ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

**Artisan Technology Group**
(217) 352-9330 | sales@artisantg.com | artisantg.com

**Full-service, independent repair center**
with experienced engineers and technicians on staff.

**We buy your excess, underutilized, and idle equipment**
along with credit for buybacks and trade-ins.

**Custom engineering**
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

**Expert team | Trust guarantee | 100% satisfaction**

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Murata / Datel DVME-630A2* at our website: ***Click HERE***

# LEGAL NOTICE

## LICENSE AGREEMENT

This notice constitutes a license AGREEMENT between DATEL, Inc. and the purchaser of DATEL data acquisition software.

### Definitions

For the purpose of this legal notice, a "FILE" shall mean a contiguous collection of machine-readable symbols, bytes, characters or codes which may be used by the Central Processing Unit (CPU) in the ordinary operation of DATEL products on the user's computer or processing equipment. FILES generally are supplied on magnetically recorded computer disks or tapes, printed listings or by digital communication link including telephone modem communication. A "PROGRAM" is a FILE which may be loaded and processed on the user's computer to perform a single function or related group of functions. "SOFTWARE" shall mean one or more FILES or PROGRAMS supplied on magnetically recorded disks or tapes to perform a group of related functions for DATEL board products. Human readable "SOURCE" files may be displayed as text format using alphabetic letters, numbers and punctuation such that an experienced programmer will fully recognize and understand the nature and design of the PROGRAM contained in the SOURCE FILE. SOURCE FILES generally must be further processed on the user's computer before being executed. "EXECUTABLE", "RUNNABLE" or "OBJECT" files are only machine-readable and are intended for direct execution by the user's computer without further initial processing.

### Usage

The files included on DATEL computer disks or tapes are supplied to DATEL customers solely for the purpose of illustrating how DATEL computer-controlled products operate or to allow modifications of this software by the customer for a particular application. In the case of binary executable-only files, the user may not disassemble, reverse compile, reverse engineer or otherwise remove the proprietary status of these programs. By using these software products, you agree to and will comply to the terms of this AGREEMENT.

The programs and files on DATEL-supplied disks or tapes are proprietary to DATEL. All supplied programs and files are the property of and copyrighted by DATEL, Inc. The programs and files may be reproduced, transferred, copied, printed out, displayed, modified or used in the customer's application as long as they control DATEL board products and as long as source format files are not further distributed to third parties beyond DATEL's customer without written permission.

If the user modifies these programs for a specific purpose, such modification, which retains a substantial portion of the original unmodified software, must include this proprietary notice in the source program files. The files supplied on DATEL computer disks or tapes may not be disclosed, copied, reproduced or transferred to third parties beyond DATEL's customer without written permission. DATEL customers must control the distribution of supplied computer disks or tapes and keep them in safekeeping to prevent unauthorized distribution to third parties. Customer-modified binary executable files which are based on these source files may be distributed to third parties without restriction.

### Disclaimer

Although the programs on DATEL disks or tapes have been carefully checked and have executed properly, errors are possible. Program files are offered "as is" and DATEL cannot assume any responsibility for damages in the event of modification of these files or damages arising from loading and running software which has been modified for a particular purpose. DATEL cannot be liable for loss of revenues or loss of data using this computer information. The maximum extent of DATEL's liability shall be the cost of repair or replacement of this software. The user is solely responsible for determining if the DATEL files and programs are suitable for his or her application. DATEL recommends thorough testing before committing any modified program to an application. DATEL retains the right to modify this software at any time in accordance with its policy of product support and improvement.

### Upgrade Policy

From time to time, DATEL may add features or enhancements to existing software products. These additional features will be described in DATEL's product literature and in product support bulletins distributed to the field. A registered user may obtain an upgrade at any time (either in or out of the warranty period) for source disk upgrade or for executable disk upgrade at the current software upgrade prices. DATEL shall not be obligated to provide upgrades at no charge.

To receive the upgrade, a purchase order must be placed then the Subsystems Application Engineering Department must be contacted for registration verification and to receive the proper disk.

IMPORTANT: You must be a registered user to be eligible for the upgrade. The user must have previously returned the warranty card so that DATEL will have the current name and address of the user. This policy may be modified at any time.

### Customer Product Support Policy

DATEL will answer telephone and written questions of reasonable duration about this software during DATEL's normal business hours. Although DATEL may offer programming advice, in general we will not design the user's application or write specific application software. More extensive support may be available from DATEL under special agreement. DATEL recommends local software consultation by third party contractors at the customer's site for larger projects. DATEL will review known repeatable errors (if any) found in this software in a timely fashion. Users should clearly and exactly describe the conditions of such errors in writing. DATEL will replace defective disks which are not damaged in shipping or by customer misuse.

This AGREEMENT is dated August 12, 1991

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

## TABLE OF CONTENTS


Unpacking, Warranty, Disclaimer, Proprietary Notice

**Section 1 - General Information**

1.0 Scope of this Manual 1-1
1.1 Suggested Equipment 1-1
1.2 Implementation Procedure 1-1
1.3 General Description 1-2
1.4 Functional Specifications 1-4

**Section 2 - Board Configuration and Installation**

2.0 Determining Your Host Environment 2-1
2.1 Bus Slot Selection 2-1
2.2 Host Memory Map Compatibility 2-1
2.3 Host System Timing 2-2
2.4 Host Interrupt Compatibility 2-3
2.5 Jumpers and Switches 2-4
2.6 Memory Base Address Selection 2-6
2.7 Address Modifier Selection 2-6
2.8 Interrupt Level Selection 2-7
2.9 16- or 32-bit Data Bus Selection 2-8
2.10 82C54 Timer Clock Source Jumpers 2-9
2.11 FIFO Flag Jumpers 2-9
2.12 A/D Channel Advance Jumper 2-10
2.13 DSP Start Mode Jumpers 2-10
2.14 DSP Reset Jumper 2-10
2.15 EPROM Jumpers 2-10
2.16 Board Installation 2-11

**Section 3 - Host Architecture and I/O Connections**

3.0 Memory Mapping - Logical and Physical Address 3-1
3.1 DVME-630 VME bus Memory 3-2
3.1.0 VME bus Command Register: (Write only to MEMBASE + 7FFFC, Dh) 3-2
3.2 Initial Software Tests 3-2
3.2.0 Your Debugger 3-3
3.2.1 Access Size 3-3
3.3 I/O Connections 3-4
3.3.0 RS-232-C Serial Port 3-4
3.3.1 Memory Bus Expansion 3-4
3.3.2 Analog Input Connector, P3 3-6
3.3.3 DSP Serial Port, External Clock and Trigger, P5 3-6

**Section 4 - Programming Information**

4.0 Hardware Organization Overview 4-1
4.1 VME bus Command Register (Write MEMBASE + 7FFFDh) 4-1
4.2 Local DSP Memory Map 4-2
4.3 DVME-630 On-board I/O Registers 4-4
4.4 82C54 Programmable Timer/Counter 4-4
4.5 DSP Command Register (Write only to 804004h) 4-6
4.6 Start Channel Address Register (Write only to 804005h) 4-7
4.7 FIFO Data Register (Read only at 804007h) 4-7
4.8 Miscellaneous DSP Register Bits 4-8


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

**Section 5 - AFB Programming**


5.0 DVME-630 General Control Methods 5-1
5.1 Software Introduction 5-1
5.2 Executive Package 5-2
5.3 Software Hierarchy 5-2
5.4 Simultaneous Concurrent Operation and DPR Access 5-2
5.5 Power Up Sequence 5-3
5.6 Dual Port Random Access Memory Mapping 5-3
5.7 Book-keeping Area 5-4
5.8 Local Control and Status Flags 5-5
5.9 Local Pointers and Buffer Addresses 5-8
5.10 User Extended Library 5-8
5.11 Application Function Block Structure 5-9
5.12 AFB Markers 5-10
5.13 Double Buffering 5-11
5.14 System Speed Hierarchy 5-11
5.15 System Stack 5-13
5.16 Buffer Size Considerations 5-13
5.17 Cycle Times 5-13
5.18 Highest A/D Speeds versus Memory 5-14


**Section 6 - Application Function Block Library**


6.0 A/D Scan Functions 6-1
6.0.0 INITAD 6-1
6.0.1 INITTIM0 6-1
6.0.2 INITTIM1 6-2
6.0.3 INITTIM2 6-2
6.0.4 SADTSC 6-3
6.0.5 SADCR 6-3
6.0.6 SSTR 6-3
6.0.7 SADSPC 6-4
6.0.8 STADS 6-4
6.0.9 SFIFOI 6-5
6.0.10 SADR 6-5
6.0.11 RFIFO 6-5
6.0.12 CALAD 6-5
6.0.13 SCOMMREG 6-6
6.0.14 DMA_INTRPT 6-6
6.1 Data Formats 6-7
6.2 FFT Array Formats 6-7
6.3 DSP AFB Functions 6-9
6.3.0 CFFTC (D) 6-10
6.3.1 FFT (D) 6-10
6.3.2 BITREV (D) 6-10
6.3.3 DCT (D) 6-11
6.4 Filter Routines 6-11
6.4.0 PAUSE 6-11
6.4.1 CIRFIR (D) 6-12
6.4.2 LINFIR (D) 6-12
6.4.3 IIR (D) 6-13
6.4.4 FIR(D) 6-13
6.5 FFT Windowing 6-16
6.5.0 WINDHAM (D) 6-17


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

6.5.1 WINDHAN (D) 6-17
6.5.2 WINDREC (D) 6-18
6.5.3 WINDBLH (D) 6-18
6.5.4 WINDRCO (D) 6-19
6.6 Array Conversion Routines 6-19
6.6.0 IEEDSP 6-19
6.6.1 DSPIEEE 6-19
6.6.2 MATADD (D) 6-20
6.6.3 MATMUL (D) 6-20
6.6.4 HSTGRM (D) 6-20
6.6.5 INT2FLOAT (D) 6-21
6.6.6 TWIDDLE_R (D) 6-21
6.6.7 TWIDDLE_C (D) 6-21
6.6.8 MAGFFT (D) 6-21
6.6.9 CALL_SINE (D) 6-22
6.6.10 CALL_COS (D) 6-22
6.6.11 CALL_CONST (D) 6-22
6.6.12 FLOAT2INT (D) 6-22
6.6.13 DBFFT (D) 6-23
6.7 Buffer Definition and Data Transfer Functions 6-23
6.7.0 DEFSBUF 6-23
6.7.1 DEFDBUF 6-24
6.7.2 DEFNBUF (D) 6-24
6.7.3 UNRAV4 6-25
6.7.4 UNRAV2 6-25
6.7.5 UNRAVX (D) 6-25
6.7.6 FLOATXFER 6-26
6.7.7 DPRXFER 6-26
6.7.8 ADDXFER 6-27
6.7.9 INT2PC 6-27
6.7.10 SET_IBUF (D) 6-27
6.7.11 IBUF_READY (D) 6-28
6.7.12 IBUF_RELEASE (D) 6-28
6.7.13 SET_OBUF (D) 6-29
6.7.14 SWITCH_BUFFERS (D) 6-29
6.7.15 SIGN_EXTEND (D) 6-29
6.7.16 CONCAT (D) 6-30
6.7.17 CIRBUF (D) 6-30
6.7.18 NCIRBUF (D) 6-30
6.8 MEM-30 High Speed Expansion Memory 6-34
6.8.0 EXPENABLE 6-35
6.8.1 EXPXFER (D) 6-35

