raw output of the baseline filter.

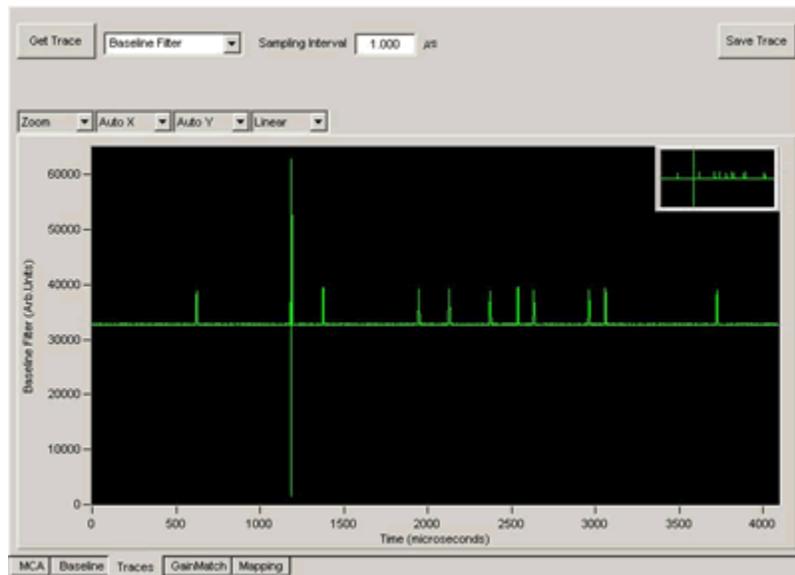


Figure 4.23: The raw output of the **Baseline Filter**. Approximately 10 x-ray events are visible as trapezoid, as well as a preamplifier reset spike. The goal of baseline acquisition is to sample the baseline noise.

Now select **Baseline History** from the drop-down list, set the **Sampling Interval** to 10.000 μ s and press [**Get Trace**] to view the baseline running average. You want to achieve a waveform similar to that shown in **Figure 4.25**, where noise is filtered out but the average still tracks real variations. If you see something more like **Figure 4.24** or **Figure 4.26**, adjust the **Baseline Average Samples** setting in the **Configuration** panel and press [**Apply**]. Acquire another trace and adjust as necessary. In most cases the values 128 and 256 yield the best results. At high rates it may be advantageous to reduce the number of samples as low as 16. For near perfect preamplifiers the average can be increased to 1024 or more. In any case the optimization is not complete until you acquire a spectrum and verify the energy resolution has improved.

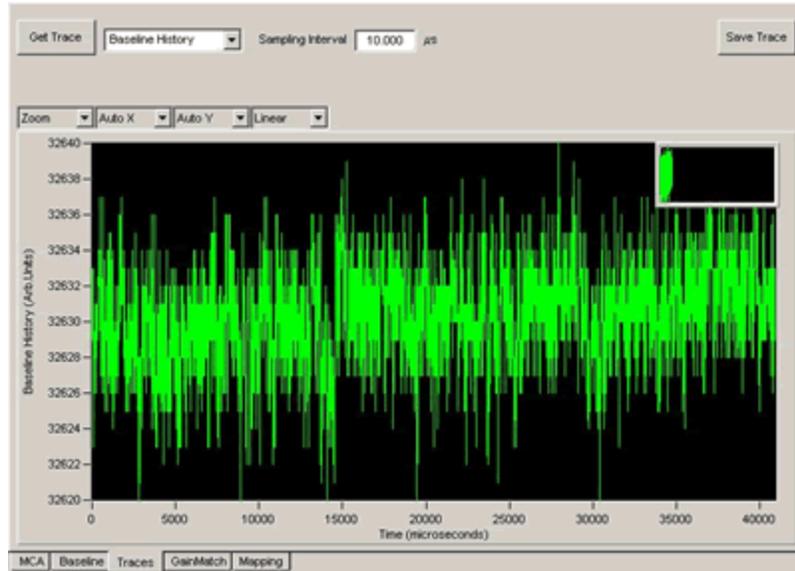


Figure 4.24: The baseline average with the number of samples set too low (16 samples in the average). Notice that there is still a lot of noise, as well as some real variations in the baseline.

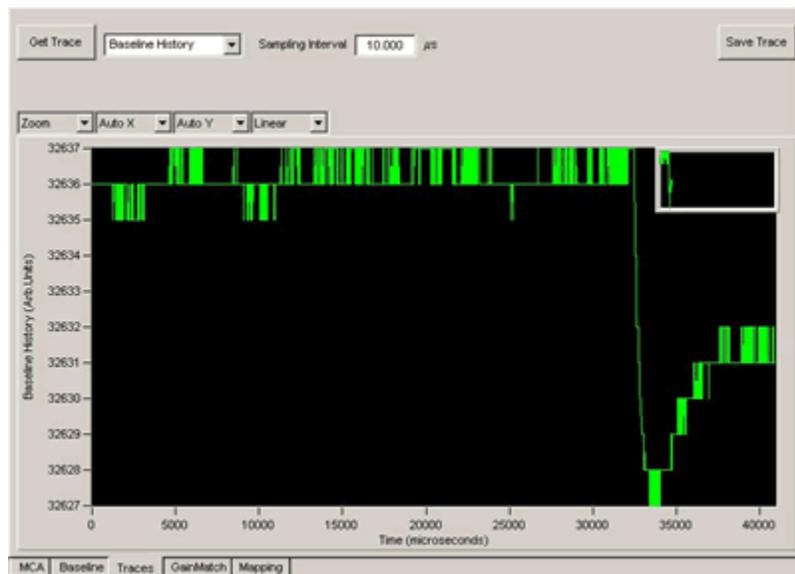


Figure 4.25: The baseline average with the number of samples set properly (256 samples in the average). Notice that there is virtually no noise, but that real baseline variations are tracked. In this case the downward variation is due to some curvature in the preamplifier output following a reset.

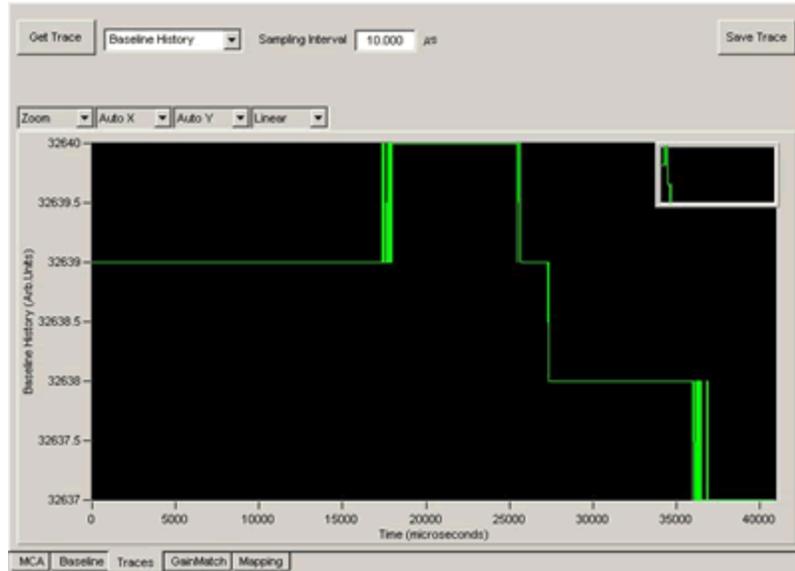


Figure 4.26: The baseline average with the number of samples set too high (4096 samples in the average). Notice that although there is no noise, the average lags behind a real downward variation in the baseline.

*To open the **Baseline** panel, click on the **Baseline** tab in the main window.*

4.7.2 The Baseline Panel

Baseline measurements are continually updated samplings of the output of the energy filter when no event is being processed. A running average of these baseline samples is then made to reduce the noise in this measurement and the result is subtracted from instantaneous raw pulse-height measurements to determine their true amplitudes. Please first review section 6.4 for a thorough discussion of baseline acquisition.

The **Baseline** panel displays a histogram of the instantaneous baseline samples. The histogram, in combination with the **Baseline History** and **Baseline Filter** traces (see section 4.7.1.4 above) are powerful tools for diagnosing electronic noise and common nonlinearities in the detector and preamplifier. Select the **Baseline** tab and press [**Get Baseline**] to acquire a baseline histogram. You should see a Gaussian peak with few, if any, outliers, as in **Figure 4.27**. If there are many outliers to the right of the peak, as in **Figure 4.28**, the threshold is set too high. If the right side of the peak is attenuated, non-Gaussian as in **Figure 4.29**, the threshold is set too low.

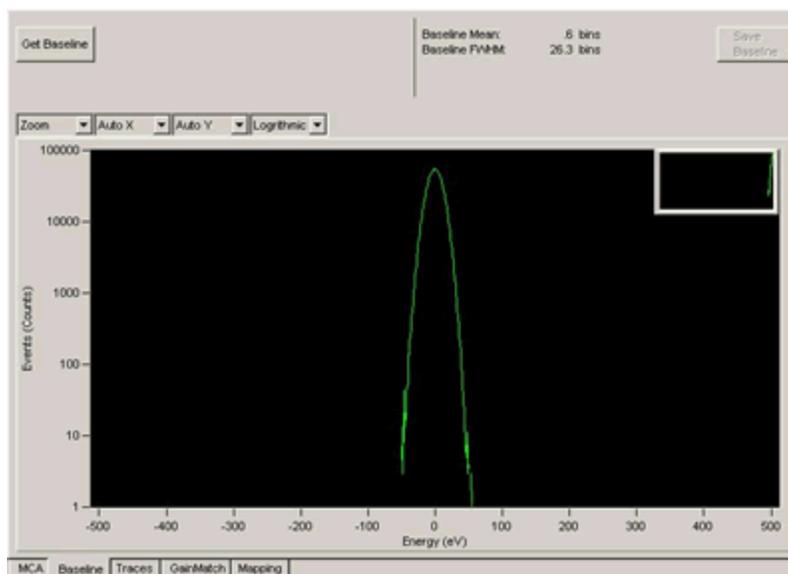


Figure 4.27: A good baseline histogram: the shape of the noise peak is Gaussian with no outlying data points.

4.7.2.1 The Baseline Threshold

Section 4.3.2 includes a discussion about setting thresholds based on visual feedback in the **MCA** panel. Threshold settings also affect baseline acquisition: Baseline acquisition is enabled only when the **Fast Trigger Filter** output is below **Trigger Threshold** and the **Baseline Filter** output is below the **Baseline Threshold**. It is assumed that the **Trigger Threshold** is set conservatively, so that baseline acquisition is dominated by the **Baseline Threshold**. Note: The baseline threshold is not available for decimation 0, i.e. peaking times less than or equal to 500 ns.

Edit the **Baseline Threshold** value in the **Configuration** tab and press **[Apply]**. Typical values range from 150 eV to 1000 eV. The baseline filter length is linked to the energy filter length, or peaking time, thus the baseline threshold should be optimized every time the peaking time is changed. All thresholds must be readjusted if the gain changes significantly. For this reason it is useful to save INI files for commonly used peaking times after optimizations are complete.

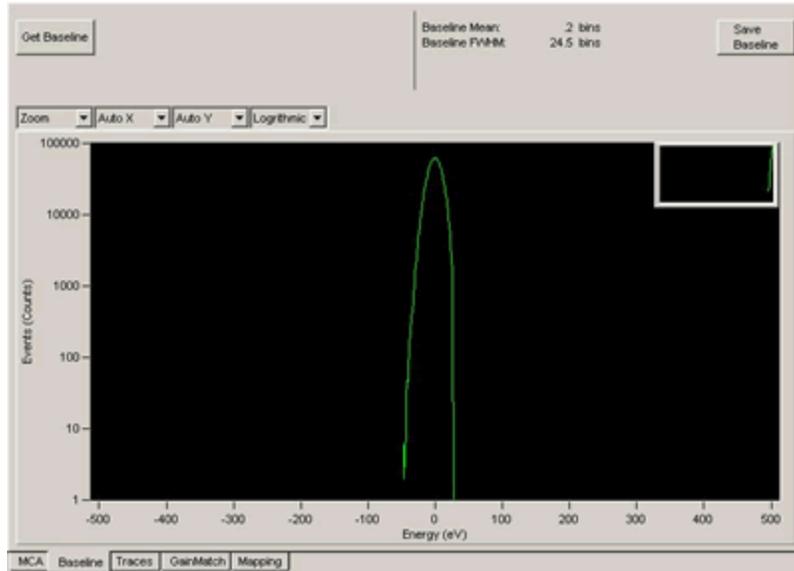


Figure 4.28: A baseline histogram with the threshold too low. Notice that the right side of the noise peak is attenuated. The rest of the noise peak will show up in the energy spectrum.

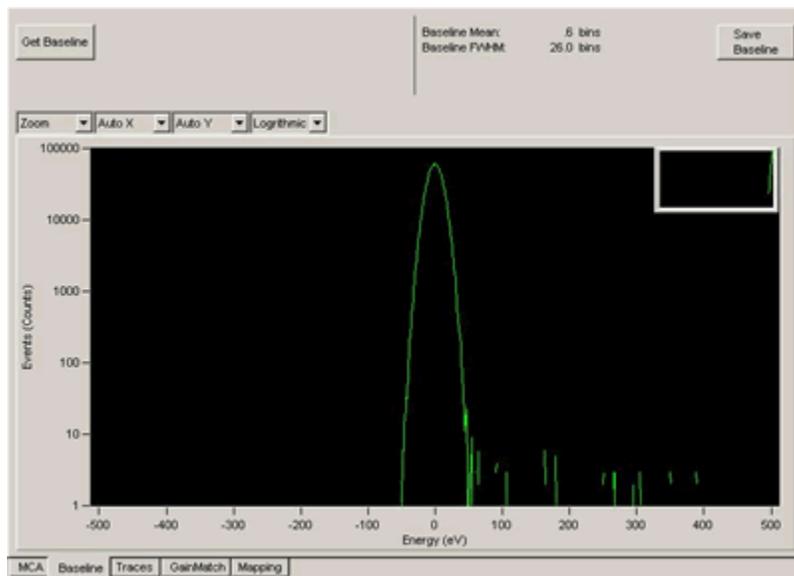


Figure 4.29: A baseline histogram with the threshold too high. The baseline samples to the right of the noise peak are partial energy events that should be in the energy spectrum.

Warning: Changing DSP parameters without understanding them is discouraged..

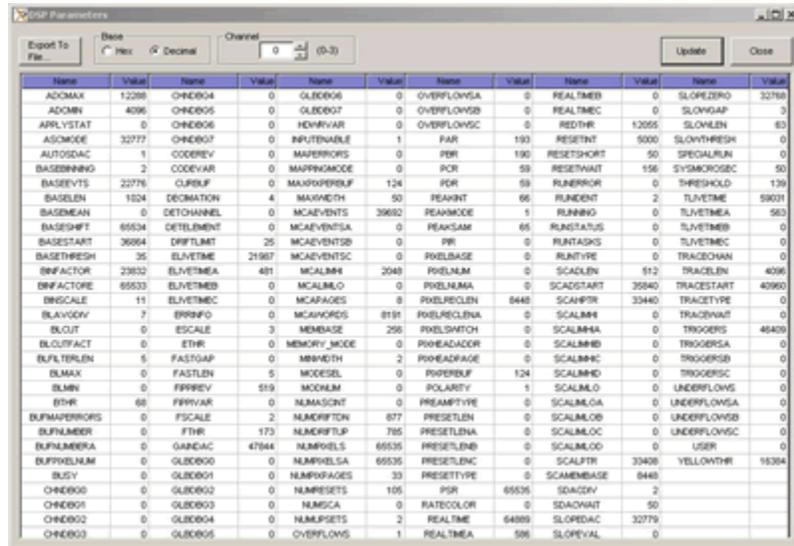
4.7.3 DSP Parameters

The **DSP Parameters** panel, accessible via the **Tools** menu, provides a diagnostic display of all DSP's internal parameters for the currently selected processing channel. The **Hex** and **Decimal** radio buttons determine whether parameters are displayed in hexadecimal or decimal format. Press the **[Update]** button to refresh the display. Note that various active parameters will change

every time the **[Update]** button is pressed. The **Channel** selector immediately loads the new processing channels parameters.

4.7.3.1 Generating a Diagnostic DSP Parameters File

The **[Export to File...]** button generates a ASCII text file containing all of the parameters for the currently selected processing channel. You may be asked to generated this file by technical support. More likely you will be asked to generate a full **Error Report**, as described in section .



Name	Value	Name	Value	Name	Value	Name	Value	Name	Value	Name	Value
ADCMAR	1208	CH0004	0	GL0006	0	OVERFLOWA	0	REALTIME	0	SLOPEZ0	32768
ADCMN	4096	CH0005	0	GL0007	0	OVERFLOWB	0	REALTMC	0	SLOWGAP	3
APRLVSTAT	0	CH0006	0	HDMVVAR	0	OVERFLOWC	0	REDTHR	12055	SLOWRESH	83
ASCMODE	32777	CH0007	0	INFTNABLE	1	PAR	193	RESETHRT	5000	SLOWTRUN	0
AUTOSDAC	1	CODEREV	0	MAPERRORS	0	PRR	190	RESETHWT	50	SPECIALRUN	0
BAS0000	2	CODEVAR	0	MAPNGARCOE	0	PCR	59	RESETHWT	156	SYNCRGSEC	50
BAS0001	22776	CRPKF	0	MAPXPKPKF	124	PCR	59	RUNERROR	0	THRESHOLD	179
BAS0002	1824	DECMATCH	4	MAXRNGTH	50	PEAPRT	66	RUNRNT	2	TLVTIME	58071
BAS0003	0	DETCHANNEL	0	MCAEVENTS	3982	PEAKMODE	1	RUNRNG	0	TLVTIMEA	583
BAS0004	65534	DETLEMENT	0	MCAEVENTSA	0	PEAKSAM	65	RUNSTATUS	0	TLVTIMEB	0
BAS0005	36684	DRFTLMT	25	MCAEVENTSB	0	PR	0	RUNTASKS	0	TLVTIMEC	0
BAS0006	35	ELVTIME	21987	MCAEVENTSC	0	PKELBASE	0	RUNTYPE	0	TRACCHAN	0
BAS0007	23632	ELVTIMEA	481	MCALM4	2048	PKELNUM	0	SCADLEN	512	TRACLEN	4096
BAS0008	65533	ELVTIMEB	0	MCALM0	0	PKELNUMA	0	SCADSTART	35840	TRACSTART	40960
BAS0009	11	ELVTIMEC	0	MCAPAGES	8	PKELRELEN	6448	SCAPTR	33440	TRACTYPE	0
BLAVGV	7	ERRINFO	0	MCAWORDS	8191	PKELRELENA	0	SCALM4	0	TRACVAT	0
BLOUT	0	ESCALE	3	MEMBASE	256	PKELSMATCH	0	SCALM4A	0	TROGERS	46409
BLOUTFACT	0	ETHR	0	MEMORY_MODE	0	PROEADACOR	0	SCALM4B	0	TROGERSA	0
BLUTERLEN	5	FASTGAP	0	MRWIDTH	2	PROEADPAGE	0	SCALM4C	0	TROGERSB	0
BLMAX	0	FASTLEN	5	MODESEL	0	PPPERBUF	124	SCALM4D	0	TROGERSC	0
BLMR	0	FFRFRV	69	NOCHLM	0	POLARITY	1	SCALM0	0	UNDERFLOWA	0
ETHR	68	FFRVAR	0	NUMASCNT	0	PREAMPTR	0	SCALM0A	0	UNDERFLOWB	0
BURMAPERRORS	0	FSCALE	2	NUMCRITON	877	PRESETLEN	0	SCALM0B	0	UNDERFLOWC	0
BURNUMBERA	0	ITHR	173	NUMCRITUP	795	PRESETLENA	0	SCALM0C	0	UNDERFLOWD	0
BURNUMBERB	0	QANDAC	4784	NUMNLS	65535	PRESETLENB	0	SCALM0D	0	USER	0
BURNUMBERC	0	GL0000	0	NUMNLSA	65535	PRESETLENC	0	SCALPTR	33408	YELLOWTR	16384
BUSY	0	GL0001	0	NUMNLSB	33	PRESETLEND	0	SCAMEMBASE	6448		
CH0000	0	GL0002	0	NUMNLS	105	PR	65535	SDAGV	2		
CH0001	0	GL0003	0	NUMSCA	0	RATECOLOR	0	SDAGVAT	50		
CH0002	0	GL0004	0	NUMSETS	2	REALTIME	64889	SLOPEDAC	32779		
CH0003	0	GL0005	0	OVERFLOWS	1	REALTIMEA	586	SLOPEVAL	0		

Figure 4.30: The DSP Parameters panel. Do not modify these values unless as instructed by XIA support staff.

4.7.3.2 Modifying DSP Parameters

In some cases, as directed by the XIA support staff, it may be necessary to modify the DSP's operating parameters directly. To edit a parameter select the field using the mouse, enter the new value and press [Return]. If you do not press [Return] the parameter will return to its unmodified value when another item is selected. Changing parameters in this panel without a deep understanding of XIA's DSP processors may produce exotic and unpredictable results. We recommend doing so only under the guidance of XIA support staff.

4.7.4 Submitting a problem report:

XIA encourages customers to report any problems encountered using any of our software via email. In most cases, the XIA engineering team will need to review bug information and run tests on local hardware before being able to respond.