**Section 7 - The DVME-630 Executive Software**
7.0 Executive Description 7-1
7.1 System Requirements 7-1
7.2 Your Text Editor 7-2
7.3 Disks Supplied 7-2
7.4 Transferring the Software 7-3
7.5 Source File Format 7-3
7.6 Compiling the Programs 7-3
7.7 Running D630SET 7-5
7.8 Main Screen 7-6

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

7.9 Configuration Menu 7-6
7.10 Running the AFB 7-7
7.11 D630SET Return Codes 7-8
7.12 Configuration File Format 7-8
7.13 DVME-630EXEC Files 7-8
7.14 PAUSE Processing 7-9
7.15 Adding a New AFB Function 7-9

**Section 8 - Programming the Timer/Counters**
8.0 Introduction 8-1
8.1 Timer/Counters 8-2
8.2 Channel Addressing 8-2
8.3 A/D Start Clock 8-3
8.4 Trigger 8-3
8.5 Programmable Interval Timer 8-4
8.5.1 A/D Sample Counter, 82C54 Counter 0 [AFB INITTIM0] 8-5
8.5.2 A/D Start Clock, 82C54 Timer 1 [AFB INITTIM1] 8-5
8.5.3 Trigger Rate DSP Timer TCLK1 [AFB SSTR] 8-6
8.6 Simultaneous Sample/Hold Mode (DVME-630A only) [82C54 Counter 2] 8-6
8.7 A/D Timing Diagram 8-6
8.8 Sample Counter Sizing 8-8
8.9 Using an External Trigger Clock 8-8
8.9.1 Direct External Clocking 8-9
8.9.2 Multichannel External Clocking 8-9

**Section 9 - AFB Programming Examples**
9.0 Introduction 9-1
9.1 Debugging 9-1
9.1.0 AFB Design Strategy 9-2
9.1.1 Loops 9-2
9.2 AFB Structure Rules 9-5
9.2.0 AFB Source File Format 9-6
9.3 AFB Examples 9-6
9.3.0 Example 1 - Fill a Buffer with a Constant 9-6
9.3.1 Example 2 - Fill a Large Buffer with A/D Data 9-7
9.3.2 Example 3 - Swapped Buffer A/D Fill 9-8
9.3.3 Example 4- Looping Swapped Buffer Operation 9-10
9.3.4 Example 5 - FFT Collection 9-11

**Section 10 - Miscellaneous Applications**
10.1 Hard Disk Output 10-1
10.2 Disk Speedups 10-1
10.3 Communications Network or Data Link Output 10-2
10.3.0 Noise Reduction Suggestions 10-3
10.4 Programmable 82C54 Counter Timer Information 10-3
10.4.0 Block Diagram 10-3
10.4.1 Control Word Register 10-5
10.4.2 Read Back Command 10-6
10.4.3 Read Back Status 10-6
10.4.4 Counter Register 10-6
10.5 82C54 Modes of Operation 10-7
10.5.0 Mode 2 - Rate Generator 10-7

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

10.5.1 Mode 4 - Software Triggered Strobe 10-8
10.5.2 Mode 5 - Hardware Triggered Strobe 10-8
10.6 A/D FIFO Half Full Interrupt Processing 10-9

**Section 11 - Analog Module Configuration and Calibration**
11.0 Module Identification 11-1
11.1 Analog Module Configuration 11-1
11.2 Input Voltage Range Selection 11-2
11.3 A/D Output Binary Coding 11-2
11.4 DVME-630F Channel Selection 11-2
11.5 Input Amplifier Programmable Gain Selection 11-7
11.6 Calibration 11-7
11.6.0 Equipment Required 11-7
11.6.1 Procedure 11-8
11.6.2 Zero/Offset Adjust 11-8
11.6.3 A/D Gain Adjustment 11-9
11.6.4 Universal Coding Chart 11-10
11.7 Sampling Rate Theory 11-11
11.8 Averaging Noisy Data 11-11
11.9 Unused Input Channels 11-12
11.10 High Speed Multichannel Scanning 11-12
11.11 Repair and Troubleshooting 11-13
11.12 DVME-630D Input Resistor 11-14
11.13 Dual A/D Model DVME-630F 11-14

**Appendix A**
DVME-630 Data Sheet
Engineering Drawings

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

# CAUTION

*Do not apply power to this board until you have read this manual*

## Unpacking

This product should include a DVME board in a static-protective bag, optional software disks and this User Manual. Upon receipt, carefully remove the board from the anti-static bag and visually inspect the board for physical damage which may have occurred in shipment. In the event of damage, save all packing materials for inspection by your carrier to validate shipping damage claims. Should the board need repair at a later date, it may be safely returned in its original packing materials. If you do not intend to use the board immediately, store it in the anti-static bag until use.

*CAUTION - Always store the board in the anti-static bag*

This board contains components that can be damaged by static discharge and should be handled with caution. Make sure your body is earth grounded before handling the board, especially in dry, low humidity conditions. Avoid board contact with clothing having a high synthetic content. Avoid contact after walking across carpeting. Before you first touch the board, continuously hold the VME chassis it will be installed into. In severe static discharge environments, consider grounding wrist straps or other protective measures.

## Warranty

This board was precision manufactured with the highest quality methods. DATEL warrants that this product was manufactured free of defects in workmanship or materials under normal use and service as described in the literature for this product. Obligations under this warranty are limited to replacing or repairing, at DATEL's option, any of said products, at DATEL's factory or facility, transportation charges prepaid, and which are, after examination disclosed to the satisfaction of DATEL to be thus defective, for a period within one year of shipment. This warranty shall not apply to any such products which have been repaired or altered except by DATEL or which have been subject to misuse, negligence or accident. In no case shall DATEL's obligation exceed the original purchase price. The aforementioned provisions do not extend the orignal warranty period of any product which has either been repaired or replaced by DATEL. This warranty does not contain a guarantee, either expressed or implied, of merchantability or fitness for a particular purpose. DATEL cannot assume liability for damage or loss as a result of the use of this product. It is the customer's sole responsibility to determine if this product is suitable for his or her application.

## Disclaimer

All information contained in this document has been carefully examined and is believed to be entirely accurate. However, DATEL assumes no responsibilities for errors or omissions. DATEL reserves the right to make changes to this manual without prior notification in accordance with DATEL's policy of product support and improvement.

## Proprietary Notice

The information and the design of this product are the exclusive property of DATEL. This entire document is proprietary to DATEL and shall not be disclosed to third parties without the express written consent of DATEL. This information is not to be photocopied, transcribed, recorded, communicated or reproduced by any means without permission. The foregoing does not apply to vendor proprietary rights or rights under the patents of third parties. Use of this product does not convey a license, either expressed or implied.

Copyright (c) 1992 by DATEL, Inc.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

# SECTION 1

# GENERAL INFORMATION

## 1.0 Scope of this Manual

Thank you for your interest in DATEL's DVME-630 A/D DSP board. This manual contains complete information to configure, install, program and use the DVME-630. Preliminary troubleshooting may be accomplished with this manual but extensive component level repairs should be referred to DATEL. Users with critical downtime requirements should consider stocking whole spare boards.

DVME-630 users should be familiar with all aspects of developing a high performance data acquisition application. This includes selecting, connecting and installing sensors and transducers, complete operation of the host computer and software programming. Arithmetic procedures for handling A/D data, possible error sources (both hardware and software) and system timing should be clearly understood. It is assumed that the user is knowledgeable in the theoretical background of Digital Signal Processing mathematics. Some knowledge of digital logic, microprocessors, and precision linear circuits will prove helpful.

Although there are useful example software programs available for the DVME-630, they cannot possibly cover every conceivable application. The user should be a competent programmer of hardware peripherals.

## 1.1 Suggested Equipment

Besides the DVME-630, you will find the following equipment helpful in developing your application:

- A voltage reference source such as DATEL's DVC-350 handheld calibrator will be valuable for testing your software. The DATEL DVC-8500 Voltage Reference may be used for periodic DC recalibration.

- A VME bus extender board will be needed for access to potentiometers during recalibration and for troubleshooting.

- A selection of various probes, test leads, cables and electronic tools may be helpful when testing signal sources. These should only be used by skilled persons.

- For DSP FFT testing, a high purity sine wave generator and low pass filters will help you qualify the system.

- A digital pulse generator may be used to test external trigger inputs. An oscilloscope will verify A/D triggering or the Pacer Timer output.

## 1.2 Implementation Procedure

CAUTION: Read this manual before you install the board and apply power.

The general outline for application development is shown and will be expanded throughout this manual.

1. Study your host computer and its documentation to decide how the DVME-630 will be integrated. You must determine the correct board slot, memory addressing, interrupt usage, software interface, etc.

2. Configure the on-board hardware (jumpers for the A/D converter, timers, interrupt, memory addressing, etc.). Most items will not require changing however the memory base address usually MUST be changed.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

3. If you are using DATEL's software, do a one-time transfer from the MS-DOS disks to your VME disk. This may involve an MS-DOS disk reader utility on your VME system, an electronic bulletin board mail system, local area network or a transfer via the serial port from an MS-DOS computer. For the DVME-630EXEC package, there is at least one binary file. Make sure the transfer is error-free. Make sure you understand how your host system handles binary files.

CAUTION: Be sure to save your entire host system on backup disks or tapes before installing the DVME-630.

4. Edit the host side source files for the proper memory base address to be used by the DVME-630. Also insert your own utilities (if needed) to allow absolute physical access to the DVME-630. Compile the host side files and fix the compilation or link errors (if any). The included MAKE files should simplify compilation if you use a compatible version of MAKE.

5. Install the physical I/O device driver and rebuild your system if that is required for physical I/O. If possible and unless you are experienced, do a separate independent test of the physical I/O access function in a safe area of RAM before rebuilding your full system. Make sure you can access any area of 24-bit address space.

6. Install the board, apply power and confirm that your host computer operates normally. Do preliminary debugger testing to be sure you can access the Dual Port RAM. Run the EXEC software. Try some of the simple AFB examples in the manual. Connect analog input signals and confirm proper data.

7. Develop and debug your final software. Document the application so that it can be maintained and for application questions.

### 1.3 General Description

Advanced performance from the DVME-630's on-board Digital Signal Processor (DSP) offers a broad range of high speed waveform analysis and recording applications. The DVME-630 will acquire up to sixteen analog input channels, digitize them and store them in local memory while DSP math processing and data transfer is done concurrently. The system is intended for preprocessing "seamless" A/D data streams to the host system.

The DVME-630 is ideal for non-stop, continuous Fast Fourier Transform (FFT) processing, communications receiver signal collection to disk or simultaneous graphics display of spectral data. Application areas include signal recovery from noisy channels, harmonic distortion analyzers and vibration/resonance filtering systems. For use with ultrasonic, sonar or acoustic signals, the interrupt-driven, simultaneous block transfers of data insure no information loss. Other uses include high speed mapping and imaging, satellite channels, astrophysics, seismology, biomedical signals, array processing, control systems, simulators, engine analyzers, aerodynamics and vehicle systems.

The board consists of a pluggable analog input subsection, timer-counters, DSP central processing unit (CPU), dual port RAM, local RAM, VME bus interface, registers and DC power supply. Input signals pass through a very high speed channel multiplexer (except models DVME-630D and -630F) to a sampling analog-to-digital (A/D) converter. On model DVME-630A, all four channels are acquired simultaneously by a quad simultaneous sample/hold (SSH) section. A choice of channels, input ranges, speeds and resolution is offered in the analog section.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

A/D triggering for spectral and FFT applications must be precisely controlled. This is handled by a programmable timer-counter section which can control the interval between A/D conversions and the interval between multi-sample A/D scans. The number of samples may also be counted for repeating array sampling. The timer-counter may use an on-board crystal oscillator or an external timebase for precision phase-tracking.

The digital output of the A/D passes directly to a first-in, first-out (FIFO) memory. The FIFO acts to decouple the precision timing of the A/D section with the block transfers governed by the DSP internal direct memory access (DMA) controller. Additional timers internal to the DSP are also used.

A/D FIFO data may be sent to dual port random access memory (DPR) shared with the host VME bus. The DPR is organized as 128K by 32 bits of fast static RAM. The user may also set up the DPR as 256K by 16 bits, allowing all access using only the P1 VME bus connector if needed. Local A/D-FIFO block transfers are controlled by the DMA controller or program transfers in the DSP. The DMA may run in background while math processing continues. Local FIFO and DMA TC interrupts to the DSP arbitrate these activities. Typically, a swapped dual buffer method is used so that samples are not lost during other processing. Local hardware registers control all A/D, FIFO and trigger activity.

Single cycle fetch and execution, zero-overhead of looping instructions, software variable wait-states, simultaneous parallel instructions, block repeat and an internal instruction cache memory are some of the advanced high speed features of the Texas Instruments 320C30 DSP. The DSP uses 32-bit local data paths for very high speed. Simultaneous access attempts to the DPR by both the VME bus host and the DSP are resolved by high speed arbitration logic. The DSP also has a separate 8K by 32-bit local expansion memory for the stack or temporary data. The architecture of the DSP allows simultaneous processing of two tables from two sections of memory. This provides optimum processing of FFT's and other array functions.

The DVME-630 DPR appears as both read/write memory addresses and as two hardware control registers mapped into the top of the DPR. The registers include a command register and a software programmable interrupt vector ID register. The DVME-630 will operate with all 29000, 680X0, 880X0, 80X86, RISC and SPARC CPU's. The DPR base address may be located anywhere up to 16 megabytes on 512 kilobyte boundaries. The board will also operate with all popular operating systems including SUN, UNIX, OS-9, PDOS, pSOS, MS-DOS, VRTX and others.

Executive Software

Access to the on-board DSP library and all A/D-timer functions is provided in the Executive software package. After loading in the Executive from host disk to the DPR, the DSP is transitioned from reset to run using a command register control bit. The board may be reset at any time using this technique. The comprehensive Executive software package offers fast A/D sample collection and DSP math without writing any local programs. A simple, powerful, high speed command list is used to access the local DSP library. On the VME bus side, the Exec driver controls very fast buffer transfers to disk or memory using VME bus bus interrupts generated from the DVME-630. Either host DMA transfers or CPU program transfers may be used. Host interrupt levels are also jumper programmable. Software is available in professionally-written, heavily-commented "C" source format on popular disk media. The software is highly portable to most VME platforms.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

## 1.4 Functional Specifications

Specifications are typical at +25 °C, dynamic conditions, gain = 1, unless otherwise noted.

Analog Inputs

| | |
|---|---|
| Number of Channels | 4 channels (models DVME-630A,B,C)<br>1 channel (model DVME-630D)<br>16S/8D channels (DVME-630E)<br>2 channels (DVME-630F) |
| Input Configuration | Single-ended, non-isolated (models A, B, C, F). The DVME-630E accepts single-ended or differential inputs. The DVME-630D is differential. |

Full Scale Input Ranges (user-selectable)

| | 0 to +10V<br>±10V | ±5V | ±1V |
|---|---|---|---|
| DVME-630A | yes | no | 2 chans.<br>gain = 10 |
| DVME-630B | yes | yes | no |
| DVME-630C | yes | yes | no |
| DVME-630D | no | yes | no |
| DVME-630E | yes | yes | Note 8 |
| DVME-630F | 0 to +10V | yes | no |

| | |
|---|---|
| Programmable gain | Models A and E, see Note 8 |
| Input Impedance (630A,B,C,E,F) Note 5 | 10 Megohms, min. power on<br>1.5 Kilohms power off<br>2 Kilohoms (630D), 1 Kilohms (630F) |
| Input Bias Current | ±1 nA |
| Input Capacitance | 10 pF per channel |
| Input Overvoltage | ±15V (no damage) |
| Overvoltage Recovery Time | 2 microseconds max. |
| Common Mode Voltage Range (DVME-630E) | ±10V maximum referred to analog common. |
| Common Mode Rejection (DC to 60 Hz) | -80 dB (g = 100) [630E only] |
| Addressing Modes | 1. Single channel<br>2. Simultaneous Sample/Hold (DVME-630A only)<br>3. Sequential with autosequenced addressing<br>4. Random addressing by software<br>5. Dual-channel simultaneous (DVME-630F) |


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Sample/Hold

| | |
|---|---|
| Acquisition Time (FSR step) to 0.01% of FSR | 900 nS max. (DVME-630A)<br>750 nS max. (DVME-630B,E)<br>200 nS max. (DVME-630C)<br>50 nS max. (DVME-630D) 0.1%<br>165 nS max. (DVME-630F) |
| Aperture Delay | 15 nS (DVME-630A)<br>20 nS (DVME-630B,C,E,F)<br>10 nS (DVME-630D) |
| Aperture Delay Uncertainty | 50 ps max. (DVME-630A)<br>±100 ps max (DVME-630B)<br>±40 ps max (DVME-630C,E,F)<br>±10 ps max (DVME-630D) |
| Droop Rate | 1 μV/μS |
| SSH Chanel-to-Channel Linearity Tracking (DVME-630A only) | ±0.03% |

A/D Converter

| | |
|---|---|
| Resolution | 12 bits (DVME-630A,C,D,E,F)<br>14 bits (DVME-630B) |
| Conversion Period | 500 nanoseconds (DVME-630A)<br>1 microsecond (DVME-630B,C,E)<br>200 nanoseconds (DVME-630D)<br>400 nanoseconds (DVME-630F) |
| Output Coding | Positive-true right-justified straight binary (unipolar) or right-justified two's complement (bipolar) with sign extension through bit 15. |
| A/D Trigger Sources | 1. Local Pacer sample clock (software selectable)<br>2. External TTL sample clock |

Total System DC Characteristics
Gain = 1 (Note 6)

| | |
|---|---|
| Integral Non-linearity at +25 °C | ±1 LSB of FSR (DVME-630A,C,D,E,F)<br>±1.5 LSB of FSR (DVME-630B) |
| Differential Non-linearity at +25 °C | ±0.75 LSB of FSR max.(DVME-630A,C,E)<br>±1 LSB of FSR (DVME-630B,D,F) |
| Full Scale Temperature Coefficient | ±0.1 LSB per °C max (DVME-630A,C,D,E,F)<br>±0.3 LSB per °C max (DVME-630B) |


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

| | |
|---|---|
| Zero or Offset Temperature Coefficient | ±0.1 LSB per °C max (DVME-630A,C,E,F)<br>±0.3 LSB per °C max (DVME-630B,D) |
| Power Supply Rejection | ±0.004% per % of bus +5V |
| A/D Memory | |
| Architecture | First-In, First-Out (FIFO) |
| Memory Capacity | 1024 or 4096 A/D samples. |
| Total System Dynamic Performance (Note 1) | |
| System Bandwidth (single channel, FSR input) | 1 MHz (DVME-630A,C,F)<br>200 KHz (DVME-630B,E)<br>2.5 MHz (DVME-630D) |
| Total Throughput to FIFO (single channel, gain = 1) | 700 nanoseconds (DVME-630A)<br>2 microseconds (DVME-630B,E)<br>1 microsecond (DVME-630C)<br>250 nS (DVME-630D)<br>500 nS (DVME-630F) |
| Throughput to FIFO per A/D Sample (sequential channels, gain = 1) see Note 3 | 1 microsecond (DVME-630A)<br>3 microseconds (DVME-630B)<br>2 microseconds (DVME-630C)<br>4 microseconds (DVME-630E)<br>500 nS (DVME-630F) |
| Total Harmonic Distortion FS input (Note 2) | -72 dB (DVME-630A,C,E)<br>-75 dB (DVME-630B)<br>-68 dB (DVME-630D)<br>-70 dB (DVME-630F) |
| Trigger Control | |
| Programmable Timer/Counter Type | 82C54 (requires additional wait states during access). |
| Functions | 1. EOC sample count<br>2. A/D start rate (16 bit divisor)<br>3. Scan trigger rate (16 bit divisor) |
| Pacer Sample Counter | 1 to 65,536 samples. Drives the Acquire flag/interrupt gate for A/D start pulses. Larger sample blocks may be programmed. |
| Timer Clock Source | 1. Internal 8 MHz crystal clock (user-selectable)<br>2. External TTL input, 10 MHz max. |


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Local Microcomputer

| | |
|---|---|
| CPU Type | Texas Instruments TMS 320C30 Digital Signal Processor |
| Local Data Bus | 32 bits |
| CPU Clock Speed | 32.000 MHz |
| Local DMA Controller | Internal to 320C30 CPU |
| Primary Memory (Dual ported to VME bus) | 128K x 32 static RAM |
| PROM Option | 32K x 32 of primary memory may be replaced by user-programmed Read Only Memory. Additional wait states may be needed during PROM access. |
| Expansion Memory (No VME bus access) | 8096 x 32 static RAM |
| Internal DSP Memory | Two 1024 x 32 |
| Dual Port Access | Hold mode by control bit or from VME bus dynamic hold per each access. |
| CPU Test Port | Supports TI XDS1000 Extended Development System. |
| Local Interrupts to DSP | Int 0-3 from VME host request, A/D FIFO or sample count acquire flags or optional external interrupt. |
| Wait States | May be software programmed via CPU register up to 7 waits. Defaults to 7 waits at reset. |

VME bus Interface

| | |
|---|---|
| Data Bus Width | 16 bits using P1 connector or 32 bits using P1 and P2. Changed by alternate transceiver sockets on board. |
| Address Bus | 24 address lines (A23-A1) plus 6 Address Modifiers. |
| Address Modifier Codes | 39 hex or 3D hex, selectable. |
| VME bus Interrupt | 1 line, selectable IRQ 1-7*. Asserts maskable programmable 8-bit vector ID code. |
| VME bus Interrupt Sources | via DSP I/O bit (software programmable). |
| Architecture | SAD24:SD16 or SAD24:SD32 slave consisting of 128K x 32-bit longwords of dual ported RAM. The DVME-630 is not a bus master. A small register area overlays the top of the DPR. DSP location 0 is mapped at the bottom of DPR. |