All software-related bug reports should be e-mailed to software_support@xia.com and should contain the following information, which will be used by our technical support personnel to diagnose and solve the problem:

- ✓ Your name and organization

- ✓ Brief description of the application (type of detector, relevant experimental conditions...etc.)
- ✓ XIA hardware name and serial number
- ✓ Version of the library (if applicable)
- ✓ OS
- ✓ Description of the problem; steps taken to re-create the bug
- ✓ Full Error Report (see section 4.7.4.1) plus additional data:
 - Saved MCA data, if relevant (see section 4.7.4.2)
 - Saved Baseline data, if relevant (see section 4.7.4.3)
 - Saved Trace data, if relevant (see section 4.7.4.4)

Please compress the Error Report into a ZIP archive and attach the the support request email.

4.7.4.1 Generating a Full Error Report

If you are unable to solve your problem using the diagnostic tools in xManager, the last resort is to send as much information as possible to XIA support. This task is facilitated by the **Generate Error Report...** command under the **Help** menu. This feature creates a sub-directory titled "xManager_XXXXXX_Rpt" (where "XXXXXX" is the date) in the xManager installation directory that contains all DSP parameters, xManager registry settings, INI and log files, Handel log files, etc.

4.7.4.2 Saving MCA Data

If you are having difficulty acquiring a spectrum, or the spectrum looks strange, please save and submit a sample MCA file with the **Error Report**. In the **MCA** tab, acquire a spectrum, then press [**Save MCA Data**]. Save the file in the "xManager_XXXXXX_Rpt" sub-directory in the xManager installation directory.

4.7.4.3 Saving Baseline Data

If you are having difficulty acquiring a good baseline histogram, or the spectrum looks strange, please save and submit a sample baseline histogram file with the **Error Report**. In the **Baseline** tab, acquire a histogram, then press [**Save Baseline**]. Save the file in the "xManager_XXXXXX_Rpt" sub-directory in the xManager installation directory.

4.7.4.4 Saving Trace Data

If your ADC or filter output traces look strange, please save and submit a sample trace file with the **Error Report**. In the **Trace** tab, acquire a trace, then press [**Save Trace**]. Save the file in the "xManager_XXXXXX_Rpt" sub-directory in the xManager installation directory.

5 Mapping Mode

Mapping mode supports a number of time-resolved experiments, but is primarily intended for x-ray scanning applications wherein the x-ray beam is systematically scanned over a surface in order to produce a pixelated spectral map of the sample. Data is stored sequentially at each 'pixel advance' instruction, which can be provided by the host computer or triggered by an external logic signal, i.e. GATE or SYNC.

Mapping mode utilizes a different System FPGA configuration, i.e. firmware must be downloaded when switching between acquisition modes. A dual-memory architecture allows the DXP-XMAP to continuously acquire data in this mode. Host readout must be synchronized with the hardware in order to sustain continuous operation.

5.1 Pixel Advance Settings

The pixel advance settings are included in the **Configuration Wizard** utility, described in section 3.2. To run the utility select **Configuration Wizard...** from the **Tools** menu.

At the beginning of a run the pixel number starts at zero, corresponding to the x-ray beam initial position. The pixel number advances in several possible ways, either using digital hardware lines for real time applications or by computer control. These methods are described in detail below.

5.1.1 LBUS PCI Extension (Run Start Synchronization)

It is assumed that the LBUS PCI bus extension connections are used (see also section 7.2.2). In the figures below, 'Start run' commands have been issued by the host to all XMAP modules. The transition from LO to HI on LBUS indicates that all XMAP modules are ready, and in fact have 'started' simultaneously, however, data acquisition does not actually begin until the GATE signal is released. The LBUS output can thus be used as a 'READY' logic input to the beam rastering hardware at the start of each run, i.e. the user generated GATE signal disables data acquisition until all such system constraints have been met. The alternative is to use a fixed, conservative waiting period between the 'Start Run' command and the GATE release.

5.1.2 Pixel Advance on GATE Edge

The recommended method for advancing the pixel is to use the GATE digital input, where the pixel advance occurs on every trailing edge of the signal (the transition from active data acquisition to the inactive state). By default the GATE signal also halts data acquisition when it is LO and the pixel advances on every falling edge, as in **Figure 5.1**.

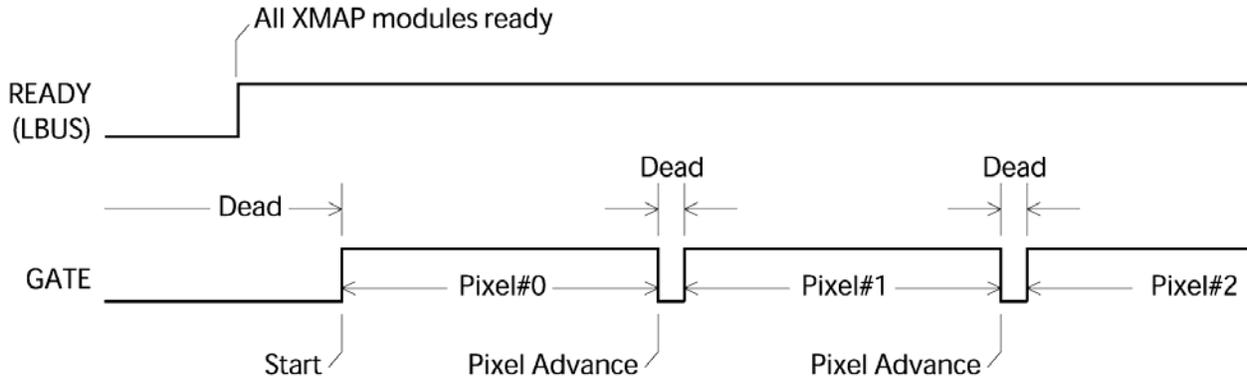


Figure 5.1: GATE pixel advance, with data acquisition halted during the LO periods. The pixel advance occurs at each falling edge.

5.1.2.1 GATE Polarity

The interpretation of the user-provided GATE signal can be inverted such that data acquisition is halted when the signal is HI (see **Figure 6.2**) and the pixel advance occurs on rising edges.

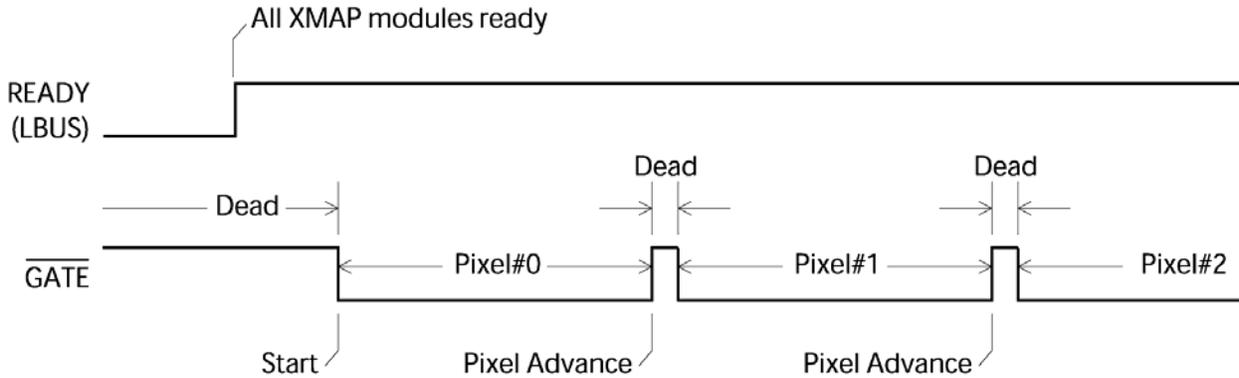


Figure 5.2: GATE pixel advance, with data acquisition halted during the HI periods. The pixel advance occurs at each rising edge.

5.1.2.2 GATE Ignore Setting

Normally the GATE signal is used both to advance the pixel and to halt data taking. In some cases the user may prefer continuous operation through the pixel advance operations. With the GATE Ignore option selected, the pixel advance occurs but data acquisition is not halted. Note that the data acquisition *is* halted at the beginning of the run until GATE is released the first time, i.e. run start synchronization is still available.

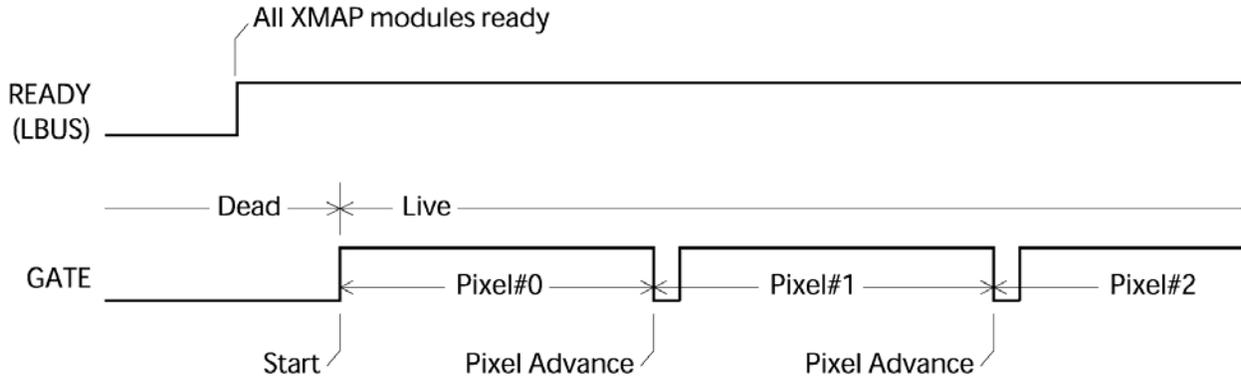


Figure 5.3: GATE pixel advance, with LO polarity and the Gate Ignore enabled. Acquisition is halted initially until GATE is released for the first time. As before, the pixel advance occurs at each falling edge, but acquisition continues during the LO periods.

The above examples all use the same cable connections, shown in **Figure 5.4** below. Only one connection each for LBUS and GATE are necessary for each PCI bus segment.

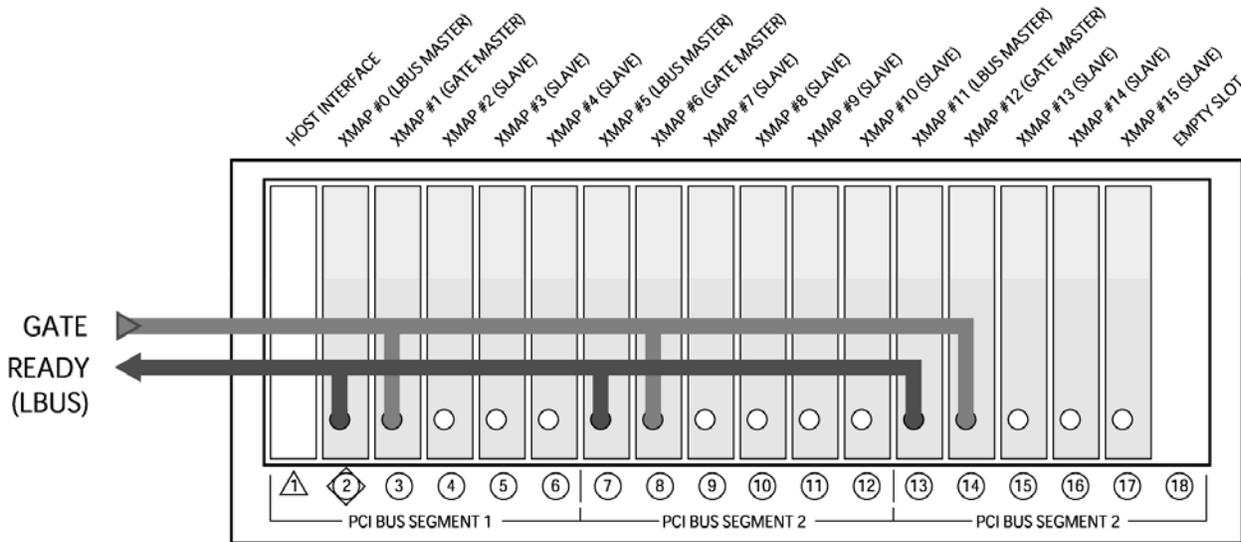


Figure 5.4: The LEMO cable connections for GATE pixel advance, with LBUS enabled. If LBUS is not used, GATE connections would shift over one module to the left, connecting to slots 2, 7 and 13.

5.1.3 Pixel Advance using SYNC Clock

The SYNC signal can also be used to generate the pixel advance. Using this method, the pixel will advance for every N positive pulses on the SYNC line, where N can be set from 1 to 65535. Note that the pulses must be greater than 40 ns wide to be guaranteed to be recognized.

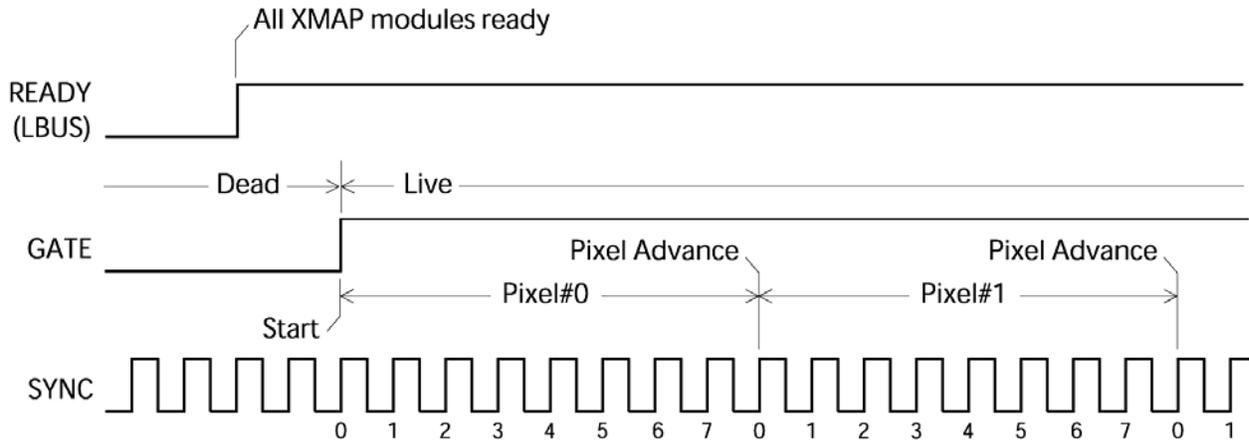


Figure 5.5: SYNC pixel advance with LBUS and GATE used to synchronize the run start. Acquisition is halted initially until GATE is released for the first time. The pixel advance occurs every N cycles of the SYNC clock.

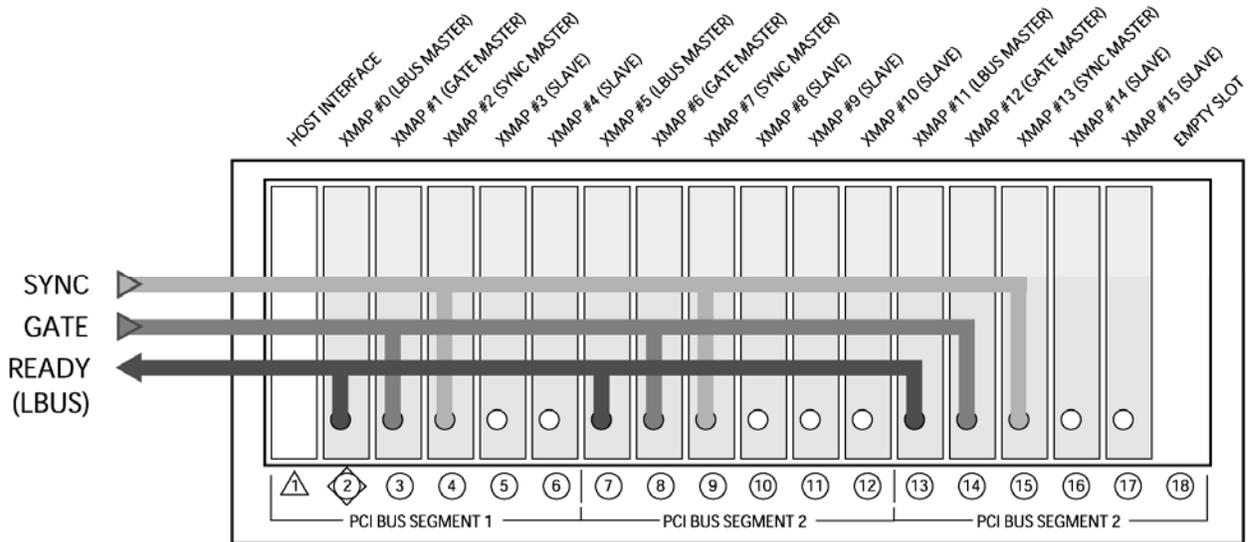


Figure 5.6: The LEMO cable connections for SYNC pixel advance, with LBUS and GATE used to synchronize the run start. If LBUS is not used, GATE connections would shift over one module to the left, connecting to slots 2, 7 and 13.

5.1.4 Pixel Advance under Host Control

It is also possible to advance the pixel using Handel. Manually advancing the pixels is slower and unwieldy because the command must be issued separately to each module, but it does provide an easy way to test mapping operations.

5.2 Mapping Mode Data Acquisition

Mapping mode acquisition is automatically enabled when the **Mapping** panel is active. Conversely, normal mode acquisition is automatically enabled when the **MCA** tab is active. In both cases the corresponding System FPGA firmware is downloaded (as evidenced by LED pattern flashing).

5.2.1 The Mapping Panel

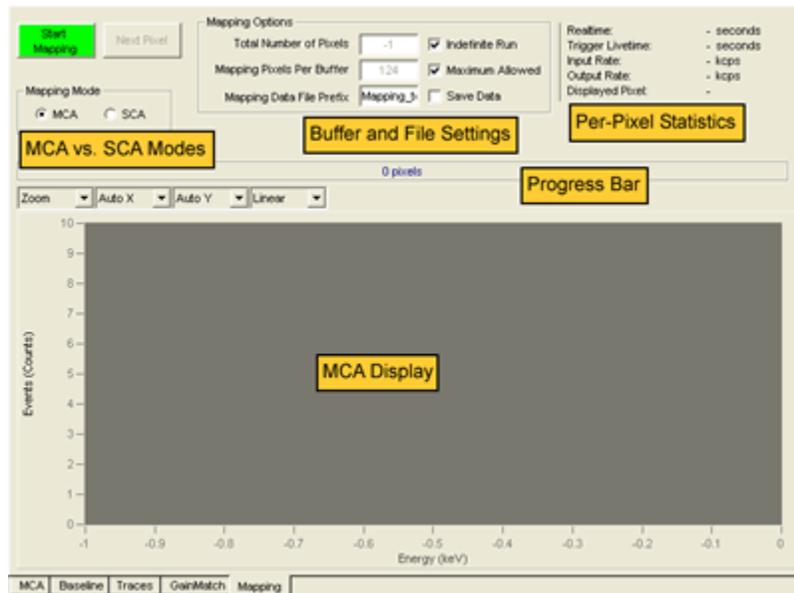


Figure 5.7: The Mapping panel.

Click the **Mapping** tab to display the **Mapping** panel. Please be patient while the System FPGA firmware is downloaded, during which period the controls will be grayed out. The layout and functionality are similar to the **MCA** panel. The **[Start Mapping]** button starts and stops mapping data acquisition runs. Output statistics are displayed as before along the top right section of the panel. Notice that data and statistics do not update in realtime. This is as a result of the dual memory architecture. The display is refreshed only after a memory buffer swap.

5.2.2 Mapping Mode: MCA or SCA

The MCA and SCA mapping modes are distinct and exclusive of one another: In MCA mode full spectra are stored in memory; In SCA mapping mode only the tables of SCA counts are stored in memory. Click the MCA or SCA radio button to select the desired mode. Section 5.3 details the data formats for MCA and SCA mapping modes.

5.2.3 Total Number of Pixels

A data acquisition run can be ended manually via the host at any time. In most cases the length of a data acquisition run corresponds to the number of pixels in the map. The **Total Number of Pixels** setting instructs the DSP to

automatically end the run after that number of pixel advances have been detected. If **Indefinite Run** is checked, the run ends only upon the "End Run" host command.

5.2.4 Buffer Control

The XMAP uses a dual memory architecture to achieve continuous data acquisition: The XMAP writes into one 'active' buffer while the host reads the other 'inactive' buffer. The size of each buffer is 2MB, organized as 1Mword by 16 bits. The XMAP must periodically swap buffers. The swap must occur after the host has completed reading the inactive buffer and before the active buffer becomes full.

The **Mapping Pixels Per Buffer** setting controls when the swap takes place. The DSP also calculates the maximum number of pixels that can be stored in a memory buffer based upon the **MCA Number of Bins**. If the **Maximum Allowed** checkbox is checked, *or if a greater number is entered*, the XMAP will override the **Mapping Pixels Per Buffer** and use this calculated maximum number instead.

The effects of this setting are generally qualitative:

- Changes the structure but not the content of the data stream binary file (see section 5.3).
- The xManager display updates after each buffer, thus the number of pixels per buffer sets the visual refresh rate.