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

| | |
|---|---|
| Control/Status Functions | DSP halt request, DSP reset/run, VME interrupt enable/disable, vector ID R/W, force local maskable interrupt to DSP. |
| Standards Compliance | IEEE P1014/D1.0 |
| Data Transfer | Uses 16 MHz VME bus SYSCLK signal (required) to generate DTACK* with selectable delay in 125 nS steps. Will accept host DMA controller access. The PAL may be reprogrammed for 62.5 ns steps. |
| Connectors | |
| VME bus, P1 and P2 | Two 96-pin male DIN connectors. |
| Analog Input Connectors | DB-25S |
| Memory Expansion (32-bit data) | 72-pin header, unbuffered. Uses 0.025" square pins, 0.100" spacing. |
| XDS CPU Port | 12-pin header, 0.1" pinning. |
| Fast CPU Serial I/O (Incl. trigger, clock, etc.) | 20-pin header, unbuffered, 0.1" pinning. |
| RS-232-C Asynchronous Serial | 3-pin header (Xmt, Rcv, Gnd), Port [uses software UART] 0.1" pinning. |
| Miscellaneous | |
| LED Lamp | Front panel green LED lamp may be controlled by internal register bit for alarms, etc. |
| Analog Section Modularity | The MUX-S/H-A/D module is socketed for function interchange. |
| Analog Section Adjustments | Offset and gain per channel for SSH on DVME-630A,F. A single offset and gain pot is provided on DVME-630B,C,D,E. Recommended recalibration interval is 90 days in stable conditions. |
| Power Required | +5V dc ±5% at 4 Amps max. from VME bus. |
| Operating Temperature Range | 0 to +60 °C, forced cooling is recommended. |
| Storage Temperature Range | -20 to +85 °C |
| Relative Humidity | 10% to 90%, non-condensing. |
| Altitude | 0 to 10,000 feet. |
| Outline Dimensions | Double height VME, 6U outline.<br>9.19"W x 6.3"D x 0.6"H (233,5 x 160 x 15,24 mm). |
| Weight | 22 ounces |


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Notes:

1. Total throughput includes MUX settling time after changing the channel address, S/H acquisition time to rated specifications, A/D conversion and FIFO transfer. Total throughput is not delayed by host software whenever the FIFO is not full.

2. THD test conditions are:

    1. Input frequency 250 KHz (DVME-630A,F)
       100 KHz (DVME-630B,C,E)
       500 KHz (DVME-630D)

    2. Generator/filter THD is -90 dB min.

    3. THD computed by FFT to 5th harmonic.
       THD = 20 * log10 ((( V2^2 + V3^2 + V4^2 + V5^2 )^0.5)/ Vin)

    4. Inputs are 1/2 full scale. No channel advance.

    5. A/D trigger rate = 1.5 MHz (DVME-630A), 500 KHz (630B,C,E), 4 MHz (630D), 2 MHz (630F)

3. The rates shown for sequential sampling are the maximum A/D converter start rates and include required MUX sequencing and settling delays. For example, if four channels of the DVME-630C were scanned, the maximum sample rate on any one channel would be 2 microseconds X 4 channels = 8 microseconds (125 KHz per channel).

4. To avoid overload recovery delays, do not let the analog input exceed the input voltage range.

5. The input impedance of 10 Megohms minimum avoids attenuation errors from external input source resistance. For many applications, an in-line coaxial 50-ohm shunt, inserted adjacent to the front connectors, is recommended to reduce line reflections and standing wave errors.

6. Allow 20 minutes warmup time to rated specifications for model DVME-630B.

7. The DVME-630 must be mapped in the physical memory of the host VME computer. On some operating systems (notably UNIX), a physical-to-logical addressing function supplied with the OS is used to access the board. The OS may need to know about this absolute physical memory reservation at boot-up time using a device driver. See the example programs available with the DVME-630.

8. Programmable Gain

   Gains from x1 to x100 are available on the DVME-630E by installing a precision resistor. Additional settling delay will be needed at higher gains. Fixed gains of x1 and x10 on two channels are selectable on the DVME-630A. They offer 1 Volt input ranges.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

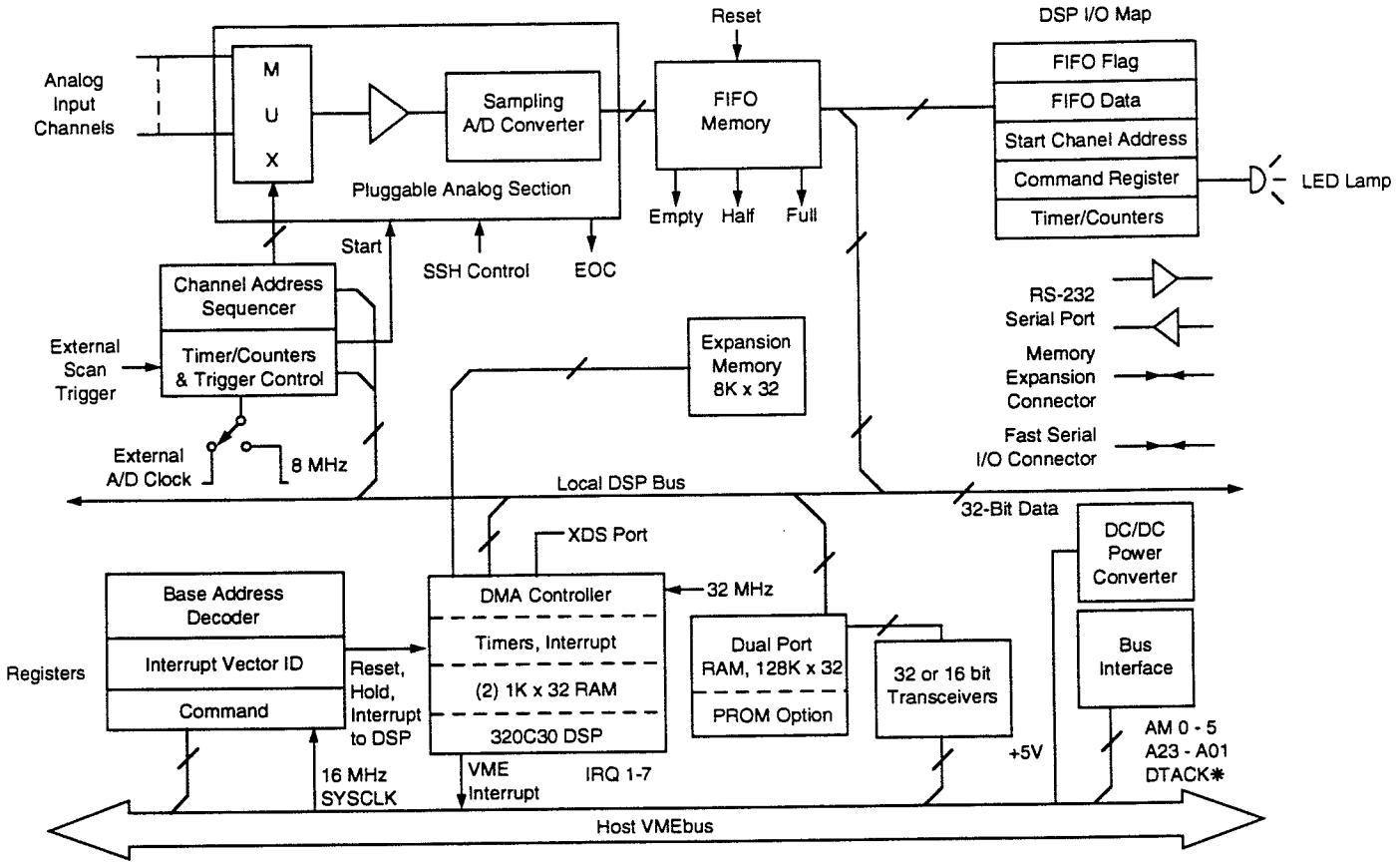



**Figure 1.0 Simplified Block Diagram**


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisanscientific.com

# SECTION 2

# BOARD CONFIGURATION AND INSTALLATION

*Use anti-static protection!*

Although the DVME-630 inputs are protected up to the overvoltage limit, many components on the board can be damaged by static discharge. Use grounding protection when handling the board.

CAUTION: Configuring the board must be done carefully by a person familiar with miniature electronic assemblies. Improper procedures may damage the board and void the warranty.

## 2.0 Determining Your Host Environment

Before installing the board in your host computer, study your host documentation. Installation must not disrupt the normal operation of your host and must be compatible with its resources. If your development environment differs from the target application, be aware of the differences.

Host compatibility involves both hardware and software. Hardware compatibility covers bus slot selection, memory mapping, bus size transceivers and interrupts.

## 2.1 Bus Slot Selection

*Choose the slot but do not install yet.*

The DVME-630 will operate in any VME bus slot with a full set of bus signals on its J1 (top) backplane connector. If you would like full 32-bit data access, your VME host must also have an active J2 (bottom) connector with VME data signals in the center row. The 16 MHz VME bus SYSCLK signal on pin J1-A10 MUST be present and is provided by all standard CPU boards. Generally, CPU boards and bus controllers are installed in the left end of the card cage (slot 1).

The DVME-630 requires a CPU board or other bus master to read data from the output buffers. If possible, position the DVME-630 away from other boards, especially those with high speed/high current digital logic. This will offer slightly improved noise performance. Avoid the VME switching power supply if possible.

The DVME-630 daisy-chains the VME bus Bus Request, Bus Grant and Interrupt signals. Therefore, boards further away from the CPU than the DVME-630 (lower priority) will have their access preserved when the 630 is installed. If you have empty slots between the CPU and the 630, and you want DVME-630 interrupts, make sure the backplane jumpers are installed to carry these signals across empty slots. Rearrange the sequence of boards to vary the hardware priority of DVME-630 interrupts. Higher priority at the same interrupt request level is nearer to the CPU board.

## 2.2 Host Memory Map Compatibility

The DVME-630 appears to the host as 256K contiguous 16-bit words (or 128K x 32-bit longwords) of dual ported random access memory (DPR) mapped in the host's memory. The bottom location, called the BASE address, is selected by on-board DIP (Dual Inline Package) switches, discussed in a later section. The BASE address may be located anywhere in 24-bit (16 megabyte) address space on 256K-word boundaries.

This contiguous memory block must not conflict with other system memory. The block cannot overlay or respond to the same address codes as other memory or peripherals in the system. Usually the 630 is located in high memory, beyond any host system Random Access Memory. Note that the highest BASE address allowed is $FF0000 (hexadecimal). When locating the DVME-630 BASE address, consult your host system memory map for an area which accepts standard I/O addressing. Verify the host Address Modifier (AM) code used for this area, normally codes 39h or 3Dh.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Avoid OS Allocation

On host power up, many operating systems (OS) test for how much contiguous read/write system RAM there is, starting from location zero. The top of RAM is one less than the first location which fails R/W testing. Do not map the DVME-630 exactly adjacent to this highest RAM address. Otherwise the OS may think the 630 DPR is available for system RAM purposes. The DVME-630 must be an unallocated OS block. Do this by leaving a "hole" between system RAM and the 630 DPR. This area will fail R/W testing and the OS will not try to allocate the DPR.

A typical host system map with this "protective hole" will look like this:

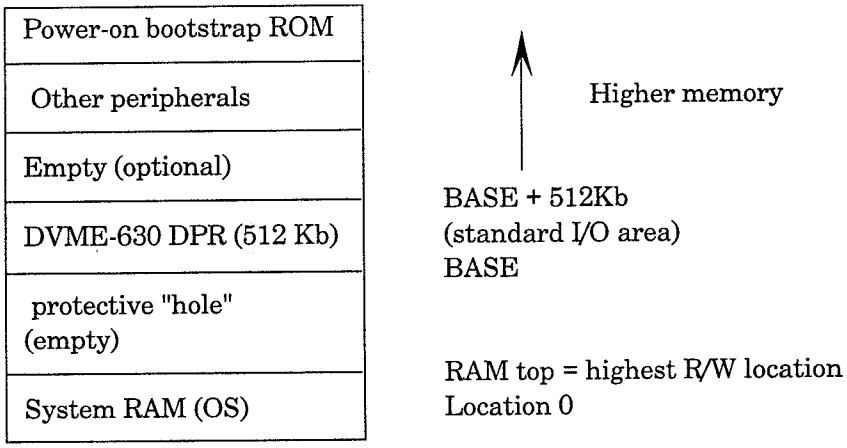



**Figure 2.1 Typical Simplified Host Memory Map**

On some systems, the memory size is set in DIP switches or in ROM. These should not be changed for the DVME-630.

## 2.3 Host System Timing

The basic timing relationship between the host and the DVME-630 should be understood before discussing interrupts. We need to determine if the host system is fast enough for both the 630 and all other non-630 tasks (disk, screen, math, communications, etc.). This determination can be done either by empirical testing or by software throughput budgeting. Ultimately, the empirical testing must succeed.

Assuming a continuous, non-stop stream of data from the 630, the sequence running on the host is something like this:

```
        Initialize the host and 630.
        Start 630 sampling.
LOOP:   If the 630 buffer overflowed, report the error and exit
        Wait for buffer flag to be READY.
        When ready, process the buffer.
        After processing, reset the buffer ready flag to EMPTY.
        Do other non-630 tasks (disk, screen, math, etc.).
        If more data is needed, go to LOOP.
        Else, quit
```


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisanscientific.com

Notice that all during this loop, the 630 continues collecting data with no sample loss. "Process the buffer" can mean immediate in-place processing or a quick block transfer then later processing. If possible, process the buffer in-place to avoid further delays physically moving data. After buffer processing, the host quickly frees up the buffer for the next block of 630 data by setting the flag to EMPTY. Detecting the buffer ready flag can be done either by polling the raw flag in the DPR or by polling a system flag set by the buffer ready interrupt. Either way, a flag will be polled.

A fundamental concept is that the downstream process (the VME host) must be faster in the average than the upstream DVME-630. As long as the host is faster, there will be a little extra capacity and the first item in the LOOP section will test false (buffer not ready yet). If the the buffer is always ready the first time it is tested, host speed is not sufficient and data will overflow.

The DVME-630 produces a hardware interrupt and a software buffer ready flag when data is ready. Either one may be used. The interrupt avoids polling of the DPR but a system flag set by the interrupt usually must be polled anyway. At the very highest speeds, interrupts are probably a little slower than polling the DPR.

### 2.4 Host Interrupt Compatibility

If you have several concurrent tasks present in your application, buffer ready interrupts from the 630 are a desirable way to maintain total system bandwidth and avoid lost A/D data. If interrupts are not used, data may be lost by inadequate polling. Polling should be avoided if you have several other asynchronous processes with unknown delays in your system.

The DVME-630 interrupt usually is used to indicate that a buffer is ready to be block transferred at highest possible speed to the host or processed in place. If the buffer is not moved quickly, the DVME-630 may have an overrun error. The host moves the buffer then immediately flags the 630 that the buffer is free. By software, the 630's interrupt may be reassigned to a different event other than a buffer ready flag.

If you use DATEL's DVME-630 software, most of the interrupt processing is done for you. Interrupts require the following decisions:

1. The Interrupt Request line to VME bus (IRQ1 through IRQ7) must be selected on the 630 by DIP switch and jumper (P7) - see Figure 2.5. This request level determines software priority of the interrupt. Board slot position determines hardware priority for interrupts at the same request level.

2. A vector ID code must be loaded by your host program into the DVME-630's Interrupt ID register after power-up. This ID code corresponds to the address of the interrupt service routine (ISR) which handles interrupts from the 630. The ID code is used by the CPU (after 2 left shifts) to develop the ISR address. This ISR address is either an absolute address or an offset from the CPU's Vector Base Register (VBR) for 680X0 CPU's. The VBR also must be loaded by host programs before interrupt usage. Interrupts must be enabled in the DVME-630 command register before usage.

3. Interrupts with or without a Real Time Operating System

   The ISR is written by the user (or derived from DATEL's software) and must be resident in memory before DVME-630 interrupts are sent. If you have decided to use interrupts, they will be handled in software in one of two ways:

   A. If you are NOT using a Real Time Operating System (RTOS), the interrupt controller and CPU registers in your host must be initialized with low level code as described.


Artisan Scientific - Quality Instrumentation Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

B. If you ARE using an RTOS, interrupts are handled more symbolically and you probably will not have to do much low level controller programming. You must still supply code to retrieve output data, reset the DVME-630 flag to EMPTY plus arbitrate with the RTOS. You will need to decide a priority for the DVME-630, how it communicates with other system tasks and how its status is managed.

Be aware that processing interrupts through an RTOS can take extra time because of overhead. Some RTOS's cannot switch tasks and execute the ISR very quickly (below 50 microseconds). You should design the DVME-630 output buffering to use larger blocks with less frequent interrupts. The actual block transfer should use the quickest hardware method available on your system (usually a host DMA controller is fastest).

With multiple random system interrupts, RTOS delays are not always predictable or easy to measure. If no other system tasks are present, straight non-interrupt polling code is usually simpler and faster.

All RTOS's operate in a similar manner however exact details of RTOS programming vary greatly between different suppliers. Consult your RTOS documentation.

4. When to do the Block Transfer

If the response to the interrupt is to do a block transfer, then there are two basic strategies to handle the transfer. Either the interrupt sets a data ready system software flag or semaphore which the RTOS processes later (at some time after returning from the interrupt). OR the ISR does the block transfer immediately while still in the ISR.

Many RTOS's (notably real time adaptations of UNIX) only allow the first method presumably because ISR's must exit very quickly. DATEL's example software uses the first method. The second method may be faster because it moves the data immediately (and therefore frees the buffer sooner). If the RTOS is lightly loaded and the DVME-630 is the main task, both methods will run at nearly the speed capacity of the host because the block transfer request occurs soon after returning from the interrupt.

The first method allows a higher priority RTOS task to override the block transfer using the scheduler system flags (not the CPU's hardware "interrupt pending" flag). The interrupt is unlikely to be missed. But for fast A/D data acquisition, there should be few other high duty cycle tasks running.

Some users may feel that an ISR is not the appropriate place to do a block transfer because it takes too long. However, the block transfer is perfectly legal if interrupts are reenabled while still in the ISR before the block transfer. This allows the ISR to be interrupted by other well-behaved ISR's which preserve all registers. The hardware interrupt is only a method to force immediate prioritized execution of a code module. Once the ISR is running and interrupts have been reenabled, the ISR acts like any other code.

You should avoid block transfers within the ISR if different source and destination addresses are used with each interrupt and these addresses cannot be communicated to the ISR. The DVME-630EXEC software posts the upload buffer pointer and length in a fixed area with each interrupt.

## 2.5 Jumpers and Switches

IMPORTANT: In the following sections describing jumper installation, use only the jumpers shown. Omit other jumpers not listed.

Jumpers consist of two parts - two or more gold square pins soldered vertically onto the board and a removable "Berg" plug which slides over pairs of pins. Each unique position of the jumper is identified by a label on the PC board and on the Assembly Diagram.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Jumpers offer high electrical integrity in minimal area for situations which cannot use a DIP switch or software selection. The plugs may be replaced with wire wrap connections if desired. Avoid losing the plugs. If a plug is not to be used, slide it over an unused single pin off to the side so that it makes no connection between two pins.

In the following sections, please refer to the Assembly Diagram (Figure 2.2) to locate switches, jumpers and other hardware. Position the board with the VME bus connector, P1 at the lower right and the front panel across the top. You should carefully check ALL factory jumpering before inserting the board. Setting up the board configuration should only be done by a person skilled at miniature electronic assembly. Improper procedures may damage the board.

DIP switches are set using a small tool such as a miniature screwdriver. The switches may be either small slide switches or rockers. The switch body will be labeled with the "ON" direction. A dot or arrow or flat on the switch body may be present designating pin 1 (switch 1) on the DIP package. When a switch is ON, it decodes a logic "0". The OFF position decodes logic "1".