5.2.5 Mapping Mode Data Acquisition

Press the **[Start Mapping]** button to begin a data acquisition run. If you are using GATE or SYNC for the pixel advance, you should see the **Progress Bar** begin to fill. If it does not, please review sections 5.1 and 7.2, adjust your hardware and use the **Configuration Wizard** (see section 3.2.3) to correct your settings if necessary. Alternatively, press the **[Next Pixel]** button to manually advance pixels. When the incremental number of processed pixels exceeds the **Mapping Pixels Per Buffer** setting, the buffer will be read and spectrum (or SCA data) and statistics corresponding to the first pixel from the buffer will be displayed. When the total number of pixels processed exceeds the **Total Number of Pixels** setting, the display will again refresh and the run will automatically end. Alternatively, the **[Stop Mapping]** button can be pressed at any time to end the run.

5.3 Mapping Mode Data

In mapping mode, the DXP-xMAP uses two completely separate memory buffers, enabling the system to take data into one buffer while the other buffer can be read out by the host. The capacity of each buffer is 2MB, organized as 1Mword by 16 bits; however, the portion that is used can be significantly smaller, depending on the **Mapping Pixels Per Buffer** setting (see section 5.2.4). Dealing with all this data in real time is a challenge. xManager streams the buffer data to a binary file for offline analysis. For visual feedback, the display is refreshed with data from the first pixel of each buffer that is read out.

5.3.1 Mapping Data Options

The name of the output file is automatically generated with the user specified **File Prefix**. For diagnostic purposes, the user can elect not to save output data. If the **Save Mapping Data** checkbox is un-checked, the output data will be lost.

The default location of the output file is:

C:\Program Files\xia\xManager 0.x\data

To change the folder, select **Options** from the **Tools** menu, then select the **Mapping Data** tab to display the **Mapping Data File Options**. For the present release each data acquisition run generates a single binary file in the specified folder, with a unique filename with the specified prefix.

5.3.2 Mapping Data Format

Currently the data file created is a binary concatenation of all the buffers read out during the run. **Table 5.1** shows the structure of the file created. The following sections specify the buffer level formatting.

Buffer 0: Module 0
Buffer 0: Module 1
Buffer 0: Module 2
Buffer 0: Module 3
Buffer 1: Module 0
Buffer 1: Module 1
Buffer 1: Module 2
Buffer 1: Module 3
...
Buffer i-1: Module 0
Buffer i-1: Module 1
Buffer i-1: Module 2
Buffer i-1: Module 3

Table 5.1: The Mapping mode binary data output file format of a 4 XMAP module system for a data acquisition run comprising 'i' buffers.

5.3.3 Single Buffer Format

The buffer starts out with a buffer header, containing general information about the data contained in the memory block, followed by pixel

data. For each pixel there is a pixel data block containing a header and spectra for all 4 channels. The pixel data block format depends on the mapping mode: MCA, SCA or List Mode. **Table 5.3** shows the buffer level structure. The following sections describe the header and pixel data blocks in detail.

Buffer Header
Pixel X Data Block
Pixel X+1 Data Block
...
Pixel X+j-1 Data Block

Table 5.2: A single buffer for a data acquisition run with **Mapping Pixels Per Buffer = 'j'**.

5.3.3.1 Statistics Units

Realtime and Livetime are expressed in 320ns units.

5.3.3.2 Buffer Header

For all timing applications that use the dual buffers, the buffer header will have a fixed, 256-word length (the word size is 16 bits in this mode). The contents of the header are defined below:

<i>Word Number</i>	<i>Contents</i>
Word Number	Contents
0	Tag Word 0: 0x55AA
1	Tag Word 1: 0xAA55
2	Buffer Header Size (=256)
3	Mapping Mode: 1: Full Spectrum 2: Multiple ROI 3: List Mode
4	Run Number
5 to 6	Sequential Buffer Number (low word first)
7	BufferID (0:A, 1:B)

8	Number of Pixels in buffer
9 to 10	Starting Pixel Number (low word first)
11	Module Serial Number?/Module #
12	Detector Channel 0 (set by host in DSP)
13	Det. Element, Ch0
14	Detector Channel 1 (set by host in DSP)
15	Det. Element, Ch1
16	Detector Channel 2 (set by host in DSP)
17	Det. Element, Ch2
18	Detector Channel 3 (set by host in DSP)
19	Det. Element, Ch3
20	Channel 0 Size (number of words)
21	Channel 1 Size (number of words)
22	Channel 2 Size (number of words)
23	Channel 3 Size (number of words)
24	Buffer errors: Buffer overrun 0: No error >0: Number of extra pixels combined with last pixel in buffer
25-31	Reserved (set to 0)
32-63	32 User words (set in USER DSP array)
64-255	Reserved (set to 0)

Table 5.3: The buffer header contents.

5.3.3.3 Mapping Mode 1: Full Spectrum Mapping

The pixel header for full spectrum mapping mode is described below; due to the constraint that the spectra sizes are a multiple of 256 and must start on an even multiple of 256, the size of the pixel header is 256 words in this mode.

The data block for full spectrum mapping mode contains four sections; each section holds the spectrum from one of the four detector channels in the

module. The length of the spectra are constrained to be a multiple of 256, and must start on a memory location that is a multiple of 256. For (at least) the first version of the mapping firmware, the spectra must be the same size for all channels in a system (and will in general equal $256 * 2^n$, ie 256, 512, 1024, 2048, etc). The format for the entire pixel block is described in the table below.

Word Number	Contents
0	Tag Word 0: 0x33CC
1	Tag Word 1: 0xCC33
2	Pixel Header Size (=256)
3	Mapping Mode (=1)
4 to 5	Pixel Number (low word first) In the case of a mapping error where one pixel record combines data from several pixels, this is the number of the last pixel recorded.
6 to 7	Total Pixel Block size in words (including header) (low word first)
8	Channel 0 Size (K words)
9	Channel 1 Size (L words)
10	Channel 2 Size (M words)
11	Channel 3 Size (N words)
12 to 31	Reserved (set to 0)
32 to 39	Channel 0 Statistics: Realtime (2 words, low word first) Livetime (2 words) Triggers (2 words) Output events (2 words)
40 to 47	Channel 1 Statistics

48 to 55	Channel 2 Statistics
56 to 63	Channel 3 Statistics
64 to 255	Reserved (set to 0)
256 to $(256 + K - 1)$	Channel 0 Spectrum
$(256 + K)$ to $(256 + K + L - 1)$	Channel 1 Spectrum
$(256 + K + L)$ to $(256 + K + L + M - 1)$	Channel 2 Spectrum
$(256 + K + L + M)$ to $(256 + K + L + M + N - 1)$	Channel 3 Spectrum

Table 5.4: The pixel data block for Mapping Mode 1.

5.3.3.4 Mapping Mode 2: Multiple SCA Mapping

The pixel header for multiple SCA (or ROI, for Region Of Interest) mapping mode is described below; there is no constraint on the data alignment in the buffer, so the header length is shorter than the 256 words required for full spectrum mapping.

There are four sections in the pixel data block for this mode, containing the SCA totals for each of the four detector channels. Up to 64 SCA's can be defined for each channel. A 16K word array is used to hold the mapping between MCA bins and SCA regions; the user can either select to have all channels use the same SCA definitions for all channels (which supports the full maximum MCA length of 16K channels), or use separate definitions of the SCA regions for each channel (where the maximum supported MCA length is 4K channels). This format specification does allow differences in the number of SCA's between channels. Two words (32 bits total) are used to store the total number of events in each SCA region; the low word is stored first in memory.

The format for the entire pixel data block is described in the table below.

Word Number	Contents
0	Tag Word 0: 0x33CC
1	Tag Word 1: 0xCC33
2	Pixel Header Size (=64)
3	Mapping Mode (=2)
4 to 5	Pixel Number (low word first) In the case of a mapping error where one

	pixel record combines data from several pixels, this is the number of the last pixel recorded.
6 to 7	Total Pixel block size in words (including header) (low word first)
8	Number of ROI, Channel 0 (K ROI's)
9	Number of ROI, Channel 1 (L ROI's)
10	Number of ROI, Channel 2 (M ROI's)
11	Number of ROI, Channel 3 (N ROI's)
12	ROI Size in words (=2)
13 to 31	Reserved (set to 0)
32 to 39	Channel 0 Statistics: Realtime (2 words, low word first) Livetime (2 words) Triggers (2 words) Output events (2 words)
40 to 47	Channel 1 Statistics (as above)
48 to 55	Channel 2 Statistics (as above)
56 to 63	Channel 3 Statistics (as above)
64 to $(64 + 2*K - 1)$	Channel 0 ROI data
$(64 + 2*K)$ to $(64 + 2*K + 2*L - 1)$	Channel 1 ROI data
$(64 + 2*K + 2*L)$ to $(64 + 2*K + 2*L + 2*M - 1)$	Channel 2 ROI data
$(64 + 2*K + 2*L + 2*M)$ to $(64 + 2*K + 2*L + 2*M + 2*N - 1)$	Channel 3 ROI data

Table 5.5: The pixel data block for Mapping Mode 2.

5.3.3.5 Mapping Mode 3: List Mode Mapping

The pixel header for list mode mapping is described below; there is no constraint on the data alignment in the buffer, so the header length is shorter than the 256 words required for full spectrum mapping. Note that in list mode, the data for all channels is mixed; the channel number is embedded into the data word itself.

For the data block in list mode, each word contains data for a single event. The events are stored in the order they are processed, and so the data are not separated according to channel. The channel number information is embedded into the upper two bits of the data word itself; the lower 14 bits are used to store energy information (typically MCA channel number).

Word Number	Contents
0	Tag Word 0: 0x33CC
1	Tag Word 1: 0xCC33
2	Pixel Header Size (=64)
3	Mapping Mode (=3)
4 to 5	Pixel Number (low word first) In the case of a mapping error where one pixel record combines data from several pixels, this is the number of the last pixel recorded.
Word Number	Contents
6 to 7	Total Pixel block size in words (including header) (low word first)
8	Number of Channel 0 Events (K)
9	Number of Channel 1 Events (L)
10	Number of Channel 2 Events (M)
11	Number of Channel 3 Events (N)
12 to 31	Reserved (set to 0)
32 to 39	Channel 0 Statistics: Realtime (2 words, low word first) Livetime (2 words)

	Triggers (2 words) Output events (2 words)
40 to 47	Channel 1 Statistics
48 to 55	Channel 2 Statistics
56 to 63	Channel 3 Statistics
64 to (64 + K + L + M + N -1)	Event data

Table 5.6: The pixel data block for Mapping Mode 3.

6 Digital Filtering: Theory of Operation and Implementation Methods

This chapter provides an in-depth discussion of x-ray pulse-processing theory both generally and as implemented in the DXP-XMAP. The topics include x-ray detection how, digital trapezoidal filter basics, thresholds, baselines, peak sampling, pileup inspection, and input and output count rates. Topics are covered to illustrate the theoretical issues, practical implementation, and how to adjust parameters to obtain best performance.

The acronym DXP stands for “Digital X-ray Processor” and refers to a digital processing technology, for which XIA has received several US and International patents.

6.1 X-ray Detection and Preamplifier Operation:

Energy dispersive detectors, which include such solid state detectors as Si(Li), HPGe, HgI₂, CdTe and CZT detectors, are generally operated with charge sensitive preamplifiers. When an x-ray is absorbed in the detector material it releases an electric charge $Q_x = E_x/\epsilon$, where the material constant ϵ is the amount of energy needed to form an electron-hole pair. Q_x is integrated onto the preamplifier’s feedback capacitor C_f , to produce the voltage $V_x = Q_x/C_f = E_x/(\epsilon C_f)$. Measuring the energy E_x of the x-ray therefore requires a measurement of the voltage step V_x in the presence of the amplifier’s noise σ . Figure 6.1 and Figure 6.3 depict reset-type and RC-type charge sensitive amplifiers, respectively. In both figures the detector D is biased by voltage source HV (either positive or negative) and connected to the input of amplifier A. Note that the *signal polarity* must be distinguished from the *bias voltage polarity*. The signal polarity is positive if the voltage step V_x is a rising edge, as displayed in Figure 6.1. Whether signal polarity is positive or negative depends upon the preamplifier’s design and does not depend upon bias voltage polarity, which is specified on the detector and is determined by its design.

6.1.1 Reset-Type Preamplifiers

Figure 6.1a is a simplified schematic of a reset-type preamplifier, wherein C_f is discharged through the switch S from time to time when the amplifier’s output voltage gets so large that it behaves nonlinearly. Switch S may be an actual transistor switch, or may operate equivalently by another mechanism. In pulsed optical reset preamps light is directed at amplifier A’s input FET causing it to discharge C_f . In transistor reset preamps, the input FET may have an additional electrode which can be pulsed to discharge C_f . The output of a reset-type preamplifier following the absorption of an x-ray of energy E_x in detector D is a voltage step of amplitude V_x . Two x-ray steps are shown in Figure 6.3b as a step.

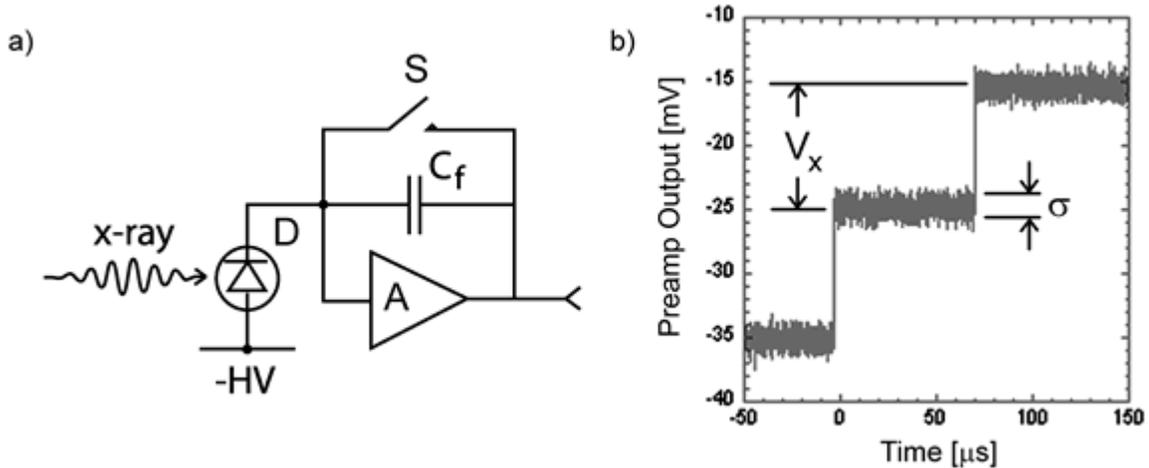


Figure 6.1: a) Reset-type charge sensitive preamplifier with a negatively biased detector; b) Output on absorption of x-ray rays. Note that the steps have a rising edge, so that the signal polarity is positive.

Figure 6.2 depicts the large-signal sawtooth waveform that results from successive x-ray steps followed by the reset. Note that the units here are Volts and milliseconds vs. millivolts and microseconds in the previous figure.

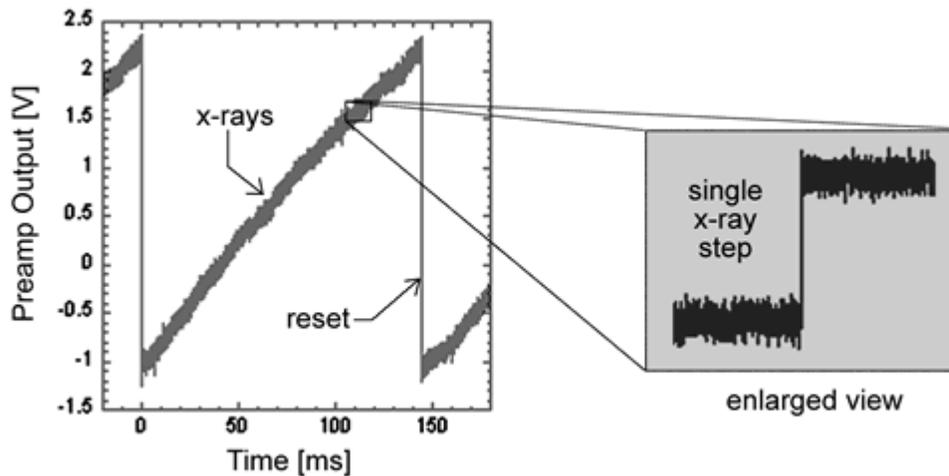


Figure 6.2: The large-signal reset waveform for a reset-type preamplifier with positive signal polarity, as displayed on a real oscilloscope. Note that the large signal character of the DXP-XMAP diagnostic ADC readout, used in xManager's **Trace** panel, looks quite different because of the dynamic range reduction carried out in the ASC, as described in section 0.

6.1.2 RC-Type Preamplifiers

Figure 6.3a is a simplified schematic of an RC-type preamplifier, wherein C_f is discharged continuously through feedback resistor R_f . The output of an RC-type preamplifier following the absorption of an x-ray of energy E_x in detector D is, again, a voltage step of amplitude V_x . The continuous discharge of C_f through R_f results in an exponential voltage decay after the x-ray step, with decay constant τ , where:

$$\tau = R_f C_f$$

Equation 6-1

In practice the decay time may depend on subsequent circuitry, i.e. if a pole-zero cancellation circuit is used, thus τ may not be directly related to the feedback elements of the front-end. The point of this simplified model is that the resulting waveform is a single-pole RC decay. The discussion in section 6.2 through section 6.6.2 assumes a reset-type preamplifier, but is mostly applicable to RC-type preamplifiers. section 6.7 describes the few key differences in the processing of RC-type preamplifier signals.

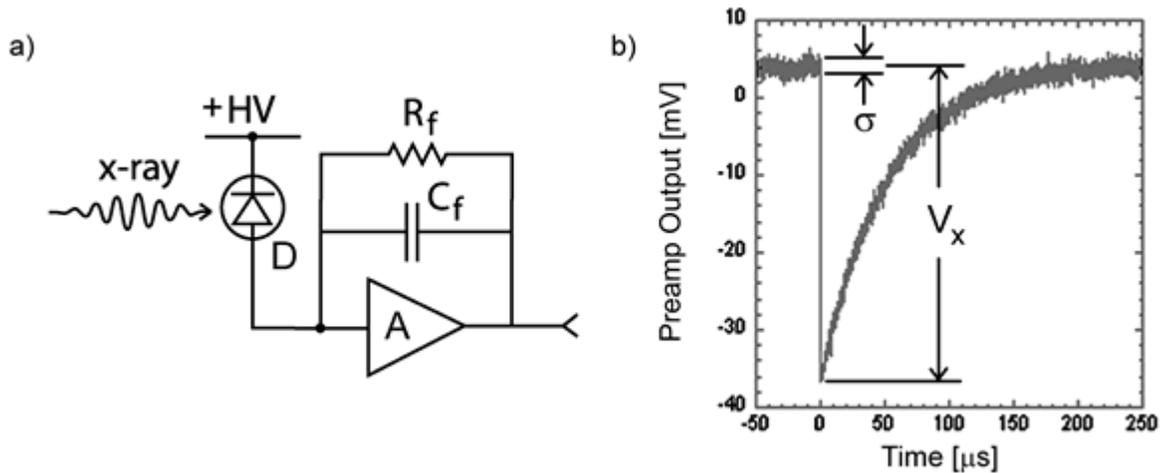


Figure 6.3: a) RC-type charge sensitive preamplifier with a positively biased detector; b) Output on absorption of an x-ray. Note that the step has a falling edge, thus the signal polarity is negative.

6.2 X-ray Energy Measurement & Noise Filtering:

Reducing noise in an electrical measurement is accomplished by filtering in the frequency, or conversely, the time domain. When discussing digital pulse-processor filters it's more straightforward to use the time domain. Traditional analog pulse-processing filters use combinations of a differentiation stage and multiple integration stages to convert the preamp output steps, such as shown in Figure 6.1b, into either triangular or semi-Gaussian pulses whose amplitudes (with respect to their baselines) are then proportional to V_x and thus to the x-ray's energy.

6.2.1 Digital Filtering Theory

Digital filtering proceeds from a slightly different perspective. Here the signal has been digitized and is no longer continuous, but is instead a string of discrete values, such as shown in Figure 6.4. The data displayed are actually just a subset of Figure 6.3b, which was digitized by a Tektronix 544 TDS digital oscilloscope at 10 MHz (10 million per second). Given this data set, and some kind of arithmetic processor, the obvious approach to determining V_x is to take some sort of average over the points before the step and subtract it from the value of the average over the points after the step. That is, as shown in Figure

6.4, averages are computed over the two regions marked “Length” (the “Gap” region is omitted because the signal is changing rapidly here), and their difference taken as a measure of V_x . Thus the value V_x may be found from the equation:

$$V_{x,k} = -\sum_{i(\text{before})} w_i v_i + \sum_{i(\text{after})} w_i v_i$$

Equation 6-2

where the values of the weighting constants w_i determine the type of average being computed. The sums of the values of the two sets of weights must be individually normalized.