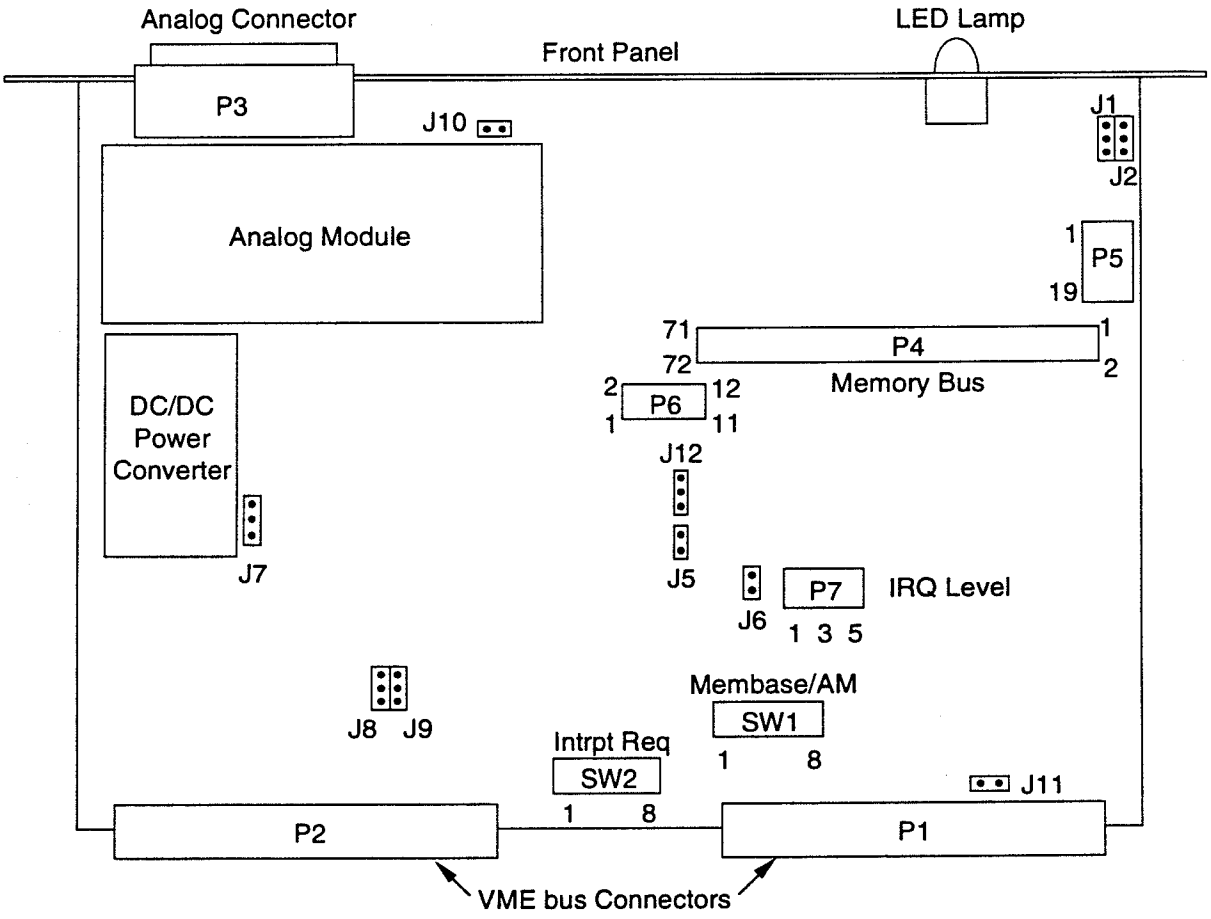



**Figure 2.2 DVME-630 Jumper and Switch Locations**


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

## 2.6 Memory Base Address Selection

DIP switch SW1 is used for BOTH the BASE address and the Address Modifier. SW1 is located in the lower center of the board. Set the desired BASE address code according to the following table:

| Address Bit | A23 | A22 | A21 | A20 | A19 | Not used |
|---|---|---|---|---|---|---|
| DIP switch SW1 | 3 | 4 | 5 | 6 | 7 | 8 |

ON = "0", OFF = "1"

Address bits 18 through 1 are decoded for the on-board DPR but are set to zero when determining the BASE address.

Example: To set the memory base address at 800000h (8 Mb), set switch 3 = OFF and switches 4 - 8 = ON as follows:

| SW1 | 3 | 4 | 5 | 6 | 7 | |
|---|---|---|---|---|---|---|
| Address Bit | A23 | A22 | A21 | A20 | A19 | A18-A01 |
| | OFF | ON | ON | ON | ON | set to 0 |
| Binary | 1 | 0 | 0 | 0 | 0 | |
| Hex | "8" | | | | "0" | |

Notice that the next available BASE address is 880000h (8.5 Megabytes).

**Figure 2.3 BASE Address Switch Assignments**

## 2.7 Address Modifier Selection

The DVME-630 may be accessed in either privileged (supervisory) or non-privileged (user) modes. These modes have Address Modifier codes which are described in the VME bus specifications. If you use a device driver, remember that the CPU will be in supervisor mode during an interrupt service routine triggered by the 630. Normally you may leave the AM jumpers in the factory configuration.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisanscientific.com

Important: Select ONLY the switches shown:

| Switch SW1 1 2 | Responds to AM Code | Access Description |
|---|---|---|
| Off Off (factory supplied) | 39h<br>3Ah<br>3Dh<br>3Eh | Standard non-privileged data access<br>Std. non-privileged program access<br>Standard supervisory data access<br>Standard supervisory program access |
| Off On | 39h<br>3Ah | Standard non-privileged data access<br>Std. non-privileged program access |
| On Off (not used) | 29h<br>2Dh | Short non-privileged access<br>Short supervisory access |
| On On (not used) | 29h | Short supervisory access only |

"0" = ON, "1" = OFF

**Figure 2.4 Address Modifier Selection**

## 2.8 Interrupt Level Selection

The DVME-630 supports a fully standard VME 8-bit vectored interrupt. The vector ID register is described in a following section. If you are writing your own code, DATEL recommends that you develop your application first in status mode by polling the DPR flag addresses to detect when data is ready. When status mode is working flawlessly, then you may use interrupts instead.

Interrupts may be generated on levels 1 through 7. DIP switch SW2 in the bottom center of the board is used to set the interrupt request line. Jumper field P7 in the right center of the board selects the request level. To generate interrupts, BOTH the command register interrupt enable bit AND the interrupt level and request lines AND the vector ID register must be set properly. Note the order of the jumpers in P7:


Artisan Scientific - Quality Instrumentation Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

P7
1 3 5

| Interrupt Level | IRQ Selection | P7 Jumpers 1 | 3 | 5 | Jumper in = "0" Jumper out = "1" |
|---|---|---|---|---|---|
| 1 | SW2-2 ON | IN | IN | OUT | [highest priority] |
| 2 | SW2-3 ON | IN | OUT | IN | |
| 3 | SW2-4 ON | IN | OUT | OUT | (factory supplied) |
| 4 | SW2-5 ON | OUT | IN | IN | |
| 5 | SW2-6 ON | OUT | IN | OUT | |
| 6 | SW2-7 ON | OUT | OUT | IN | |
| 7 | SW2-8 ON | OUT | OUT | OUT | |
| No intrpt. | SW2 all OFF | | | | |

SW2-1 is not used

IMPORTANT: Leave all other SW2 switches OFF except the selected IRQ interrupt switch shown. If you do not want interrupts, leave ALL SW2 switches OFF. Interrupts may also be disabled in the command register.

**Figure 2.5 Interrupt Level Selection**

## 2.9 16- or 32-bit Data Bus Selection

*NOTICE*
*Do not use 32-bit access modes when the bus transceivers are set for 16-bit mode.*

The DVME-630 will operate with either 16- or 32-bit access to its Dual Port RAM. These two modes are selected by the bus transceiver positions and jumper J11 (see below).

In the 16-bit mode, the DVME-630 does not use the VME bus P2 connector. In 32-bit mode, the P2 center row of pins is used for data. In 32-bit mode, both 16-bit and 32-bit transfers are supported if the host bus is fully compatible with the VME bus DS0*, DS1* and LWORD* sizing signals. Byte transfers are not available for either mode. The VME bus 16 MHz SYSCLK signal is REQUIRED for all modes.

Be careful writing "C" code which does memory reference pointer instructions. An "int" object may generate either 16- or 32-bit instructions, depending on the compiler. A "short int" is almost always 16 bits and a "long" is usually 32 bits. Also, the addressing should obey VME guidelines. All VME addresses are byte resolution. All 16-bit word addresses must be even and 32-bit addresses must be an integer multiple of four. If you try to split a 32-bit "long" over the 32-bit boundary, undefined results are possible.

The bus transceivers are set for 16-bit mode from the factory. By removing the U44 and U45 integrated circuits and reinstalling them in adjacent sockets (and changing J11), the alternate access size is selected. If you are not familiar with how to remove and install IC's in sockets, get some experienced assistance. The IC's are easily damaged by improper insertion. Be particularly careful of bending and hiding a pin under the body of the IC. The sockets are not meant for a large number of insertions. Therefore make your selection and avoid changing it.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

32-Bit Mode

For 32-bit data bus size using both P1 an P2, install transceivers U44 and U45 in the sockets NEARER the P2 connector. Then install the J11 jumper.

16-Bit Mode

For 16-bit data bus size using only P1, install transceivers U44 and U45 in the sockets FURTHER from the P2 connector. Then remove the J11 jumper.

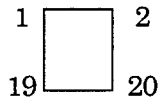
### 2.10 82C54 Timer Clock Source Jumpers

The A/D start clock source for the on-board 82C54 timer is normally supplied from the local 8 MHz crystal oscillator. The timer may also be clocked from an external logic source on connector P5 pin 12. This provides synchronous trigger operation or phase tracking to external sources.

These jumpers are located on the right edge of the DC/DC converter.

| Clock Source | Jumper |
|---|---|
| 8 MHz onboard oscillator | Close J7 pins 1-2, open 2-3 (standard) |
| External clock from P5-12 (10 MHz maximum) | Close J7 pins 2-3, open 1-2 |

The External TRIGGER* may also be supplied through P5-9. The trigger source (internal or external) is selected through the internal command register. The orientation of P5 is as follows:




Connector P5 (see Section 3.3.3)

If possible, BOTH the trigger and A/D start clocks should be derived from the same source, either internal or external. This will avoid phase problems and missed triggers.

### 2.11 FIFO Flag Jumpers

The half full and full status bits from the A/D FIFO memory are available as a status flag or DSP interrupt through DSP INT1. Normally the half full bit is used since data would be lost while responding to the full flag. Half full is used to start a local block transfer from the FIFO. The FIFO continues collecting A/D samples after the half full bit is true and during the transfer.

IMPORTANT: HF occurs at half full +1. For the 1024-sample FIFO, half full is 513 samples. 2049 samples are guaranteed at HF for the 4096-sample FIFO. These jumpers are located on the lower left corner of the board.

| Flag | Jumper |
|---|---|
| Half full | Close J8 2-3, open 1-2 (standard) |
| Full | Close J8 1-2, open 2-3 |


Artisan Technology - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

### 2.12 A/D Channel Advance Jumper

These jumpers determine when the channel address sequencer increments relative to the A/D start or EOC signals. The selection is optimized for the settling time characteristics of the A/D converter. The jumpers are located on the left edge of the analog module. Normally the jumpers should only be changed if a different analog module is installed. Observe the settling time requirements after channel advance and before the next A/D start.

| Model | Advance Source | Jumper |
|---|---|---|
| DVME-630A (SSH) | Start convert trailing edge | Close J9 pins 1-2 |
| All others | EOC true leading edge | Close J9 pins 2-3 |

### 2.13 DSP Start Mode Jumpers

The DSP is normally used in the "microprocessor" mode where it receives its power-on instructions from location 0 in primary on-board memory (either PROM or SRAM). The user may instead have the DSP mask-programmed by Texas Instruments to start up from internal PROM in the "microcomputer" mode. This jumper is normally not changed.

| Start Mode | Jumper |
|---|---|
| External memory | Close J10 (standard) |
| Internal PROM | Open J10 |

### 2.14 DSP Reset Jumper

This jumper is a convenience for hardware testing. Closing the jumper momentarily forces a hardware reset of the DSP. The VME bus command register also does this under software. This jumper is normally left open.

| Status | Jumper |
|---|---|
| DSP run | Open J6 (standard) |
| DSP reset | Close J6 |

### 2.15 EPROM Jumpers

User-programmed read-only memory may be installed in place of one bank of 32K x 32 static RAM starting at location 0. This will eliminate the download requirement at start-up. Four 32K x 8 EPROM's are required at locations U2, U4, U6 and U8. 28-pin EPROM's may be obtained from Wafer Scale Integration (part number WS57C256F), AMD, Cypress, Toshiba, TI and others. DATEL will review custom programming requirements under special order.

The DSP defaults to seven wait states at reset and is software programmable according to certain memory boundaries. The wait states may be selected to the EPROM speed. This allows full speed from the static RAM portion of memory and reduced rates for slower EPROM's.

| Page 0 Memory | Jumpers |
|---|---|
| Static RAM (standard) | Close J1 2-3 and J2 1-2 |
| EPROM | Close J1 1-2 and J2 2-3 |


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

## 2.16 Board Installation

*Remember anti-static protection.*

If you have followed all the steps above, all jumpers should now be properly positioned. Confirm that your host computer and its operating system (OS) function normally. Shut off host power.

Using the board slot you have chosen, carefully slide the DVME-630 into the VME card guides. Make sure your hands are grounded before handling the board. Do not force the board. If it does not seat properly, observe the J1 connector to see that it is receiving the board.

Now reapply power with the DVME-630 installed. Confirm that your OS boots up normally. Verify the memory size message if it is normally present. If the OS fails to boot up normally, immediately turn power off and determine the cause. Remove the DVME-630 and check for overheated components. The DC/DC converter and A/D section normally run quite warm.

Notice that the DVME-630 LED lamp will remain DARK and can only be turned on by software. Do not make any connections to the front panel until you have read the section on I/O connections. If you want to quickly determine that the board is addressed properly using your Monitor/Debugger, refer to the programming section.


Artisan Technology - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

# SECTION 3

# HOST ARCHITECTURE AND I/O CONNECTIONS

## 3.0 Memory Mapping - Logical and Physical Addresses

High performance VME CPU boards and operating systems (OS) such as UNIX work together to allocate memory when a program needs it. Many new CPU boards have memory management units (MMU) which map real physical memory as required. The address code on the pins of the CPU or in its address registers may be different than the actual physical memory address being accessed. Local memory on the CPU board may be treated differently than memory on another VME board. Some very large memories use paging or segment registers to switch a bank of memory. The OS and MMU work together to provide logical addresses rather than physical addresses. If you have a choice, select an OS which has straightforward access to physical memory.

Your program must access the DVME-630 as real physical memory. To do this, you may need a real-to-logical address pointer conversion function provided with your OS. This may be a device driver which must be loaded at boot time. If your system uses logical addressing, EVERY access to the DVME-630 in your high level program will need to use this conversion. Some systems include a "shared physical memory" function for this purpose. On DATEL's Force UNIX 68030 system, built-in IOS and IOCTL drivers supplied by Force were used to do the physical-to-logical conversion and for 32-bit access.

On still other VME hosts, physical I/O is mapped into high 32-bit addressed memory beyond the address range of the DVME-630 24-bit address decoder. This apparent paradox is resolved by the six VME bus Address Modifier bits (AM0 - AM5) which are encoded for 32-bit "standard I/O access". In effect, the VME bus CPU generates a 32-bit address code but mapping decoders generate the proper AM code before it leaves the CPU board onto VME bus. This allows physical I/O such as the DVME-630 to respond.

NOTICE: You will find MEMBASE hardwired at 0xFC800000 in the DATEL sample programs for this reason. On DATEL's Force UNIX system, this is the area for "standard I/O access". Your system may differ and you should edit the programs accordingly.

If you have other physical peripherals such as dual-ported RAM interfaces for SCSI disk drives, examine the source code of their device drivers to understand how physical peripherals are accessed.

The address decoder bits on the 630 respond to the middle address bits A23 to A19. The upper bits A31 to A24 are used by your system's address generators (but not detected by the 630). Your program however must generate ALL the bits properly. Simpler host systems with only 24 address bits will not need these tricks.

*NOTICE*

*Do not access the DPR until the DSP is reset and put into hold using the command register.*

At power up, the DSP is in the run mode. This is for automatic running of applications which have replaced the lower bank of memory with EPROM's. For other applications which use the existing all-RAM DPR, there will be no valid DSP control code in the DPR at power up. The DSP will attempt to run but operation will be undefined.

Initialize the DPR first by writing 00C0h into the command register at MEM BASE+7FFFCh. This places the DSP into reset and hold mode. Next, write valid control code into the DPR starting at location zero (the boot vectors). Usually this will be from a COFF object file.

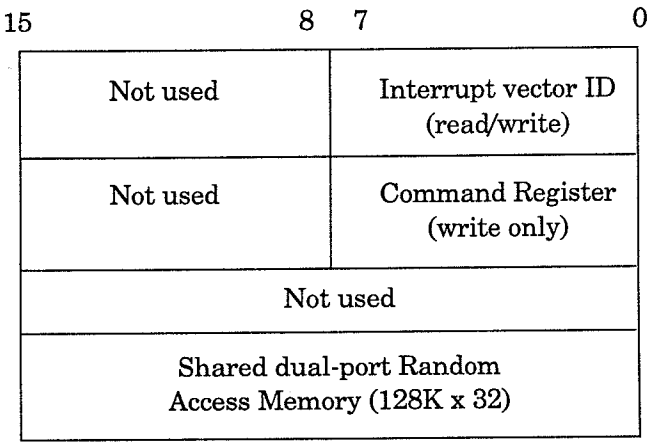
Note that if the VME CPU is an Intel 80X86 or other machine which swaps bytes within words (and words within longwords), you must write a code of C000h, not 00C0h into the command register. The actual byte of C0h is written to byte address MEMBASE+7FFFDh but you must use word or longword access.


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Once all code and data are successfully loaded into the DPR and the VME host is ready to process DVME-630 output, transition the DSP to run mode by writing 0000h to the command register.

### 3.1 DVME-630 VME bus Memory

As seen from the VME bus, the DVME-630 consists of two contiguous sections in one block of memory. The lower section consists of almost 256K words of dual port random access memory (DPR). The upper section consists of hardware registers. The DPR will be fully defined by software in a later section. The DPR map is as follows:




VME bus Interface

**VME bus word address:**

MEMBASE + 7FFFE,Fh

MEMBASE + 7FFFC,Dh

MEMBASE + 7FFF8-Bh

MEMBASE + 7FFF6,7h

MEMBASE + 0 = DSP location 0

Notice that the DSP location zero is at the bottom of the DPR.

### 3.1.0 VME bus Command Register: (Write only to MEMBASE+7FFFDh)

| 15 8 | 7 | 6 | 5 | 4 | 3 2 1 0 |
|---|---|---|---|---|---|
| Not used | DSP Run/ Rst | DSP Hold Req. | DSP Intr Req. | VME Intr Enbl | Not used |

Bit 4: 0 = disable VME bus interrupts, 1 = enable VME bus interrupts

Bit 5: 0 = no request, 1 = force interrupt to DSP

Bit 6: 0 = no request, 1 = request DSP hold

Bit 7: 0 = run DSP, 1 = reset DSP

Both the command and vector registers default to all zeroes at power up.

### 3.2 Initial Software Tests

At power up, the DSP will attempt to run but operation will be undefined until a valid program is loaded. Using your debugger, put the DSP in reset by writing 0080h to the command register near the top of the DPR. If you get a bus timeout error message, this indicates that the addressing is not working. The bus timeout occurs when DTACK* is not returned from the DVME-630 in sufficient time. More severe bus errors lock up the whole system, forcing a restart.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The next test you should perform is to see if you can write and read from the RAM section of the DPR using your debugger. Write several test values into the bottom of the DPR in word mode and confirm by reading them back. Do not use byte mode access. You may still be able to use the DVME-630 if these debugger tests fail. But the debugger access is quick and requires no software yet.

### 3.2.0 Your Debugger

Your debugger must be able to send the absolute physical address code to the DVME-630. Some host VME systems have both a low level PROM-based debugger and a disk-based debugger. The PROM debugger is usually available before the operating system is booted. Try this debugger first.

Sometimes the disk-based OS debugger will not allow you to read/write to absolute phyiscal memory and instead uses only logical addresses. Occasionally the OS disk debugger (if that is all you have) can be fooled by writing a tiny assembly program to read/write absolute memory. You can even manipulate the MMU to get absolute physical access if you understand it thoroughly. The debugger must have an ASSEMBLE command for this to work.

### 3.2.1 Access Size

Some debuggers will let you select byte, word and longword access. The DVME-630 can only accept words or longwords. If your debugger only does byte access, use the ASSEMBLE command to do word or longword read/writes.

Some systems physically trap outgoing addresses and will cause a hardware exception for attempted access to what the OS thinks is "non-allocated" memory (i.e., the DPR). Be careful doing these operations. Save any important files to backup storage first.

If you do not have a low-level debugger, try writing a very simple "C" program to read/write data in the DPR using pointers and the logical-to-physical address conversion function if needed. Your "C" compiler may also have an in-line #assem directive to allow real physical memory access from a few assembly instructions.

Be aware of big and little endian anomalies (the sequence of LSB/MSB bytes in words and longwords - LSB's are highest on 680X0 machines). Once you can read/write to the DPR, try to compile and run the programs on the disk (after editing them for any base address changes).


Artisan Scientific - Quality Instrumentation Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

## 3.3 I/O Connections

### 3.3.0 RS-232-C Serial Port

The header pin field at J12 provides RS-232-C transceivers for serial communications. Two data lines are supplied plus ground. ±5V RS-232-C data levels are used. No hardware handshakes are supplied and most applications do not need them. Handshakes may be software-encoded using XON/XOFF or ACK/NAK flow control characters. The serial ports require a "software UART" character generator subroutine (see the STDIO.ASM file available from the DATEL Mansfield Subsystems Application Dept.) which sets the data format and baud rate. The ports may also be assigned for any general purpose digital I/O usage. DSP CPU register instructions are required for access. This port has been employed as a diagnostic Monitor port to a CRT terminal. Other possible usage would include program or data uplink/downlink.

| J12 Pin | Function | DSP Bit |
|---|---|---|
| 1 | TxD Transmitted data | DX0 |
| 2 | RxD Received data | DR0 (Note - close J5) |
| 3 | GND Logic Ground | Ground |

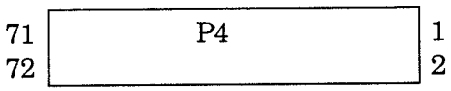
Install J5 to connect DR0 for the RS-232-C serial port.

### 3.3.1 Memory Bus Expansion

Connector P4 provides full access to the DSP memory bus. CAUTION: These signals are not buffered and would require transceivers to access another board. The external interrupt signal on P4-2 connects directly to the DSP INT3 input.

Unlike the PC-430, an expansion bus interface is not provided on the DVME-630.