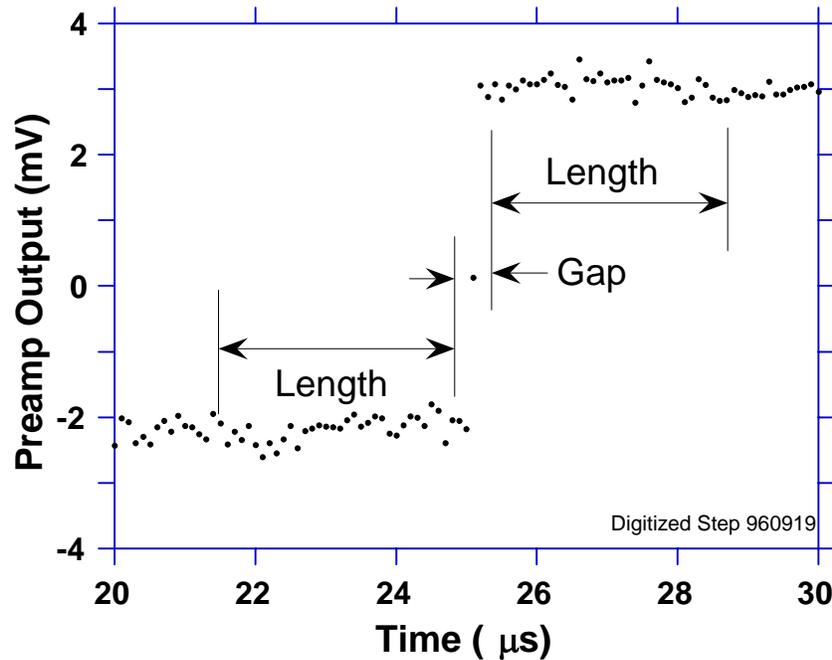


Figure 6.4: Digitized version of one of the x-ray steps of Figure 6.3b.

The primary differences between different digital signal processors lie in two areas: what set of weights $\{w_i\}$ is used and how the regions are selected for the

computation of
$$V_{x,k} = -\sum_{i(\text{before})} w_i v_i + \sum_{i(\text{after})} w_i v_i$$

Equation 6-2. Thus, for example, when the weighting values decrease with separation from the step, then the equation produces “cusp-like” filters. When the weighting values are constant, one obtains triangular (if the gap is zero) or trapezoidal filters. The concept behind cusp-like filters is that, since the points nearest the step carry more information about its height, they should be more strongly weighted in the averaging process. How one chooses the filter lengths results in time variant (the lengths vary from pulse to pulse) or time invariant (the lengths are the same for all pulses) filters. Traditional analog filters are time invariant. The concept behind time variant filters is that, since

the x-rays arrive randomly and the lengths between them vary accordingly, one can make maximum use of the available information by adjusting Length on a pulse by pulse basis.

In principal, the very best filtering is accomplished by using cusp-like weights and time variant filter length selection. There are serious costs associated with this approach however, both in terms of computational power required to evaluate the sums in real time and in the complexity of the electronics required to generate (usually from stored coefficients) normalized $\{w_i\}$ sets on a pulse by pulse basis. A few such systems have been produced but typically cost about \$13K per channel and are count rate limited to about 30 Kcps. Even time invariant systems with cusp-like filters are still expensive due to the computational power required to rapidly execute strings of multiply and adds. One commercial system exists which can process over 100 Kcps, but it too costs over \$12K per channel.

6.2.2 Trapezoidal Filtering

The DXP processing system developed by XIA takes a different approach because it was optimized for very high speed operation and low cost per channel. It implements a fixed length filter with all w_i values equal to unity and in fact computes this sum afresh for each new signal value k . Thus the equation implemented is:

$$L V_{x,k} = \sum_{i=k-2L-G+1}^{k-L-G} v_i + \sum_{i=k-L+1}^k v_i$$

Equation 6-3

where the filter length is L and the gap is G . The factor L multiplying $V_{x,k}$ arises because the sum of the weights here is not normalized. Accommodating this factor is trivial for the DXP's host software. The operations are carried out using hardwired logic in a field programmable gate array (FPGA) that is called the FiPPI because it implements Filtering, Peak capture, and Pileup Inspection.

In the FiPPI, Equation 6-3 is actually implemented by noting the recursion relationship between $V_{x,k}$ and $V_{x,k-1}$, which is:

$$L V_{x,k} = L V_{x,k-1} + v_k - v_{k-L} - v_{k-L-G} + v_{k-2L-G}$$

Equation 6-4

While this relationship is very simple, it is still very effective. In the first place, this is the digital equivalent of triangular (or trapezoidal if $G = 0$) filtering which is the analog industry's standard for high rate processing. In the second place, one can show theoretically that if the noise in the signal is white (i.e. Gaussian distributed) above and below the step, which is typically the case for the short shaping times used for high signal rate processing, then the average in Equation 6-4 actually gives the best estimate of V_x in the least squares sense. This, of course, is why triangular filtering has been preferred at high rates. Triangular filtering with time variant filter lengths can, in principle, achieve both somewhat superior resolution and higher throughputs but comes at the cost of a significantly more complex circuit and a rate dependent resolution, which is unacceptable for many types of precise analysis. In practice, XIA's design has

been found to duplicate the energy resolution of the best analog shapers while approximately doubling their throughput, providing experimental confirmation of the validity of the approach.

6.3 Trapezoidal Filtering in the DXP:

From this point onward, we will only consider trapezoidal filtering as it is implemented in the DXP according to Equation 6-3 and Equation 6-4. The result of applying such a filter with Length $L = 20$ and Gap $G = 4$ to the same data set of Figure 6.4 is shown in Figure 6.5. The filter output V_x is clearly trapezoidal in shape and has a risetime equal to L , a flattop equal to G , and a symmetrical falltime equal to L . The basewidth, which is a first-order measure of the filter's noise reduction properties, is thus $2L+G$.

6.3.1 Comparing DXP Performance

This raises several important points in comparing the noise performance of the DXP to analog filtering amplifiers. First, semi-Gaussian filters are usually specified by a *shaping time*, which is roughly half of the peaking. Their pulses typically are not symmetric so that the basewidth is about 5.6 times the shaping time or 2.8 times their peaking time. Thus a semi-Gaussian filter typically has a slightly better energy resolution than a triangular filter of the same peaking time because it has a longer filtering time. This is typically accommodated in amplifiers offering both triangular and semi-Gaussian filtering by stretching the triangular peaking time a bit, so that the *true* triangular peaking time is typically 1.2 times the selected semi-Gaussian peaking time. This also leads to an apparent advantage for the analog system when its energy resolution is compared to a digital system with the same nominal peaking time. A valid energy resolution comparison *must start with filters that have equal basewidths, and thus equal throughput*, e.g. The energy resolution of an analog system with shaping time of $1 \mu\text{s}$ should be compared to that of a DXP with a peaking time of $2.8 \mu\text{s}$.

6.3.2 Decimation and Peaking Time Ranges

A practical limitation on the implementation of Equation 6-4 is that two FIFO memories are required, one of length L and one of Length $L+G$. Since memory space is limited in FPGAs, we have restricted our designs to values of $L+G$ less than 128. The DXP-XMAP samples at 50 MHz (20 ns clock period), so this corresponds to a maximum peaking time of $2.56 \mu\text{s}$. XIA overcomes this limitation by first pre-averaging the data stream from the ADC by performing sequential sums of D data points, where $D = 2^N$. We refer to this pre-averaging procedure as “Decimating by N ”. By feeding the decimated data in an Equation 6-4 filter, we now obtain peaking times that are extended to $L*D$. It is important to understand that no data are lost in this procedure; we have merely rearranged the order of the summations represented in Equation 6-3. By extension, a “Decimation N FiPPI” is one that decimates the data by N before applying the energy filter. The common decimation values in the DXP-XMAP are 0, 2, 4, and 6, corresponding to averaging times of 20 ns (no averaging), 80 ns, 320 ns, and $1.28 \mu\text{s}$, respectively.

Decimation by N means to pre-average sequential sums of length $D = 2^N$.

<i>Decimation N</i>	<i>ADC Clock Period Δt</i>	<i>#ADC Samples in Average 2^N</i>	<i>Decimation Period $\Delta t * 2^N$</i>	<i>Peaking Time Range*</i>
0	20 ns	1	20 ns	80 ns – 2.58 μ s
2	20 ns	4	80 ns	320 ns – 10.32 μ s
4	20 ns	16	320 ns	1.28 μ s – 41.28 μ s
6	20 ns	64	1.28 μ s	5.12 μ s – 163.84 μ s

*Experience has shown that an absolute minimum slow filter length of 4 should be used.

Table 6.1: FiPPI decimation details.

In practice it is important to realize that implementing an energy filter in a Decimation N FiPPI sets certain limitations on the flat-top lengths that can be obtained in trapezoidal filters. Because the decimation process is uncorrelated with the arrival of x-rays, the gap G must be 3 or greater to assure that the filter's peak truly represents the x-ray's energy. Therefore, the minimum Decimation N gap time is $G * 2^N * \Delta t$, where Δt is the ADC's sampling interval. With the DXP-XMAP's $\Delta t = 20$ ns sampling interval, for instance, the smallest useful flat-top in Decimation 6 is $3 * 1.28 \mu\text{s} = 3.84 \mu\text{s}$.

Given the significant overlap in peaking time ranges, it is generally better to choose a lower decimation value, such that a shorter gap time can be used. Decimation 0 has other limitations, i.e. no intermediate baseline filter, and is thus an exception to this rule. The FDD firmware file defines the actual, i.e. non-overlapping, peaking time ranges used.

6.3.3 Time Domain Benefits of Trapezoids

One extremely important characteristic of a digitally shaped trapezoidal pulse is its extremely sharp termination on completion of the basewidth $2L+G$. This may be compared to analog filtered pulses which have tails which may persist up to 40% of the peaking time, a phenomenon due to the finite bandwidth of the analog filter. As we shall see below, this sharp termination gives the digital filter a definite rate advantage in pileup free throughput.

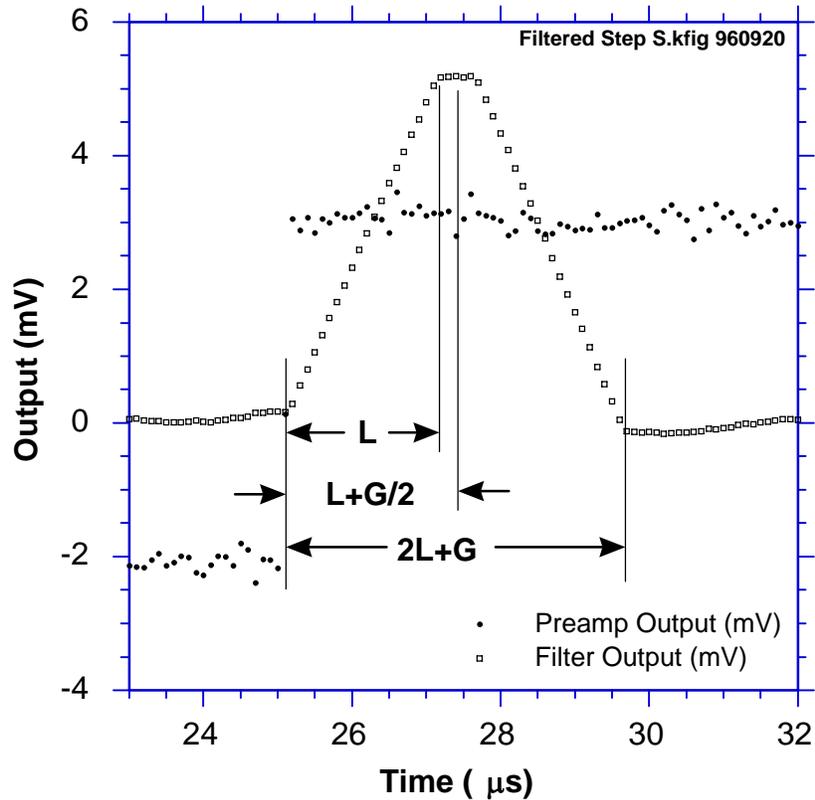


Figure 6.5: Trapezoidal filtering the Preamp Output data of Figure 6.4 with $L = 20$ and $G = 4$.

6.4 Baseline Issues:

6.4.1 The Need for Baseline Averaging

Figure 6.6 shows the same event as is Figure 6.5 but over a longer time interval to show how the filter treats the preamplifier noise in regions when no x-ray pulses are present. As may be seen, the effect of the filter is both to reduce the amplitude of the fluctuations and reduce their high frequency content. This signal is termed the *baseline* because it establishes the reference level or offset from which the x-ray peak amplitude V_X is to be measured. The fluctuations in the baseline have a standard deviation σ_e which is referred to as the *electronic noise* of the system, a number which depends on the peaking time of the filter used. Riding on top of this noise, the x-ray peaks contribute an additional noise term, the *Fano noise*, which arises from statistical fluctuations in the amount of charge Q_X produced when the x-ray is absorbed in the detector. This Fano noise σ_f adds in quadrature with the electronic noise, so that the total noise σ_t in measuring V_X is found from

$$\sigma_t = \text{sqrt}(\sigma_f^2 + \sigma_e^2)$$

Equation 6-5

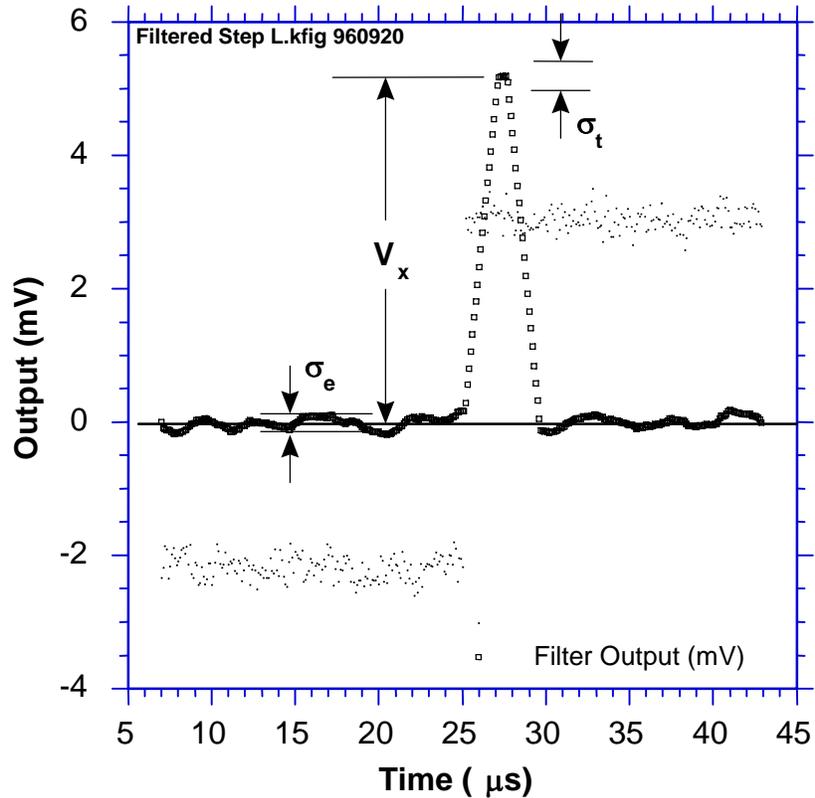


Figure 6.6: The event of Figure 6.5 displayed over a longer time period to show baseline noise.

The Fano noise is only a property of the detector material. The electronic noise, on the other hand, may have contributions from both the preamplifier and the amplifier. When the preamplifier and amplifier are both well designed and well matched, however, the amplifier's noise contribution should be essentially negligible. Achieving this in the mixed analog-digital environment of a digital pulse processor is a non-trivial task, however.

In the general case, the mean baseline value is not zero. This situation arises whenever the slope of the preamplifier signal is not zero between x-ray pulses. This can be seen from Equation 6-3. When the slope is not zero, the mean values of the two sums will differ because they are taken over regions separated in time by $L+G$, on average. Such non-zero slopes can arise from various causes, of which the most common is detector leakage current.

When the mean baseline value is not zero, it must be determined and subtracted from measured peak values in order to determine V_x values accurately. If the error introduced by this subtraction is not to significantly increase σ_t , then the error in the baseline estimate σ_b must be small compared to σ_e . Because the error in a single baseline measurement is σ_e , by definition, this means that multiple baseline measurements will have to be averaged. This number, N_B is the Baseline Average. For example, if $N_B = 128$ measurements are averaged then the total noise will be as shown in Equation 6-6.

$$\sigma_t = \text{sqrt}(\sigma_f^2 + (1+1/128)\sigma_c^2)$$

Equation 6-6

This results in less than 0.5 eV degradation in resolution, even for very long peaking times, when resolutions of order 130 eV are obtained.

6.4.2 Raw Baseline Measurement

The output of the energy filter (or a derivative of the energy filter, the *intermediate filter*) is sampled periodically in the explicit absence of an x-ray step, defined by a baseline threshold. In practice, the DXP initially makes a series of N_B baseline measurements to compute a starting baseline mean. It then makes additional baseline measurements at quasi-periodic intervals to keep the estimate up to date. These values are stored internally and can be read out to construct a spectrum of baseline noise, referred to as the Baseline Histogram. This is recommended because of its excellent diagnostic properties. When all components in the spectrometer system are working properly, the baseline spectrum should be Gaussian in shape with a standard deviation reflecting σ_n . Deviations from this shape indicate various pathological conditions which may also cause the x-ray spectrum to be distorted and therefore have to be fixed.

*Applying a **Baseline Cut** can improve performance when the Baseline Histogram is non-Gaussian. Outlying data points are 'cut' from the running **Baseline Average** (though still included in the histogram)*

The situation is remedied by removing (“cutting”) outlying samples from the baseline average described below. If the maximum in the baseline distribution lies at E_0 , then captured baseline values that deviate from E_0 by more than ΔE^+ and ΔE^- , respectively, are not included in the running baseline average. Note that *all captured baseline values are included in the Baseline Histogram*, however, so that it is always a valid representation of the system’s behavior.

6.4.3 Baseline Average Settings and Recommendations

A FIR running average of baseline measurements is computed, which is then subtracted from sampled peak values to compute the energy of corresponding incident x-rays. The number of baseline samples averaged is set in xManager as “Baselines Average Samples”. In the DSP this is converted into the parameter BLAVGDIV according to the equation:

$$\# \text{ baseline samples averaged} = 2^{(\text{BLAVGDIV} + 1)}$$

Decimation	# Baseline Samples to Average	BLAVGDIV (DSP Parameter)
0	64	5
2	128	6
4	256	7
6	256	7

Table 6.2: Typical values used for baseline averaging. The best value for each decimation should be determined empirically, though the general trend illustrated in the table, i.e. larger number to average for higher decimations, should be followed.

6.4.4 Why Use a Finite Averaging Length?

Physically, the baseline is a measure of the instantaneous slope (volts/sec) for a pulsed-reset detector, and a measure of the DC offset for an RC-feedback preamplifier. The variation in leakage current of the detector and offset drift and 1/f noise of the preamplifier often contribute to a baseline with significant low-frequency (i.e. relative to the energy filter cutoff) noise. These variations pass through the energy filter, and thus should also pass through the baseline averaging stage to achieve good cancellation when the baseline average is subtracted from the energy filter sample. The goal is to produce a baseline average that has a sufficient number of samples to average out the high frequency noise, but which still reflects the 'local' instantaneous baseline upon which the x-ray step 'rides'. Generally speaking, the number of baseline samples in the average is set to achieve the best energy resolution performance over the desired range of input count rate. There are two considerations worth emphasizing:

1. *Excess detector/preamplifier noise and pickup (all decimations):* The values in the table above implicitly assume a flat noise spectrum from the preamplifier. A high-frequency noise peak can result in poor relative performance at the corresponding 'resonant' peaking time. Often this problem can be mediated, though not eliminated, by *increasing* the number of baseline samples in the average for the affected peaking times. On the other hand, excess low-frequency noise, i.e. wandering, can be remedied by *reducing* the number baseline samples in the average.
2. *High rate performance (decimation 0):* At higher rates, i.e. > 50% deadtime, the slow filter returns less and less often to baseline, thus the time between baseline samples grows longer. This is the primary cause of degraded energy resolution at high rates. Decimation 2,4 and 6 firmware now employs a proprietary circuit that virtually eliminates this problem, resulting in industry-leading count rate stability. This improvement cannot however be implemented in the decimation 0 firmware. The resolution can nonetheless be improved in most cases by *reducing* the number of baseline samples in the average.

6.5 X-ray Detection & Threshold Setting:

Before capturing a value of V_x we must first detect the x-ray. X-ray steps (in the preamp output) are detected by digitally comparing the output of a trapezoidal filter to a threshold.

In the DXP up to three trapezoidal filters are implemented: *fast*, *intermediate* and *slow*; each with a threshold that can be individually enabled or disabled. A fast filter very quickly detects larger x-ray steps. A slow (energy) filter averages out the most noise and can thus detect smaller x-ray steps, but has a response that is much slower. An intermediate filter (used in decimations 2, 4 and 6 only) is a derivative of the slow filter that provides a balance between the speed of the fast filter and the noise reduction of the slow filter.

The fast filter is used solely for x-ray detection, i.e. a threshold crossing initiates event processing. Its short basewidth ($2L+G$) means that successive pulses that would 'pile-up' in the slow filter can be resolved in the fast filter and rejected from the spectrum (see Figure 6.11 below). Conversely, little noise

reduction is achieved in the fast filter, thus the fast threshold cannot be set to detect particularly low x-ray energies.

The intermediate filter is used for reset-type preamplifiers, in decimations 2, 4 and 6 only. Its threshold is automatically set by the DSP and applied as part of the baseline acquisition circuitry, i.e. baseline measurements are taken when the signal is *below* this threshold. Intermediate threshold crossings by default also trigger event processing, extending the detectable energy range significantly below the fast filter threshold.

After an x-ray has been detected, the step height is measured at the slow filter output. The slow filter's excellent noise reduction also allows for *detection* of the very lowest energy x-rays however its slow response precludes accuracy both in the determination of pulse pileup and the measurement of deadtime. The intermediate filter, which does not suffer this loss of accuracy, typically provides sufficient low energy detection. When present the intermediate threshold is enabled by default, and should be used in most cases. The slow threshold should be used cautiously, and only at low rates.

6.6 Peak Capture Methods

As noted above, we wish to capture a value of V_x for each x-ray detected and use these values to construct a spectrum. This process is also significantly different between digital and analog systems. In the analog system the peak value must be "captured" into an analog storage device, usually a capacitor, and "held" until it is digitized. Then the digital value is used to update a memory location to build the desired spectrum. During this analog to digital conversion process the system is dead to other events, which can severely reduce system throughput. Even single channel analyzer systems introduce significant deadtime at this stage since they must wait some period (typically a few microseconds) to determine whether or not the window condition is satisfied.

Digital systems are much more efficient in this regard, since the values output by the filter are already digital values. All that is required is to capture the peak value – it is immediately ready to be added to the spectrum. If the addition process can be done in less than one peaking time, which is usually trivial digitally, then no system deadtime is produced by the capture and store operation. This is a significant source of the enhanced throughput found in digital systems.

Once an active threshold is exceeded, the DXP-XMAP employs one of two methods to capture the slow energy filter output such that the best measure of V_x results. For decimations 2,4 and 6 the slow filter output is monitored over a finite interval of time in the region of its maximum, and the *maximum value within that interval* is captured. This method is referred to as "peak finding" or "max capture". For decimation 0, the slow filter is *sampled at a fixed time interval* after the pulse is detected by the fast filter. This method is referred to as "peak sampling".

After describing in section 6.6.1 below how to set the Gap parameter so that there will be a quality value of the energy filter to capture, we describe the two methods in detail in section 6.6.2.

6.6.1 Setting the Gap Length

When starting with a new detector, it is important first to set SLOWGAP to a minimum of 3, and *at least one unit greater than* the smallest value, in decimated clock cycles (see Table 6.3), that encloses the entire preamplifier risetime, per section 4.7.1.2.

Decimation	# ADC Samples averaged	Decimated Clock frequency	Decimated Clock cycle interval	Peaking Time Range (in μ s)
0	1	50 MHz	20 ns	0.08– 0.50
2	4	12.5 MHz	80 ns	0.5 – 8.0
4	16	3.125 MHz	320 ns	8 -32
6	64	781.25 kHz	1.28 μ s	32 -128

Table 6.3: For decimation 0 the slow filter output is sampled a fixed time after the x-ray is detected. PEAKSAM must be set properly to achieve optimum performance.

For example, consider a preamplifier with a pulse risetime of 260ns. For decimations 4 and 6 SLOWGAP would be set to 3 or greater. For decimation 2 SLOWGAP would be 4 or greater, and for decimation 0 SLOWGAP would be set to 13 or greater. xManager will select these values automatically if you enter a Gap Time of 260 ns. SLOWGAP is independent of SLOWLEN, thus all peaking times having a common decimation can share the same SLOWGAP value.

6.6.2 Peak Sampling vs. Peak Finding

The figures below illustrate the two peak capture methods. Under the 'peak finding method' the slow filter output is monitored over a finite interval of time, and the *maximum value within that interval* is selected. The interval is set automatically, solely based on the values of the DXP parameters SLOWLEN and PEAKINT. SLOWLEN and PEAKINT are both automatically derived from the peaking time value selected in xManager and should normally not be adjusted by the user. PEAKINT is also a pileup inspection parameter, as will be discussed in further detail in section 6.8.

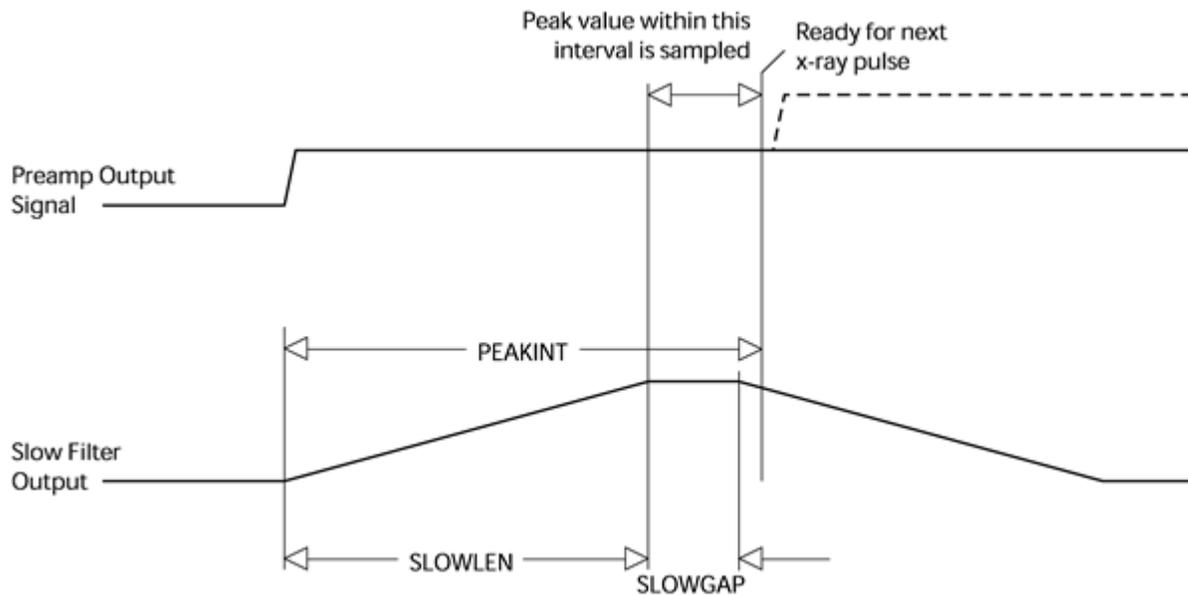


Figure 6.7: Peak finding method: The slow filter output is monitored and the peak value is selected.

In the 'peak sampling' method, the slow filter output is instead sampled a *fixed time* after the x-ray is detected. An additional 'Peak Sampling' timer is started when an x-ray step is detected which expires after PEAKSAM decimated clock cycles. PEAKSAM must be less than PEAKINT, and should typically be set such that the sample point lies in the 'flat-top' region of the slow filter output:

$$\text{SLOWLEN} \leq \text{PEAKSAM} \leq \text{SLOWLEN} + \text{SLOWGAP}$$

Equation 6-7

The precise PEAKSAM setting has a strong effect on energy resolution and should be determined empirically for each new detector. More on this below...

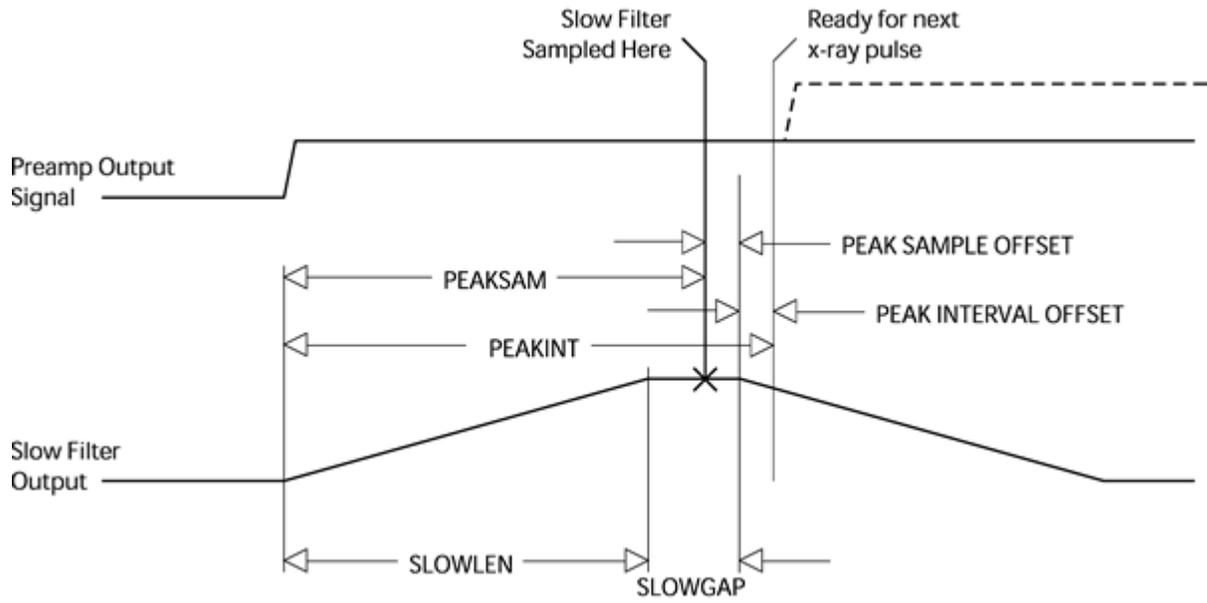


Figure 6.8: Peak sampling method: The slow filter output is sampled a fixed time after the x-ray is detected. PEAKSAMPLING must be set properly to achieve optimum performance.

In our experience values at the low end (i.e. PEAKSAMPLING ~ SLOWLEN) tend to work better. We recommend that you record the initial value of PEAKSAMPLING and then change it in steps of 1, working out from the initial value. Certain PEAKSAMPLING values may cause the DXP-XMAP to crash. Do not be alarmed, just restart and be sure to enter a valid PEAKSAMPLING value before proceeding. Making a plot of energy resolution versus PEAKSAMPLING will indicate the best value to select.

This determination need only be done for one peaking time per decimation. The result can then be applied to any value of SLOWLEN and SLOWGAP using the following recipe:

$$\text{PEAKSAMPLING} = (\text{SLOWLEN} + \text{SLOWGAP}) - X$$

Equation 6-8

6.7 Energy Measurement with Resistive Feedback Preamplifiers

In previous sections, the pulse height measurement was shown for the case of reset-type preamplifiers. The reset-type scheme is most often used for optimum energy resolution x-ray detectors. Other detectors use an RC-type preamplifier, as described in section 6.1.2. Resistive feedback is most often used for gamma-ray detectors which cover a larger dynamic range and where the electronic noise is not as significant a contribution to energy resolution.

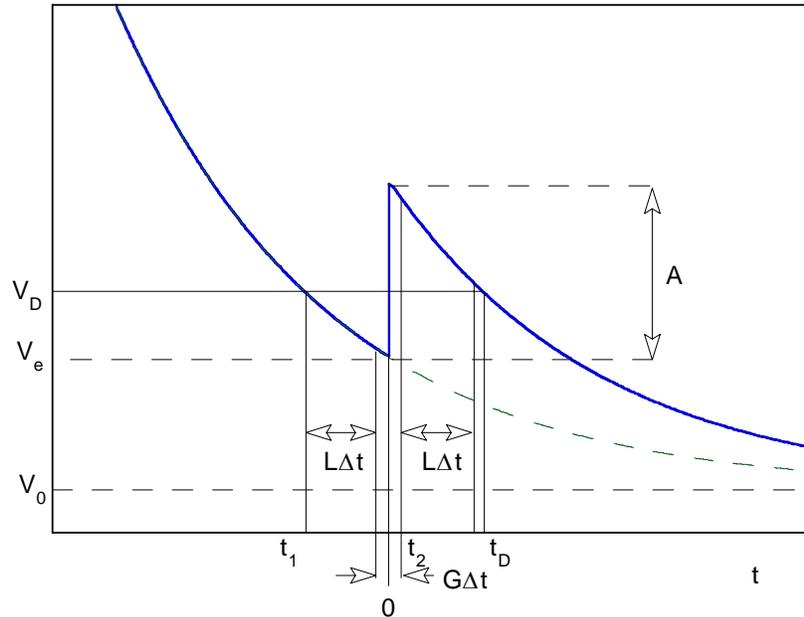


Figure 6.9: RC preamplifier output voltage. An x-ray step of amplitude A occurs at time $t=0$.

Where analog shaping amplifiers typically have a “pole-zero” adjustment to cancel out the exponential decay, the DXP uses a patented digital correction to achieve good energy resolution without a pole-zero stage. Figure 6.9 and Figure 6.10 illustrate the method used. The first shows the output voltage of a RC feedback preamplifier with a x-ray or γ -ray step of amplitude A appearing at $t=0$. V_e is the voltage just before the step pulse arrives and V_0 is the asymptotic value that the signal would decay to in the absence of steps. t_1 is the earliest time used in the slow filter, L and G are the length and gap of the trapezoidal filter in clock units, and Δt is the clock period. In addition to the normal slow filter measurement of the step height, the ADC amplitude, V_D is made at time t_D . In the following discussion, it is assumed that the signal rise-time is negligible.

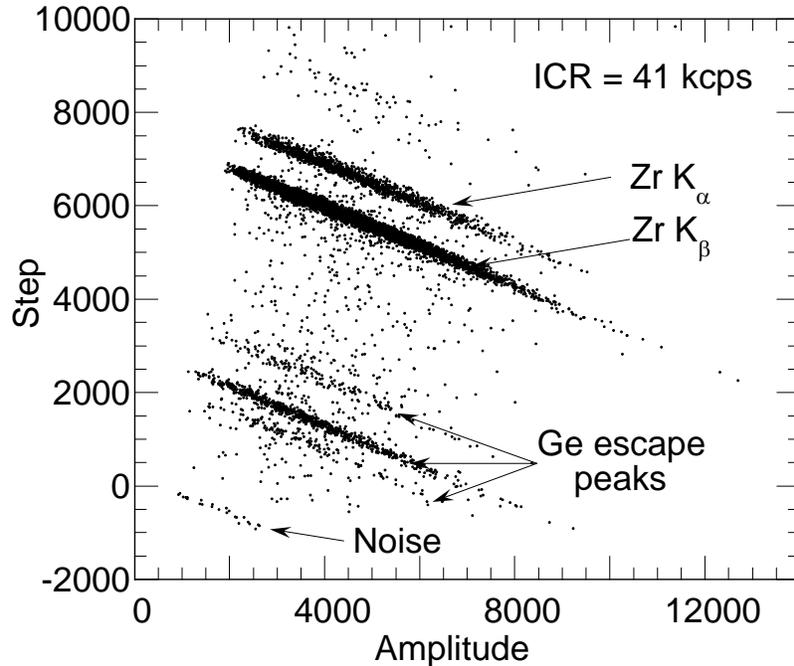


Figure 6.10: Correlation between step size and amplitude for Zr K_{α} x-ray events measured with the DXP-4C.

As Figure 6.10 makes clear, there is a linear correlation between the step height from the trapezoidal filter and the ADC amplitude, for pulses of a given energy. This is due to the fact that the exponential decay causes a deficit in the measured step height, which grows linearly with the distance from the asymptotic ADC offset at zero count rate.

The DSP reads these two values for each event that passes the FiPPI's trigger criteria, and makes a correction of the form:

$$E = k_1 (S_X + k_2 V_X - \langle S_B + k_2 V_B \rangle)$$

Equation 6-9

Here the quantities S_X and V_X are the step height and ADC amplitude measured for the step, and the corresponding values with the B subscript are "baseline" values, which are measured frequently at times when there is no trigger. The brackets $\langle \rangle$ indicate that the baseline values are averaged over a large enough number of events to not introduce additional noise in the measurement. The constant k_2 (the DSP parameter called RCFCOR) is inversely proportional to the exponential decay time; this correction factor is a constant for a detector channel at a fixed gain and shaping time. The constant k_1 is effectively a gain factor, and is taken into account with a detector gain calibration.

The parameter RCFCOR is a function of the digital filter parameters (SLOWLEN, SLOWGAP and DECIMATION) and the preamplifier decay time (the DSP parameter RCTAU). The decay time defined by RCTAU (and fractional word RCTAUFRACTION) has 50 ns granularity, and is measured and

entered by the user. At the start of an acquisition run, the DSP calculates RCFCOR using the following approximate expression:

$$\text{RCFCOR} = 2^{\text{DEC}} * (\text{LEN} + \text{GAP}) / (\text{RCTAU} - (\text{LEN} + \text{GAP}/2 + 3) * 2^{\text{DEC}})$$

Equation 6-10

The above expression is valid for peaking times less than about RCTAU/2. Alternatively, RCFCOR can be determined empirically in a special test run from a linear fit of data, as in Figure 6.10.

6.8 Pile-up Inspection:

The captured value V_x (see Figure 6.6) will only be a valid measure of its associated x-ray's energy provided that its filtered pulse is sufficiently well separated in time from its preceding and succeeding neighbor pulses so that its peak amplitude is not distorted by the action of the trapezoidal filter on those neighbor pulses. That is, if the pulse is not *piled up*. The relevant issues may be understood by reference to Figure 6.11, which shows 5 x-rays arriving separated by various intervals.

Because the triangular filter is a linear filter, its output for a series of pulses is the linear sum of its outputs for the individual members in the series. In Figure 6.11 the pulses are separated by intervals of 3.2, 1.8, 5.7, and 0.7 μs , respectively. The fast filter has a peaking time of 0.4 μs with no gap. The slow filter has a peaking time of 2.0 μs with a gap of 0.4 μs .

The first kind of pileup is *slow pileup*, which refers to pileup in the slow channel. This occurs when the rising (or falling) edge of one pulse lies under the peak (specifically the sampling point) of its neighbor. Thus peaks 1 and 2 are sufficiently well separated so that the leading edge (point 2a) of peak 2 falls after the peak of pulse 1. Because the trapezoidal filter function is symmetrical, this also means that pulse 1's trailing edge (point 1c) also does not fall under the peak of pulse 2. For this to be true, the two pulses must be separated by at least an interval of $L + G/2$. Peaks 2 and 3, which are separated by only 1.8 μs , are thus seen to pileup in the present example with a 2.0 μs peaking time.

This leads to an important first point: whether pulses suffer slow pileup depends critically on the peaking time of the filter being used. The amount of pileup which occurs at a given average signal rate will increase with longer peaking times. We will quantify this in section 0, where we discuss throughput.

Because the fast filter peaking time is only 0.4 μs , these x-ray pulses do not pileup in the fast filter channel. The DXP can therefore test for slow channel pileup by measuring for the interval PEAKINT after a pulse arrival time. If no second pulse occurs in this interval, then there is no trailing edge pileup. PEAKINT is usually set to a value close to $L + G/2 + 1$. Pulse 1 passes this test, as shown in the figure. Pulse 2, however, fails the PEAKINT test because pulse 3 follows in 1.8 μs , which is less than $\text{PEAKINT} = 2.3 \mu\text{s}$. Notice, by the symmetry of the trapezoidal filter, if pulse 2 is rejected because of pulse 3, then pulse 3 is similarly rejected because of pulse 2.

Pulses 4 and 5 are so close together that the output of the fast filter does not fall below the threshold between them and so they are detected by the pulse detector as only being a single x-ray pulse. Indeed, only a single (though

somewhat distorted) pulse emerges from the slow filter, but its peak amplitude corresponds to the energy of neither x-ray 4 nor x-ray 5. In order to reject as many of these fast channel pileup cases as possible, the DXP implements a fast channel pileup inspection test as well.

The fast channel pileup test is based on the observation that, to the extent that the risetime of the preamplifier pulses is independent of the x-rays' energies (which is generally the case in x-ray work except for some room temperature, compound semiconductor detectors) the basewidth of the fast digital filter (i.e. $2L_f + G_f$) will also be energy independent and will never exceed some maximum width MAXWIDTH. Thus, if the width of the fast filter output pulses is measured at threshold and found to exceed MAXWIDTH, then fast channel pileup must have occurred. This is shown graphically in the figure where pulse 3 passes the MAXWIDTH test, while the piled up pair of pulses 4 and 5 fail the MAXWIDTH test.

Thus, in Figure 6.11, only pulse 1 passes both pileup inspection tests and, indeed, it is the only pulse to have a well defined flattop region at time PEAKSAMP in the slow filter output.

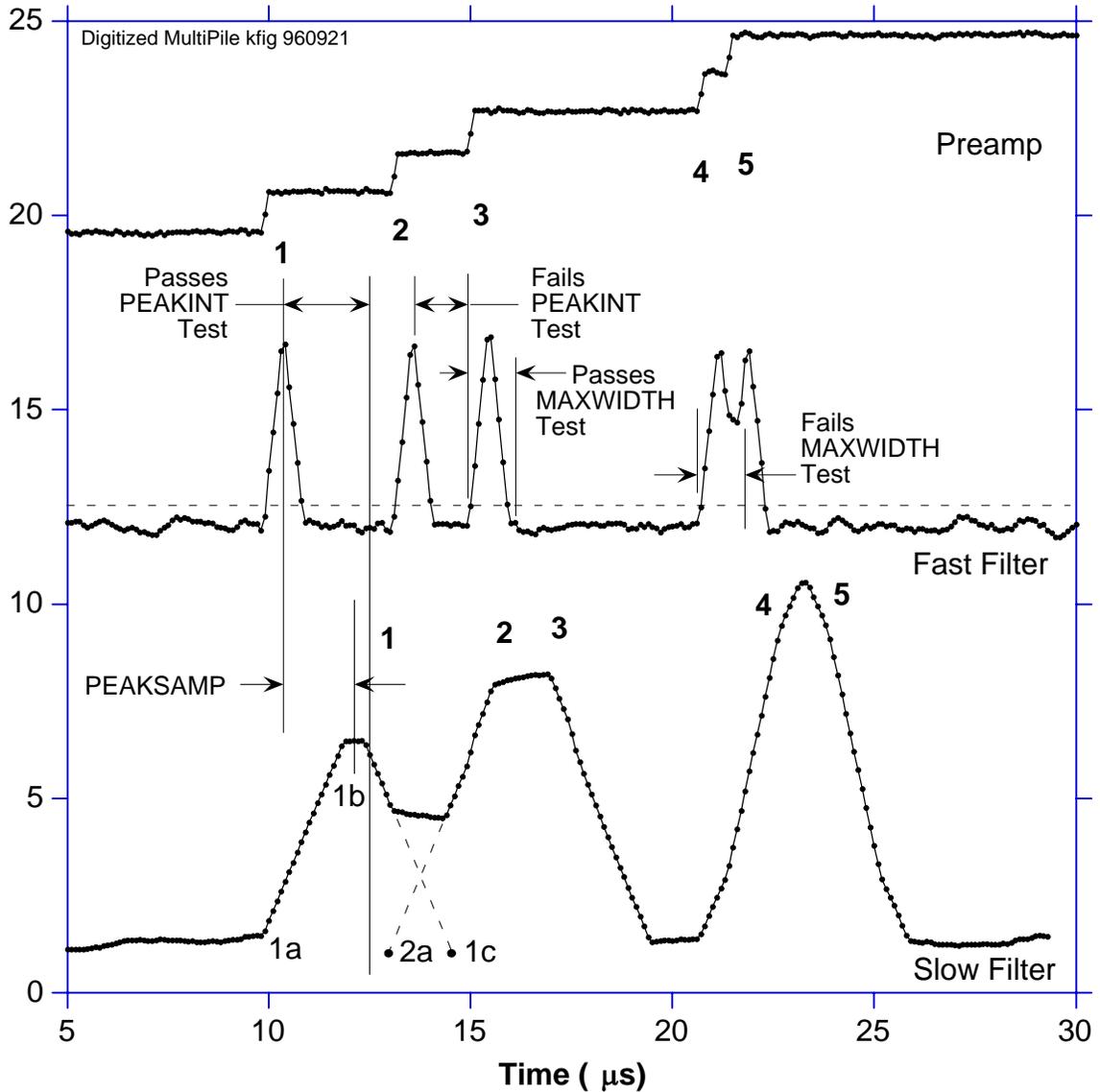


Figure 6.11: A sequence of 5 x-ray pulses separated by various intervals to show the origin of both slow channel and fast channel pileup and demonstrate how the two cases are detected by the DXP.

Note that PEAKINT and MAXWIDTH are both DSP parameters and are normally set automatically. In particular, there is almost never any benefit to a longer value of PEAKINT than the standard value as it does not improve energy resolution and only decreases throughput for a given input rate. Please see section 4.6.2.1 for details on how to adjust MAXWIDTH.

6.9 Input Count Rate (ICR) and Output Count Rate (OCR):

During data acquisition, x-rays will be absorbed in the detector at some rate. This is the *true input count rate*, which we will refer to as ICR_t . Because of fast channel pileup, not all of these will be detected by the DXP's x-ray pulse detection circuitry, which will thus report a *measured input count rate* ICR_m , which will be less than ICR_t . This phenomenon, it should be noted, is a

characteristic of all x-ray detection circuits, whether analog or digital, and is not specific to the DXP.

Of the detected x-rays, some fraction will also satisfy both fast and slow channel pileup tests and have their values of V_x captured and placed into the spectrum. This number is the *output count rate*, which we refer to as the OCR. The DXP normally returns, in addition to the collected spectrum, the REALTIME for which data was collected, the fast channel LIVETIME for which the fast channel was below threshold (and thus ready to detect a subsequent x-ray) together with the number FASTPEAKS of fast peaks detected and the number of V_x captured events EVTSINRUN. From these values, both the OCR and ICR_m can be computed according to Equation 6-11. These values can then be used to make deadtime corrections as discussed in section 6.11.

$$ICR_m = \text{FASTPEAKS}/\text{LIVETIME}; \quad \text{OCR} = \text{EVTSINRUN}/\text{REALTIME}$$

Equation 6-11

Note: The fast channel LIVETIME should only be used to determine the input count rate according to Equation 6-11. Specifically, it is NOT related to the energy filter livetime and should not be interpreted as the inverse of the processor deadtime. The DSP *does* calculate the energy filter livetime ELIVETIME, however, it is only an approximation. The most accurate deadtime measurement is obtained from ICR_m and OCR in Equation 6-11, as discussed in section 6.11.

6.10 Throughput:

Figure 6.12 shows how the values of ICR_m and OCR vary with true input count rate for the DXP and compare these results to those from a common analog shaping amplifier plus SCA system. The data were taken at a synchrotron source using a detector looking at a CuO target illuminated by x-rays slightly above the Cu K absorption edge. Intensity was varied by adjusting two pairs of crossed slits in front of the input x-ray beam so that the harmonic content of the x-ray beam striking the detector remained constant with varying intensity.

NOTE: The DXP's peaking time is twice as long as the analog system peaking time in this comparison, and yet the throughput is nearly the same.

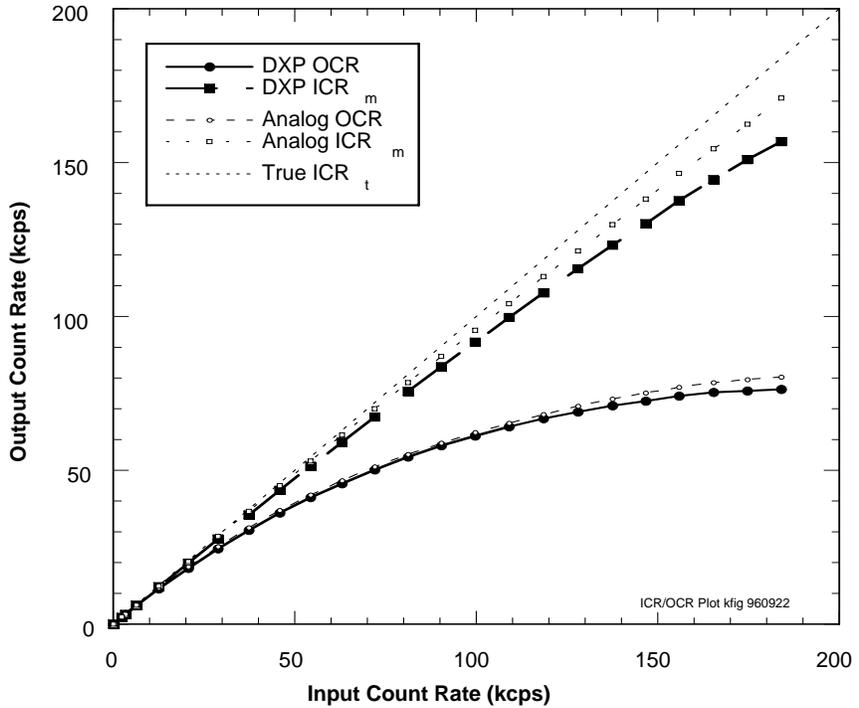


Figure 6.12: Curves of ICR_m and OCR for the DXP using 2 μs peaking time, compared to a common analog SCA system using 1 μs peaking time.

System	OCR Deadtime (μs)	ICR Deadtime (μs)
DXP (2 μs τ _p , 0.6 μs τ _g)	4.73	0.83
Analog Triangular Filter Amp (τ _p = 1 μs)	4.47	0.40

Table 6.4: Comparing the deadtime per event for the DXP and an analog shaping amplifier. Notice that that the DXP produces a comparable output count rate even though its peaking time is nearly twice as long.

Functionally, the OCR in both cases is seen to initially rise with increasing ICR and then saturate at higher ICR levels. The theoretical form, from Poisson statistics, for a channel which suffers from paralyzable (extending) dead time is given by:

$$OCR = ICR_t * \exp(-ICR_t * \tau_d),$$

Equation 6-12

where τ_d is the *dead time*. Both the DXP and analog systems' OCRs are so describable, with the *slow channel dead times* - τ_d - shown in Table 6.4. The measured ICR_m values for both the DXP and analog systems are similarly describable, with the *fast channel dead times* - τ_{df} - as shown. The maximum value of OCR can be found by differentiating Equation 6-12 and setting the result to zero. This occurs when the value of the exponent is -1, i.e. when ICR_t equals 1/τ_d. At this point, the maximum OCR_{max} is 1/e multiplied by the ICR, or:

$$\text{OCR}_{\text{max}} = 1/(e \tau_d) = 0.37/\tau_d$$

Equation 6-13

These are general results and are very useful for estimating experimental data rates.

Table 6.4 illustrates a very important result for using the DXP: the slow channel deadtime is nearly the minimum value that is theoretically possible, namely the pulse basewidth. For the shown example, the basewidth is 4.6 μs ($2L_S + G_S$) while the deadtime is 4.73 μs . The slight increase is because, as noted above, PEAKINT is always set slightly longer than $(L_S + G_S)/2$ to assure that pileup does not distort collected values of V_X .

The deadtime for the analog system, on the other hand is much larger. In fact, as shown, the throughput for the digital system is almost twice as high, since it attains the same throughput for a 2 μs peaking time as the analog system achieves for a 1 μs peaking time. The slower analog rate arises, as noted earlier both from the longer tails on the pulses from the analog triangular filter and on additional deadtime introduced by the operation of the SCA. In spectroscopy applications where the system can be profitably run at close to maximum throughput, then, a single DXP channel will then effectively count as rapidly as two analog channels.

6.11 Dead Time Corrections:

The fact that both OCR and ICR_m are describable by Equation 6-12 makes it possible to correct DXP spectra quite accurately for deadtime effects. Because deadtime losses are energy independent, the measured counts N_{mi} in any spectral channel i are related to the true number N_{ti} which would have been collected in the same channel i in the absence of deadtime effects by:

$$N_{ti} = N_{mi} \text{ICR}_t / \text{OCR}$$

Equation 6-14

Looking at Figure 6.12, it is clear that a first order correction can be made by using ICR_m of Equation 6-11 instead of ICR_t , particularly for OCR values less than about 50% of the maximum OCR value. For a more accurate correction, the fast channel deadtime τ_{df} should be measured from a fit to the equation:

$$\text{ICR}_m = \text{ICR}_t * \exp(-\text{ICR}_t \tau_{df})$$

Equation 6-15

Then, for each recorded spectrum, the associated value of ICR_m is noted and Equation 6-15 inverted (there are simple numerical routines to do this for transcendental equations) to obtain ICR_t . Then the spectrum can be corrected on a channel by channel basis using Equation 6-12. In experiments with a DXP prototype, we found that, for a 4 μs peaking time (for which the maximum ICR is 125 kcps), we could correct the area of a reference peak to better than 0.5% between 1 and 120 kcps.

7 DXP-XMAP Hardware Description

7.1 DXP-XMAP Overview

The DXP-XMAP system, shown in **Figure 7.1**, consists of four Digital Xray Processor (DXP) channels, a Digital Signal Processor (DSP), a System FPGA, SRAM memory and a PCI interface. Each of the four DXP channels accepts a preamplified signal input and produces a 16-bit pipelined output stream of x-ray energies.

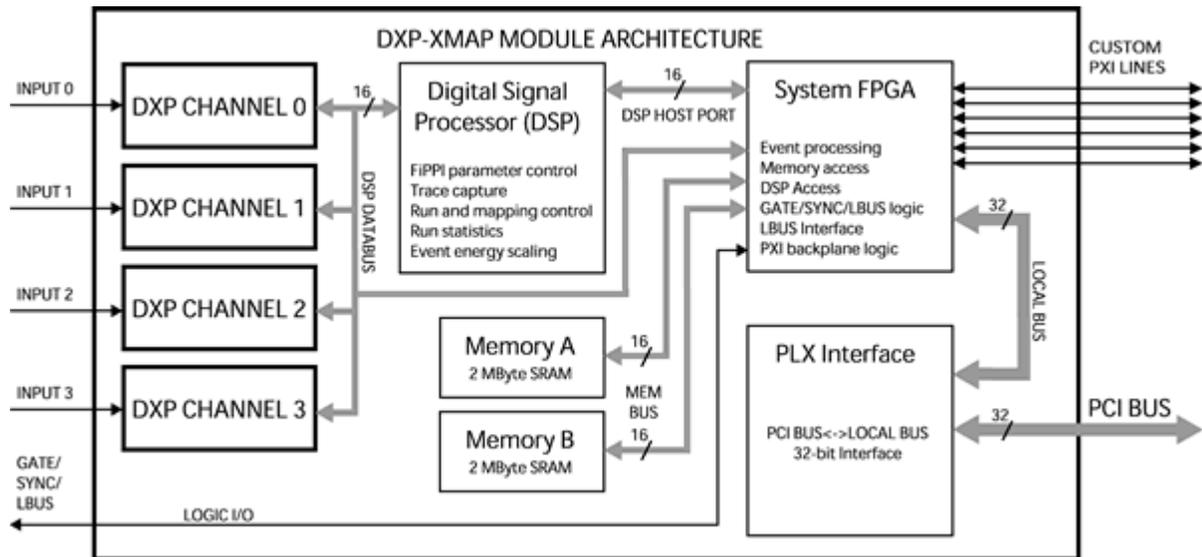


Figure 7.1: Block diagram of the XMAP system architecture.

7.1.1 Four Independent DXP Channels

The Digital X-ray Processor (DXP) is a proprietary architecture (see **Figure 7.2**) designed to rapidly measure the pulse-heights of voltage steps. Shown below are the three major DXP operating blocks: the Analog Signal Conditioner (ASC), an analog-to-digital converter (ADC), and the Filter, Peak Detector, and Pileup Inspector (FiPPI).

The DXP-XMAP contains four independent DXP Digital Xray Processor (DXP) channels. Each DXP channel accepts a detector preamplifier signal input and produces a FIFO-buffered output stream of 16-bit pulse-height, i.e. energy, measurements. Each DXP also keeps track of its run statistics, including livetime and the number of input counts detected. See sections 7.2-7.8 below for a thorough description of the DXP.

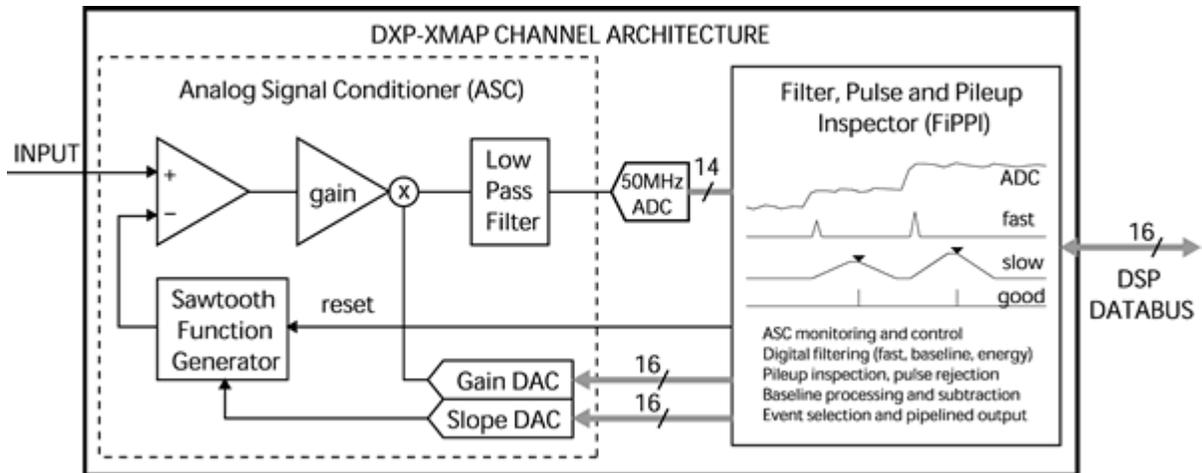


Figure 7.2: Block diagram of the DXP channel architecture.

7.1.2 Rapid Data Readout

The DXP-XMAP was designed for the rapid readout of acquired data in a multi-element detector system. Some important changes to the architecture, relative to other DXP products, facilitate this goal:

- 1) Tasks previously handled by the DSP, such as baseline handling and ASC control, have been offloaded to programmable logic in the FiPPIs. This frees the DSP to focus on event handling during data acquisition.
- 2) Memory management, previously handled by the DSP, is now tasked to programmable logic in the System FPGA. This reduces the size of the DSP's event handling loop.
- 3) The storage of MCA, SCA and statistics data in external SRAM memory. Previously this data was stored in DSP internal memory.
- 4) The use of a high speed interface. The 32-bit PCI bus operates at 33MHz, providing for transfers in excess of 130 MBytes/second. This readout speed is critical for supporting the continuous mapping data acquisition mode, particularly in systems with multiple detector elements.

7.2 Timing and Synchronization Logic

The front panel LEMO connector provides external access to the backplane PXI trigger lines. These lines, shared across all slots in each PCI bus segment, are used for timing and synchronization in multi-module DXP-XMAP systems. For more information about PCI bus segments and the PXI bus see section 2.1.2.

7.2.1 Connections Overview

One external (LEMO) TTL/CMOS connection is required for each trigger line used in each PCI bus segment. The module that accepts the LEMO connection must be configured via software as a 'Master' for that timing signal; a Master drives the designated signal onto the backplane, but also receives other backplane lines when applicable. The front-panel LED corresponding to the

chosen timing function is illuminated for each designated Master module. Other modules are automatically configured as 'Slaves', i.e. they receive all applicable backplane trigger lines.

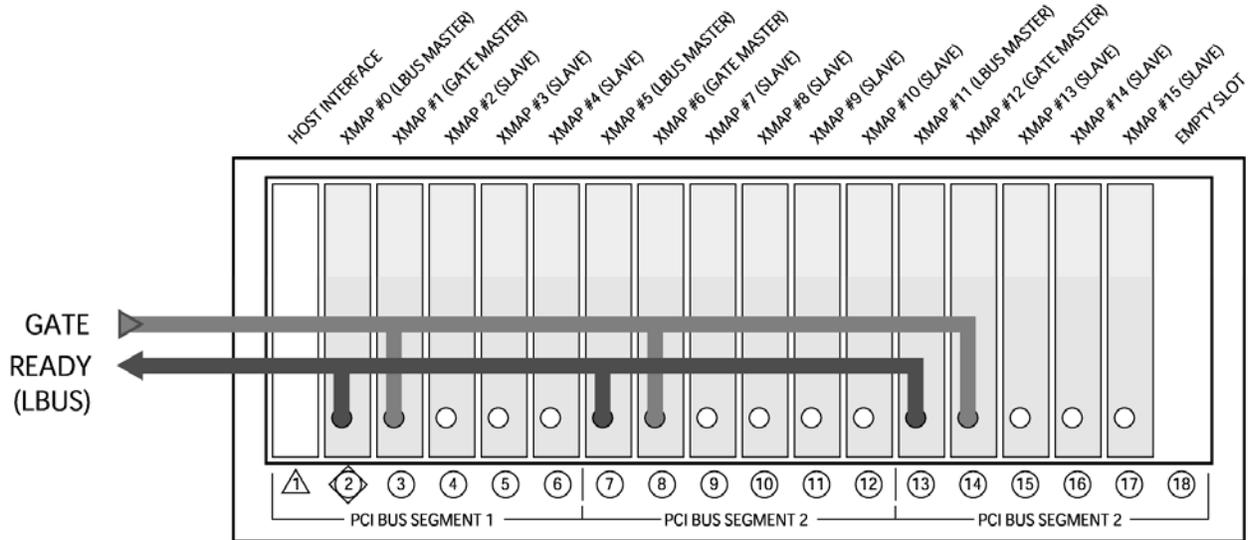


Figure 7.3: A 15-module XMAP system spanning 3 PCI bus segments that employs the LBUS and GATE timing functions. Note that an LBUS Master and a GATE Master are designated for each PCI bus segment.

7.2.2 LBUS (Run Start) Synchronization

A software command must be issued to each and every XMAP module to begin a data acquisition run, resulting in asynchronous operation in multi-module systems. RUN_PXI is a PXI backplane wired-OR trigger line that holds off data acquisition until all XMAP modules in the PXI bus segment have been initialized. A module designated as an LBUS Master provides a connection to this wired-OR line via the front panel LEMO connector. This serves two purposes: To complete the wired-OR connection between modules in separate bus segments via the front panel LEMO connector, and to provide a realtime 'READY' output. Properly implemented, LBUS ensures that connected modules start acquiring data simultaneously.

LBUS is often used in conjunction with GATE. The LBUS output connects to the user hardware that generates the GATE signal, is used as a READY constraint at the beginning of a run: The user generated GATE signal disables data acquisition until all such system constraints have been met

In the figure below, 'Start run' commands have been issued by the host to all XMAP modules. The transition from LO to HI on LBUS indicates that all XMAP modules are ready, and in fact have 'started' simultaneously, however, data acquisition does not actually begin until the GATE signal is released.

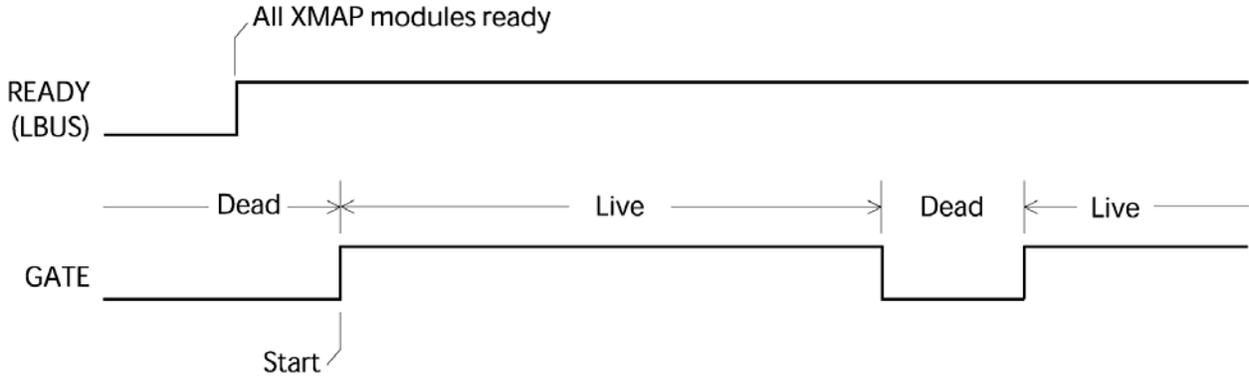


Figure 7.4: The beginning of a data acquisition run in a system that employs LBUS and GATE. 'Start Run' commands have been issued to all modules. LBUS goes HI indicating that all modules have been initialized, however, GATE disables data acquisition. User then releases GATE, resulting in a system-wide synchronized start.

7.2.3 GATE Function: MCA Mode

The GATE input allows realtime user control over data acquisition during a normal MCA data acquisition run. Data acquisition is halted when the GATE signal is asserted, i.e. incident events are not processed, and the realtime and livetime counters are disabled. The assert-polarity of the GATE signal can be set via software. The Realtime can be set to increment during GATE assertion via software. The GATE signal can be ignored entirely via software.

7.2.3.1 GATE Polarity

The interpretation of the user-provided GATE signal can be inverted in the hardware such that data acquisition is halted when the signal is HI or LO. The GATE polarity corresponds to the **input_logic_polarity** setting in the INI file.

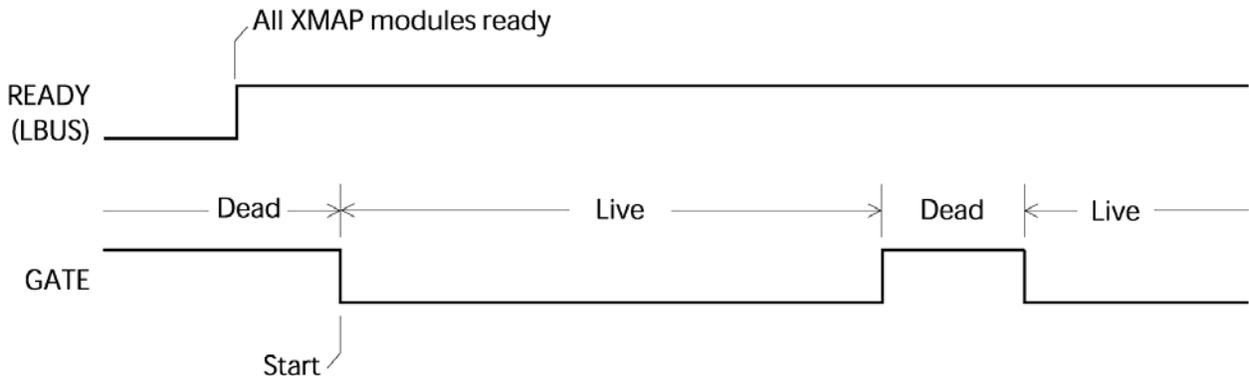


Figure 7.5: The same run start shown in Figure 7.4 above, but with the GATE polarity inverted.

7.2.3.2 Ignore GATE

This is a software-only setting. Checking the “Ignore GATE” checkbox disables the GATE logic: Data acquisition occurs irrespective of the GATE signal level.

7.2.3.3 GATE Real Time

This is a hardware-level setting that allows for incrementing only the real time counter when the GATE is asserted; data acquisition and other statistics are halted. This setting corresponds to the **gate_mode** setting in the INI file. The default setting **gate_mode = 0** halts everything. With **gate_mode = 1**, the real time increments.

7.2.4 GATE Function: Mapping Mode

In Mapping mode multiple spectra are generated as an x-ray beam is rastered across the sample. Each spectrum corresponds to a pixel. The 'pixel advance' controls when the XMAP changes from one pixel/spectrum to the next. The simplest pixel advance is implemented with the GATE signal.

7.2.4.1 Pixel Advance on GATE Edge

The recommended method for advancing the pixel number is to use the GATE input, where the pixel number advances on every trailing edge of the signal (the transition from active data acquisition to the inactive state). By default the GATE signal halts data acquisition when it is LO and the pixel advances on every falling edge (see **Figure 7.6**).

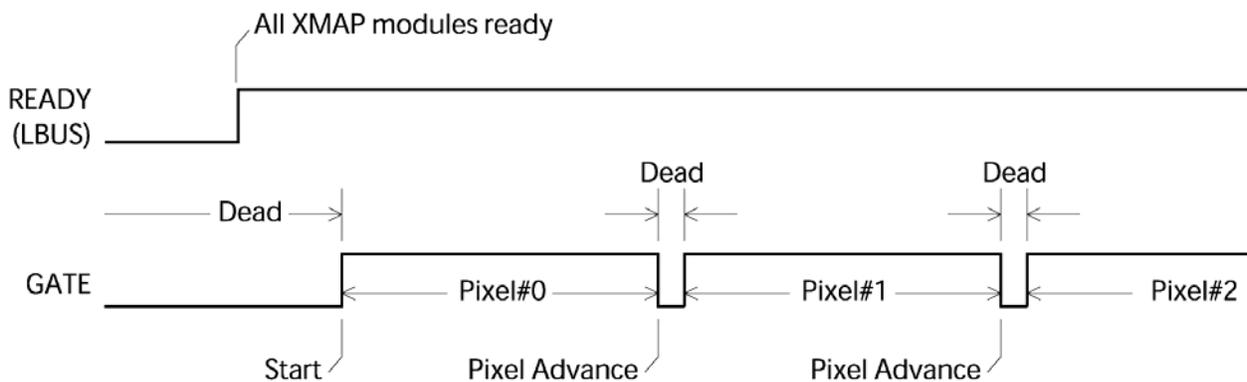


Figure 7.6: Mapping mode acquisition using the GATE input with default polarity. The pixel advance occurs on each falling edge of GATE and data acquisition is halted until the next rising edge.

7.2.4.2 GATE Polarity

The interpretation of the user-provided GATE signal can be inverted such that data acquisition is halted when the signal is HI and the pixel advance occurs on rising edges.

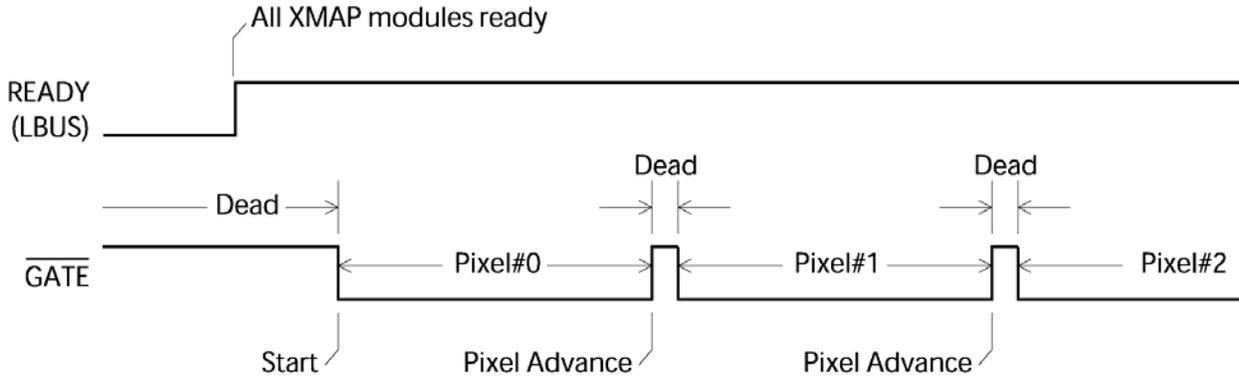


Figure 7.7: Mapping mode acquisition using the GATE input with inverted polarity. The pixel advance occurs on each rising edge of GATE and data acquisition is halted until the next falling edge.

7.2.4.3 GATE Ignore Setting

Normally the GATE signal is used both to advance the pixel and to halt data taking. In some cases the user may prefer continuous operation through the pixel advance operations. With the GATE Ignore option selected, the pixel advance occurs but data acquisition is not halted. Note that the data acquisition *is* halted at the beginning of the run until GATE is released the first time, i.e. run start synchronization is still available in this mode.

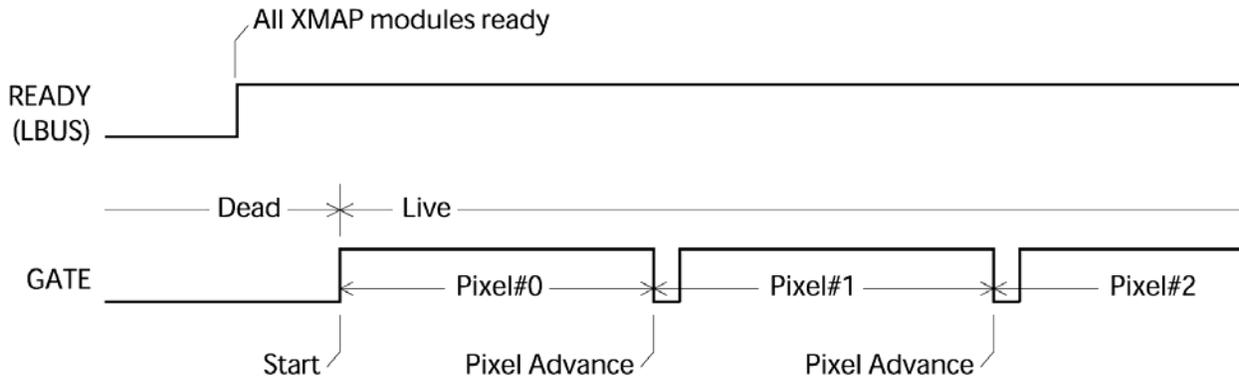


Figure 7.8: Mapping mode acquisition using the GATE input with default polarity, and with the GATE Ignore option selected. The pixel advance occurs on each falling edge of GATE but data acquisition runs continuously with no pause.

The above examples all use the same cable connections, shown in **Figure 7.3** above. Only one connection each for LBUS and GATE are necessary for each PCI bus segment.

7.2.5 SYNC Function: Mapping Mode

In Mapping mode multiple spectra are generated as an x-ray beam is rastered across the sample. Each spectrum corresponds to a pixel. The so-called

'pixel advance' controls when the XMAP changes from one pixel/spectrum to the next. A more complex pixel advance is implemented using the SYNC input.

7.2.5.1 Pixel Advance using SYNC Clock

The user provides a clock signal to the SYNC input(s). Using this method, the pixel will advance for every N LO-to-HI transitions on the SYNC line, where N can be set from 1 to 65535. Note that the pulse widths must be greater than 40 ns in duration.

7.2.5.2 Synchronous Starts with SYNC

It is especially important to synchronize the beginning of a run when using the SYNC pixel advance. Two methods are supported. The first edge detected on SYNC itself can be used (see **Figure 7.9**), where LBUS should be checked before enabling the SYNC clock. Alternatively GATE and LBUS can be used as before (see **Figure 7.10** and **Figure 7.11**).

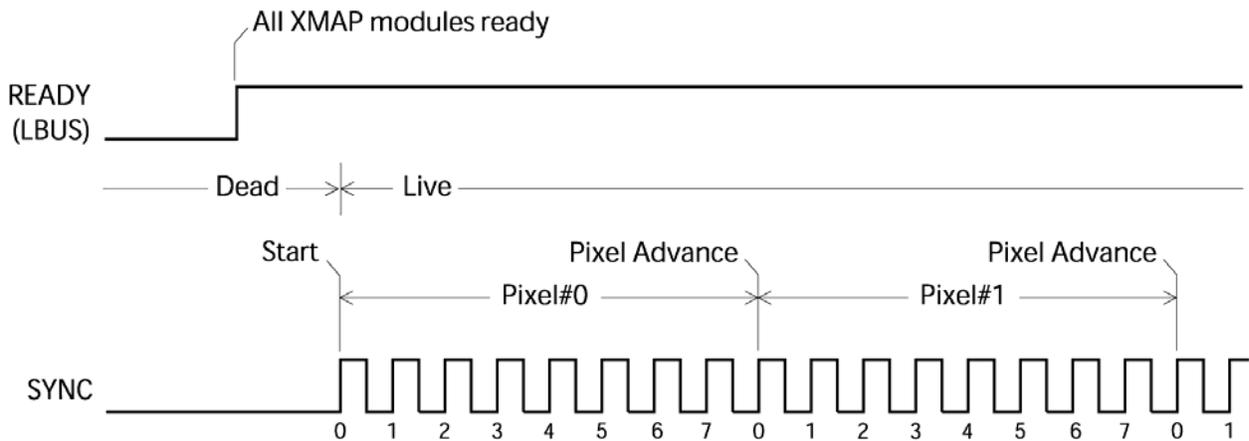


Figure 7.9: Mapping mode acquisition using the SYNC input with $N = 8$. The pixel advance occurs every 8 SYNC cycles. In this example GATE is not used, and SYNC itself is used to synchronize the run start: Data acquisition does not begin until the first rising edge is detected. User hardware must not start the SYNC clock until LBUS/READY is asserted HI.

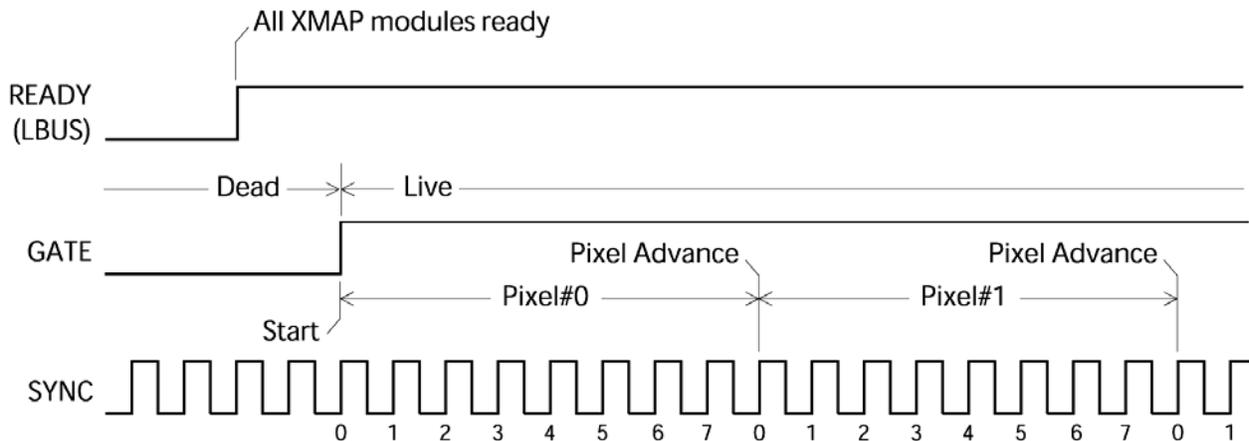


Figure 7.10: Mapping mode acquisition using the SYNC input with $N = 8$. The pixel advance occurs every 8 SYNC cycles. Note that LBUS and GATE are used to achieve a system-wide synchronous start as in previous figures.

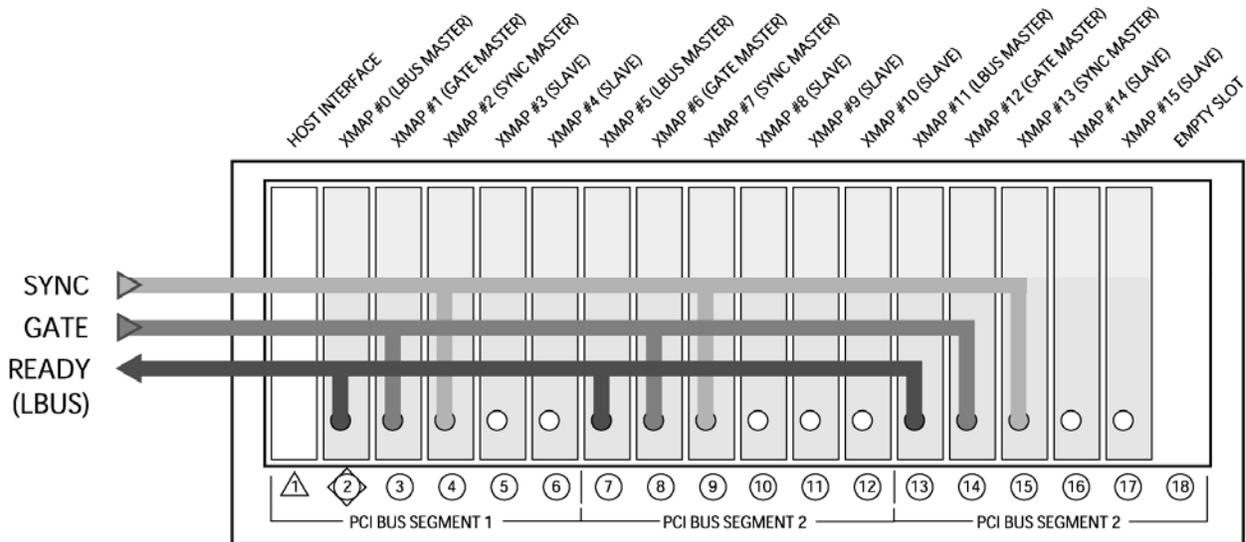


Figure 7.11: A 15-module XMAP system spanning 3 PCI bus segments that employs the SYNC, GATE and LBUS. Note that an LBUS Master, a GATE Master and a SYNC Master are designated for each PCI bus segment.

7.3 The Analog Signal Conditioner (ASC):

The ASC employs a patented ramp-subtraction technique to compress the dynamic range of the signal without adding significant noise or distortion to the signal. The typical preamplifier output signal has dynamic range of 100dB, or 16-bits. The compression produces a signal that can be fully digitized by a relatively inexpensive 14-bit ADC (with effectively 13-bits of resolution, due to integral and differential non-linearities). Importantly, the bandwidth of the signal is unaffected by the compression, limited only by the Nyquist criterion for the ADC. The digital filters thus operate on an wideband 20MHz signal.

The technique is illustrated in Figure 7.12. Here a resetting preamplifier output is shown which cycles between about -3.0 and -0.5 volts. We observe that it is not the large-signal ramp function which is of interest, but rather the individual steps, such as shown in Figure 6.1, that carry the x-ray amplitude information. Thus, if we generate a sawtooth function which has the same average slope and subtract this sawtooth from the preamplifier signal, we can amplify the difference signal to match the ADC's input range. The generator required to produce this sawtooth function is quite simple, comprising a current integrator with an adjustable offset. The current, which sets the slope, is controlled by a DAC (with DSP parameter SLOPEDAC). The DAC is automatically adjusted according the input rate to maintain the ASC output (i.e. the "Amplified Sawtooth Subtracted Data" of Figure 7.12) within the ADC input range.

A side-effect of this approach is that fluctuations in data arrival rate will cause the subtracted signal to pass outside the ADC input range while the

preamplifier output is in its operating range. These signal drifts out-of-range high or low have been termed DRIFTUPS and DRIFTDOWNS, respectively. The out-of-range condition is corrected by auto-zeroing the integrator, and thus bringing the signal to the middle of the ADC input range. Preamplifier resets are handled similarly, though the auto-zero time is extended to allow the preamplifier reset transient to settle. The preamplifier settling time (DSP parameter RESETINT) is thus an important DXP setting.

Any time the ADC is out of range, dead time is incurred. The ASC compression technique thus incurs additional deadtime, although this is typically much smaller than the deadtime caused by preamplifier resets.

Note: When viewing the ADC trace in oscilloscope mode, it is easy to mistake signal drift corrections for actual preamplifier resets.

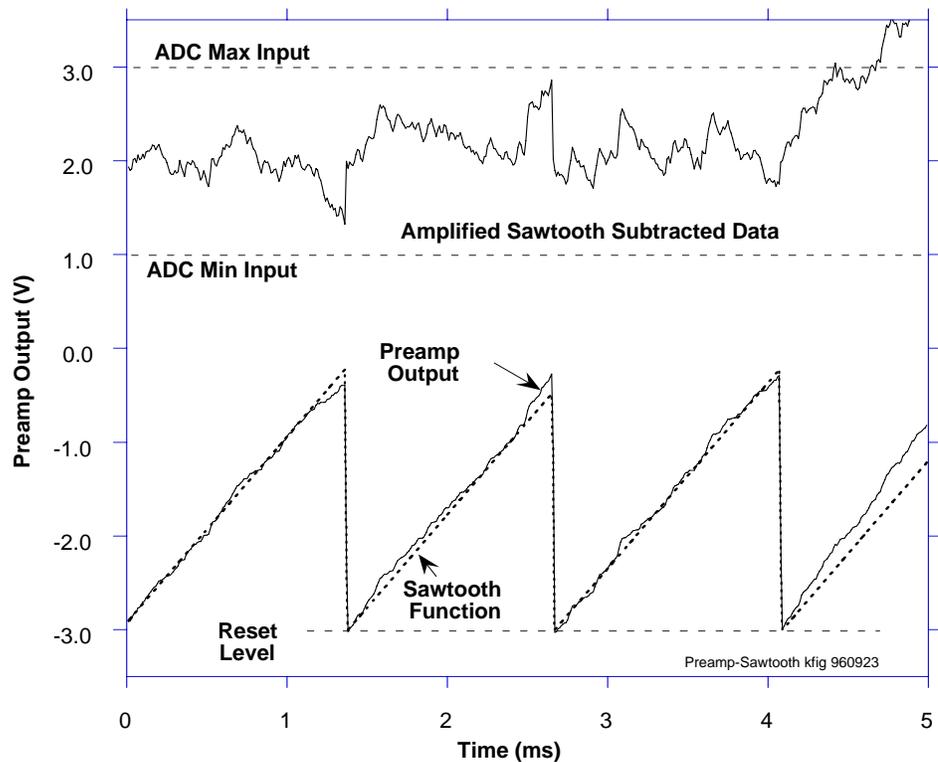


Figure 7.12: A sawtooth function having the same average slope as the preamp output is subtracted from it and the difference amplified and offset to match the input range of the ADC.

7.4 Analog to Digital Converter

Each DXP channel employs a 14-bit, 50MHz ADC to digitize the conditioned input signal. Unlike competing products, the DXP directly digitizes the wideband preamplifier signal.

7.5 The Filter, Pulse Detector, & Pile-up Inspector (FiPPI):

The FiPPI performs the various filtering, pulse detection and pileup inspection tasks discussed in sections 6.3 - 6.8. As described therein the FiPPI contains up to three digital trapezoidal filters: A fast filter for pulse detection and pileup inspection and event rejection; an intermediate filter for low energy, i.e. soft x-ray, event detection and baseline acquisition; a slow filter for pulse-height (energy) measurement. An output stream of accepted baseline-subtracted 16-bit x-ray event energies is stored in a FIFO pipeline that is periodically read out by the DSP. The FiPPI also measures live time, real time, and the number of detected (accepted *and* rejected) input events. In the DXP-XMAP, the FiPPI also takes on the tasks of ASC monitoring and control and baseline processing, reducing the work load of the DSP and thus improving the system throughput.

7.5.1 FiPPI Configuration

The FiPPI is implemented in a field-programmable-gate-array (FPGA). Configuration code for the DSP and FPGAs is contained in the XIA proprietary FDD firmware file. FiPPI code is downloaded to the XMAP hardware when:

- The hardware is initialized during startup.
- A new initialization (INI) file and/or FDD file is selected.
- The peaking time is adjusted such that a Decimation boundary is crossed.

7.5.2 FiPPI Version and Variants

Not all FiPPI configuration files are the same. FiPPIs are first distinguished by variant. Two standard FiPPI variants are available to all users: A variant for pulse-reset type preamplifiers and a variant for RC-decay type preamplifiers. From time to time improved standard FiPPI versions are released. In some cases custom FiPPI variant designs are used by certain customers. It is important to make sure that you are using the latest version of the appropriate FiPPI variant.

Note: RC-decay variant FiPPIs only have fast and slow filters, and the baseline is acquired from the slow filter.

7.5.3 FiPPI Decimation

FiPPI's are distinguished also by decimation. Decimation refers to pre-averaging of the ADC signal prior to the FPGA processing pipeline. Each decimation accommodates a specific range of peaking times, i.e. the shaping or integration time of the slow (energy) filter. Typically four (4) FiPPI configuration files are included in an FDD file. When the peaking time is changed such that a range boundary is crossed, the host software downloads the appropriate FiPPI configuration to the DXP-XMAP.

Note: Pulsed-reset variant FiPPIs with decimation 0 only have fast and slow filters, and the baseline is acquired from the slow filter.

7.5.4 Digital Trapezoidal Filtering

The digital trapezoidal filter produces a trapezoid response to a step input. A filter is defined by its length (ascending or descending time of the trapezoid), and its gap (flat-top time of the trapezoid). A constant input slope produces a constant DC offset; the bigger the slope, the bigger the offset. This offset is referred to as the baseline.

The peak value of the value of the trapezoid response to an input step, minus the baseline offset, is directly proportional to the height of the step, i.e. the x-ray energy. The goal of the FiPPI is thus to measure the baseline, to determine when to sample the peak of each trapezoid, and to subtract the baseline from the sampled peak.

7.5.4.1 Noise and Pileup

Detection occurs when the filter output crosses a constant threshold. Generally speaking, a longer filter length produces better noise reduction, and thus allows for a lower threshold. Pileup occurs when two or more successive pulses result in an output trapezoid with a peak height that is dependent on a combination of the input step heights. Piled-up events must be discarded. The minimum detectable pileup time, or pulse-pair resolution, is proportional to the filter length plus the filter gap. A tradeoff must therefore to be made between pulse pair resolution and the minimum x-ray energy that can be detected.

7.5.4.2 Fast (Trigger) Filter

The fast filter is used solely to detect incoming x-ray events and to determine whether a given event can be processed in the slow filter, or whether it should be rejected. The fast filter's peaking-time τ_p (τ_{pf}) can be adjusted from 40 ns to 2.56 μ s.

When τ_{pf} is 40 ns, the pulse pair resolution is typically less than 100 ns, however, x-rays in the 0eV - 2keV range may not be detected. When τ_{pf} is 1 μ s, x-rays with energies below 500 eV can be detected; however, the pulse pair resolution is greater than 2 μ s. We typically recommend running at $\tau_{pf} = 100$ ns, with the fast gap time set to zero.

7.5.4.3 Slow (Energy) Filter

The slow filter trapezoid peak value is sampled for the energy measurement. The slow filter's peaking-time τ_p (τ_{ps} , or simply referred to as *the* peaking time τ_p) can be adjusted from 100 ns to 164 μ s. At low input rates τ_{ps} should be set simply to optimize energy resolution. The optimal value will depend on the detector and preamplifier. At higher input rates, τ_{ps} should be chosen such that maximum throughput is achieved. The resulting improvement in statistics will compensate for any degradation of energy resolution.

The slow filter gap should always be longer than the response time of the preamplifier, i.e. the settling time of input x-ray steps. If the input signal displays a range of risetimes (as in the "ballistic deficit" phenomenon) the slow filter gap time should be extended to accommodate that range.

7.5.4.4 Intermediate (Baseline) Filter

Note: The intermediate filter is not included for decimation 0, pulsed-reset variant FiPPIs, and all RC-decay variant FiPPIs; the baseline is acquired from the slow filter.

7.5.5 Statistics

The FiPPI also includes a lifetime counter which counts the 50 MHz system clock, divided by 16, so that one “tick” is 320 ns. This counter is activated any time the DSP is enabled to collect x-ray pulse values from the FiPPI and therefore provides an extremely accurate measure of the system lifetime. In particular, as described in section 0, the DSP is not live either during preamplifier resets or during ASC out-of-ranges, both because it is adjusting the ASC and because the ADC inputs to the FiPPI are invalid. Thus the DXP measures lifetime more accurately than an external clock, which is insensitive to resets and includes them as part of the total lifetime. While the average number of resets/sec scales linearly with the countrate, in any given measurement period there will be fluctuations in the number of resets which may affect counting statistics in the most precise measurements.

All FiPPI parameters, including the filter peaking and gap times, threshold, and pileup inspection parameters are externally supplied and may be adjusted by the user to optimize performance. Because the FiPPI is implemented in a Xilinx field programmable gate array (FPGA), it may also be reprogrammed for special purposes, although this process is non-trivial and would definitely require XIA contract support.

<i>Decimation</i>	<i>#ADC Samples in Average</i>	<i>Peaking Time Range</i>	<i>'Shaping Time' Range</i>
0	1	80 ns – 500 ns	160 ns – 1.0 μs
2	4	500 ns – 8 μs	1.0 μs – 16 μs
4	16	8 μs – 32 μs	16 μs – 64 μs
6	64	32 μs – 128 μs	64 μs – 256 μs

Table 7.1: FiPPI decimation details.

7.6 The Digital Signal Processor (DSP):

The DXP-XMAP architecture was designed for speed. Because there is only one DSP for four DXP channels, the processor load has therefore been minimized. During a run, the DSP only processes events and compiles run statistics. Prior to a run, the DSP initializes the FiPPIs according to user settings. The DSP also performs various diagnostic tasks such as oscilloscope mode trace capture and baseline histogramming. Such diagnostic tasks are performed on one DXP channel at a time.

The processor is an Analog Devices ADSP-2183 16 bit Fixed-Point DSP optimized for fixed point arithmetic and high I/O rates. Different DSP program or code variants are used for different types of data acquisition.

7.6.1 Event Processing

Event processing consists of reading the event FIFOs of the DXP channels, scaling the raw value of each measured pulse-height to the appropriate spectrum bin number, and writing the scaled events to the System FPGA's event post-processor.

The ADSP-2183 has 16K words of 16-bit wide data memory and 16K words of 24-bit wide program memory, part of which is used as data memory to hold the MCA spectrum. (If more memory is required for special purposes, up to 4 Mbytes of extended memory can be added by specifying option M). Transferring data to/from these memory spaces is done through the DSP's built-in IDMA port, which does not interfere with the DSP program operation.

7.6.2 Statistics

Run statistics are updated periodically and written to the SRAM memory.

7.7 System FPGA

The System FPGA primarily serves as a bus interface and arbitrator for the various busses on the DXP-XMAP: PCI Local Bus, Memory Bus, the DSP Host Port and the DSP Data Bus.

During initialization the System FPGA routes the DSP program code from the Local Bus to the DSP Host Port. The System FPGA supports diagnostic tasks such as reading oscilloscope trace data and DSP parameters via the DSP Data Bus.

During a run, the System FPGA post-processes events and negotiates host read operations from, and DSP write operations to, the SRAM memory. The event post-processor first compares DSP-scaled energies to minimum (underflow) and maximum (overflow) user-defined values. If the measured energy is within the allowed range, the post-processor performs a read-increment-write operation to the appropriate address in memory, i.e. it increments the corresponding spectrum bin.

The DXP-XMAP currently supports basic spectrum acquisition as well as time-resolved multi-spectrum acquisition. The two data acquisition modes use different memory architectures and thus require different firmware code to be downloaded.

In MCA mode on-board memory is configured as a single 1MEG by 32-bit device simultaneously accessible to both the host and the on-board DSP. Each spectral bin is thus a 32-bit value, allowing for up to 4,294,967,295 events per bin per run. The memory is normally cleared at the beginning of a run, but can be preserved, allowing for 'pause and resume' functionality.

In 16-bit mode a data acquisition run produces multiple spectra for each DXP processing channel. Spectrum memory is configured as two 1MEG by 16-bit devices, memory A and memory B, each accessible to *either* the host *or* the on-board DSP. Each spectral bin is thus a 16-bit value, allowing for up to 65,535 events per bin. Continuous operation is achieved by reading memory A while writing memory B, and vice-versa.

7.7.1 Basic 32-bit MCA Data Acquisition

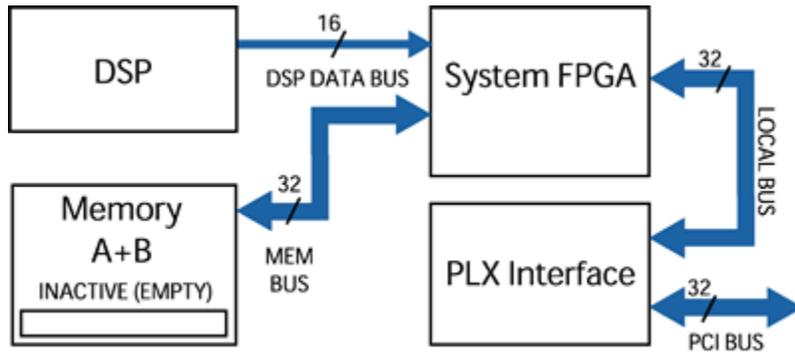


Figure 7.13: Data flow diagram for single spectrum mode.

A basic 32-bit MCA data acquisition run produces a single MCA energy spectrum, ranging from 256 bins to 16384 bins, for each DXP processing channel.

The external logic (LEMO) input can be configured to halt data acquisition, i.e. implemented as a GATE function. Data acquisition runs can be started and stopped manually, or can be stopped automatically according to a preset real time, live time or number of input or output events.

7.7.2 Full Spectrum 16-bit MCA Mapping/Scanning Mode

This mode produces multiple spectra, where an external logic signal typically is used. In the mapping mode an x-ray beam is scanned across a sample, and each spectrum corresponds to a scan point, or pixel. The external logic (LEMO) input can be configured to control the pixel advance function, which creates a new spectrum corresponding to a new pixel. Data acquisition runs can be started and stopped manually, or can be stopped automatically according to a preset number of pixels.

Pseudo-normal spectrum mode operation is supported for diagnostic purposes, with the following limitations: Data cannot be read out during a run; The 16-bit bins can easily overflow, depending on the run length and input count rate. Please refer to 5.3.3.3 for further information.

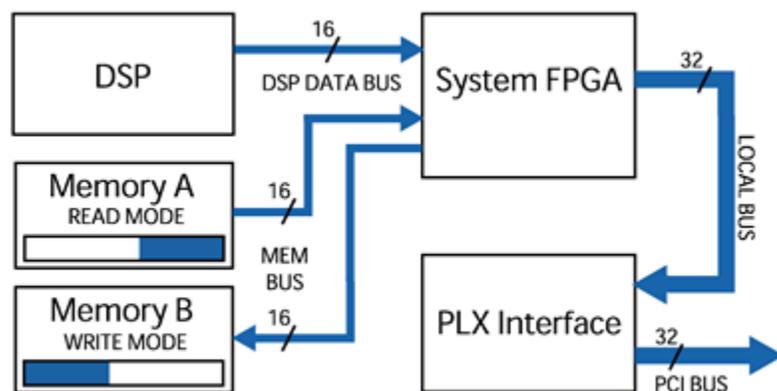


Figure 7.14: Data flow diagram for multiple spectrum mode.

7.7.3 Multiple SCA Mapping Mode

The XMAP hardware supports mapping mode with multiple user-defined SCAs. Up to 64 SCAs can be defined for each channel. A 16K word array is used to hold the mapping between MCA bins and SCA regions; the user can either select to have all channels use the same SCA definitions for all channels (which supports the full maximum MCA length of 16K channels), or use separate definitions of the SCA regions for each channel (where the maximum supported MCA length is 4K channels). This format specification does allow differences in the number of SCA's between channels. Two words (32 bits total) are used to store the total number of events in each SCA region; the low word is stored first in memory. Please refer to 5.3.3.4 for further information.

7.7.4 List Mode Mapping

The XMAP hardware also supports mapping in list mode. The events are stored in the order they are processed, and so the data are not separated according to channel. The channel number information is embedded into the upper two bits of the data word itself; the lower 14 bits are used to store energy information (typically MCA channel number). Please refer to 5.3.3.5 for further information.

7.8 3rd Party PCI Interface

The DXP-XMAP uses a PCI bus master interface controller chip manufactured by PLX technology. The PCI9054 ASIC negotiates transactions between the PCI bus and the Local Bus, supports burst transfers up to 132 Mbytes/second.

Appendices

Appendix A. DPP-XMAP Revision C Circuit Board Description

This Appendix to the DXP-XMAP User's Manual describes jumper settings, connector locations and power consumption calculations for the DXP-XMAP Revision C digital x-ray processor circuit board.

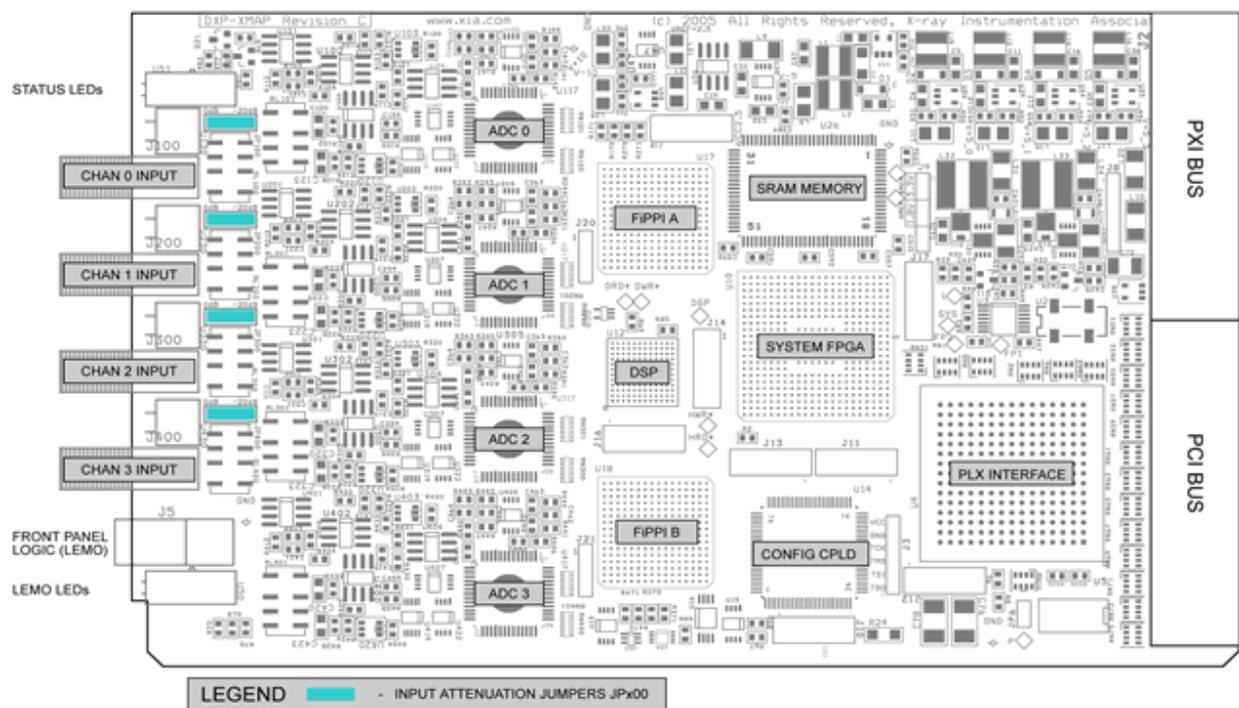


Figure A.1: Jumper, device and connector locations for the DXP-XMAP printed circuit card.

A.1. Jumper Settings

Reference	Name	Position Labels	Description
JPx100	Input attenuation jumpers	0dB	Default 0dB attenuation setting: 10.0K Ω input impedance; +/-6V input range
		-20dB	-20dB attenuation setting: 1.0K Ω input impedance; +/-24V input range.

A.2. LED Indicators

Reference	Name	Front Panel Labels	Description
Ux30	Rate Indicators	-	These bi-color LEDs are rate indicators for each channel. Each flash represents a preamplifier reset. - Green indicates a low rate (<50% dead time) - Yellow (green+red) indicates a moderate rate (>50% dead time, but less than maximum throughput) - Red indicates a high rate (at or above the point of maximum throughput).
U51*	Status Indicators	RUN	Illuminated green when a run is in progress.
		I/O	Illuminated yellow when a PCI transfer to or from the module is in progress.
		ERR	Illuminated when the module is in an error state.
U50*	Front Panel Logic (LEMO) Mode Indicators	G	Illuminated green when the modules is designated as a GATE Master.
		S	Illuminated green when the modules is designated as a SYNC Master.
		L	Illuminated green when the modules is designated as an LBUS Master.

* These LEDs flash in a test pattern when firmware is being downloaded to the board.

A.3. Connectors

Jx00 - Signal Inputs (4): SMA, connects preamplifier output to the XMAP.
Pasternack P/N: PE4545

J5 - Logic Input/Output: LEMO, configurable as GATE or SYNC input, or LBUS input/output
LEMO P/N: EPL.00.250.NTN

J1 -Standard PCI Port: High speed parallel communications port; standard pinout.
Erni P/N: 064-176

J2 -Standard PXI Port: High speed parallel communications port; standard pinout.
DDK P/N: HTC-SEB110-R11A

A.4. Power Consumption:

DXP-XMAP Power Budget							
Power Source			Load				
Name	Voltage	Source Current	Output Voltage	Per Channel	Total	Efficiency	Comment
	[V]	[mA]	[V]	[mA]	[mA]		
V+5_PXI	5	776	3.3		1000	85.00%	VCC: main digital voltage supply
		212	1.8		500	85.00%	VCC1.8: FPGA core voltage supply
		126	-5	30	120	95.00%	V-5: analog supply
		1115					
V+3.3_PXI	3.3	303	2.5		300	75.00%	VCC2.5: DSP core supply
		2852	5	400	1600	85.00%	V+5: analog supply
		3155					
V+12_PXI	12	100	10	25	100	83.00%	V+10: analog supply
V-12_PXI	-12	100	-10	25	100	83.00%	V-10: analog supply