The orientation for P4 is as follows:





Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

| DSP Signal | | | DSP Signal |
|---|---|---|---|
| INTTRIG* | 1 | 2 | EXTINT* (connects to DSP INT3) |
| LD0 | 3 | 4 | LD1 |
| LD2 | 5 | 6 | LD3 |
| LD4 | 7 | 8 | LD5 |
| LD6 | 9 | 10 | LD7 |
| Ground | 11 | 12 | Ground |
| LD8 | 13 | 14 | LD9 |
| LD10 | 15 | 16 | LD11 |
| LD12 | 17 | 18 | LD13 |
| LD14 | 19 | 20 | LD15 |
| Ground | 21 | 22 | Ground |
| LD16 | 23 | 24 | LD17 |
| LD18 | 25 | 26 | LD19 |
| LD20 | 27 | 28 | LD21 |
| LD22 | 29 | 30 | LD23 |
| Ground | 31 | 32 | Ground |
| LD24 | 33 | 34 | LD25 |
| LD26 | 35 | 36 | LD27 |
| LD28 | 37 | 38 | LD29 |
| LD30 | 39 | 40 | LD31 |
| Ground | 41 | 42 | Ground |
| +5V dc power | 43 | 44 | +5V dc power † |
| LSTRB* | 45 | 46 | LR/W* |
| LA19 | 47 | 48 | LA18 |
| LA17 | 49 | 50 | LA16 |
| Ground | 51 | 52 | Ground |
| LA15 | 53 | 54 | LA14 |
| LA13 | 55 | 56 | LA12 |
| LA11 | 57 | 58 | LA10 |
| LA9 | 59 | 60 | LA8 |
| Ground | 61 | 62 | Ground |
| LA7 | 63 | 64 | LA6 |
| LA5 | 65 | 66 | LA4 |
| LA3 | 67 | 68 | LA2 |
| LA1 | 69 | 70 | LA0 |
| Ground | 71 | 72 | Ground |

CAUTION - Signals are not buffered.

† (1 Amp max. total all pins)
"*" means low true.

**Figure 3.1 P4 Primary Memory Bus Interface**


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

### 3.3.2 Analog Input Connector, P3

All DVME-630 models use the DB-25S front panel connector. Pin numbering is shown as viewed from the front panel:

(Channel addresses use the notation, Single-ended/Differential.)

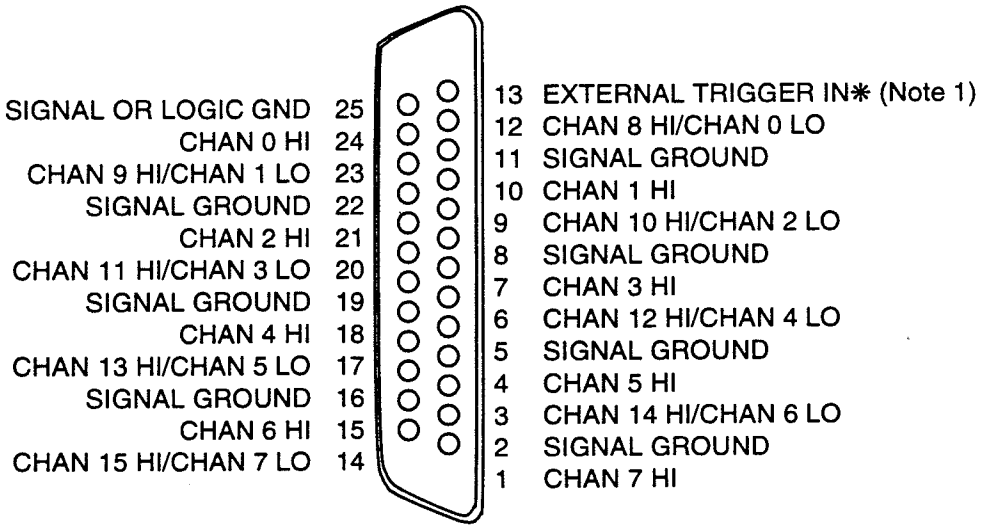



**Figure 3.2 Analog Input Connector, P3**

Note 1: The External Trigger Input was not connected up through artwork revision C boards. This trigger input is also available on P5-9.

### 3.3.3 DSP Serial Port, External Clock and Trigger, P5

| | | | |
|---|---|---|---|
| CLKX1 | 1 | 2 | DR1 |
| CLKR0 | 3 | 4 | FSR0 |
| CLKX0 | 5 | 6 | FSR1 |
| GROUND | 7 | 8 | GROUND |
| EXTTRIG* IN | 9 | 10 | CLKR1 |
| DX0 | 11 | 12 | EXTPCLK* IN |
| FSX0 | 13 | 14 | FSX1 |
| GROUND | 15 | 16 | DX1 |
| DR0 | 17 | 18 | GROUND |
| +5V dc POWER OUT | 19 | 20 | +5V dc POWER OUT (1/2 Amp max) |

**Figure 3.3 Header Connector, P5**

The EXTPCLK (pin 12) is divided by 82C54 timer 1 which then generates the A/D start clock (STCONV). Do not mix internal/external triggers and A/D clocks for applications which are sensitive to phase. All other signals refer to the DSP high speed serial ports.

Caution: These signals connect directly to the DSP and require external buffering.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

# SECTION 4

# PROGRAMMING INTRODUCTION

## 4.0 Hardware Organization Overview

The DVME-630 is unique in that there is no local software in the system at power-up since there is no on-board Read Only Memory (ROM). All executable code, all libraries, vectors and tables must be downloaded from host disk after power up. To avoid the download for dedicated applications, ROM may be installed in the first page in place of some dual port RAM. However, the downloaded RAM method offers many advantages. The ROM is usually slower than the SRAM.

The DPR is the primary area for control and data transfer. The full local software system used by the DSP also shares the lower DPR.

## 4.1 VME bus Command Register (Write MEMBASE + 7FFFDh)

Please refer to Section 3.1.0. This register includes four write-only bits.

| 15 8 | 7 | 6 | 5 | 4 | 3 2 1 0 |
|---|---|---|---|---|---|
| Not used | DSP Run/ Rst | DSP Hold Req. | DSP Intr Req. | VME Intr Enbl | Not used |

VME bus Interrupt Enable [Bit 4]
0 = Disable interrupt
1 = Enable interrupt

An interrupt to VME bus is caused by toggling the DSP TCLK 0 output from 0 to 1 to 0. The interrupt occurs on the rising edge. The DVME-630 Executive system uses this interrupt to indicate that a buffer is ready or from the INT2PC function. User-written software may reassign the interrupt to any other purpose. Interrupt switches and jumpers must also be set properly on the board to use VME bus interrupts.

DSP Interrupt Request [Bit 5]
1 = No request
1, 0, 1 Sequence = DSP interrupt request

An interrupt to the DSP INT 0 input may be sent by setting bit 5 to a one then immediately back to zero. The DSP will process the interrupt if enabled using the DSP interrupt enable register. Presently this interrupt is unassigned and not supported by the DVME-630 Executive system and is available for user programming. However, future Exec revisions may use this function.

DSP DPR Hold Request [Bit 6]
0 = (no hold) Allow DSP access to primary memory.
1 = (hold) Lock out primary memory access by DSP and execute wait states.

There are two hardware methods for the VME bus to read or write from the DPR during a simultaneous access attempt by the DSP. These use either program hold or dynamic hold. Program hold indefinitely locks out the DSP from DPR access and is useful for large block transfers such as host DMA. During program hold, the DSP executes wait states. The program hold bit 6 in the VME bus command register enables and disables DSP access to the DPR using host VME bus software. Hold is normally only needed for block operations such as memory testing by the host which may overrun DSP executable code.


Artisan Technology - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The DVME-630 also features automatic dynamic hold regardless of the state of bit 6. Dynamic hold briefly locks out the DSP from the DPR with each simultaneous access by the VME host. Dynamic hold is toggled with each access, thereby allowing the DSP to continue running between VME DPR accesses. Dynamic hold is in effect all the time and is transparent to the host. Because of the extremely high speed of the DSP, VME access will normally not experience any significant slowdown for DPR operations except simultaneous block transfers on both sides or other high duty cycle operations.

There is of course no slowdown if the VME host is accessing other non-630 portions of the host memory. Also, the DSP may execute at full speed from either internal ROM or RAM while DPR hold is in effect. Hold only concerns DPR access. To continue DSP execution, make the DSP branch to code that is not located in the DPR before the hold request. Note that DPR access includes both control code fetches and data read/write.

The DSP must be set to hold with the DSP in reset before and during download of the initial program image to the DPR. It must be reset after the program load.

| | |
|---|---|
| DSP Reset/Run | 1 = Reset DSP |
| [Bit 7] | 0 = Run DSP |

On power up, the DSP will attempt to run but operation will be undefined until valid code is downloaded and the DSP is reset. The reset/run bit 7 forces the DSP to begin execution from a known state at MEMBASE + 0 and allows recovery of the DSP from hardware lockups.

Allow several microseconds for the reset state to propagate through the DSP before running.

### 4.2 Local DSP Memory Map

Memory which is local to the DSP consists of the DPR (which is shared with the VME host) and private DSP memory (which cannot be directly accessed by the VME host). In the private section is the 8192 by 32-bit expansion RAM, internal 2K x 32 DSP RAM and registers. The registers include internal DSP CPU registers (timers, DMA, interrupts, wait states, etc.) and hardware registers on the board. These last registers include the A/D command register, 82C54 timer/counter registers, and FIFO data.

All addressing for the DSP uses 32-bit longwords. For example, address 10h for the DSP is address BASE + 40h for the VME host. The next DSP address (11h) is four bytes higher (BASE + 44h) in the VME host. To convert a DSP address to byte addressing, multiply the address by four then add the BASE address. There is no 8-bit or 16-bit access by the DSP. The VME host uses either 32 or 16-bit access through data transceivers on the DVME-630. Do not use byte addressing from the VME host.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

VME host Address = BASE + (4 * DSP Address)

| VME bus Address (bytes) | | DSP Address (32-bit doublewords) |
|---|---|---|
| BASE + 0<br>BASE + 7ffffh | 128K x 32 bits SRAM (shared DPR) | 000000<br>1FFFFh |
| No VME access | Not used | 20000h<br>7FFFFh |
| | Expansion Bus 8K x 32 SRAM (MSTRB active) | 80000h<br>801FFFh |
| | Reserved 8K (Not used) | 802000h<br>803FFFh |
| | Expansion Bus On-Board Registers (IOSTRB active) | 804000h<br>805FFFh |
| | Reserved 8K (Not used) | 806000h<br>807FFFh |
| | Peripheral Bus Internal CPU Registers | 808000h<br>8097FFh |
| | RAM Block 0, 1K (internal to CPU) | 809800h<br>809BFFh |
| | RAM Block 1, 1K (internal to CPU) | 809C00h<br>809FFFh |
| | External strobe active (Not used) | 80A000h<br>FFFFFFh |

(All addresses are in hexadecimal)

**Figure 4.1 Local DSP Memory Map**


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

### 4.3 DVME-630 On-board I/O Registers

Local DVME-630 on-board registers which are external to the DSP are the A/D timer/counters, the command register, the start channel address register, FIFO memory, and data registers. All registers are reset to zero at power-up. None of these registers may be directly accessed by the VME host. Instead, downloaded command sequences offer full control.

| DSP I/O Address | Direction | Function |
|---|---|---|
| 804000h | Read/Write | 82C54 Counter 0 |
| 804001h | Read/Write | 82C54 Counter 1 |
| 804002h | Read/Write | 82C54 Counter 2 |
| 804003h | Read/Write | 82C54 Control Word |
| 804004h | Write only | Command Register |
| 804005h | Write only | Start channel address register |
| 804006h | Not used | |
| 804007h | Read only | FIFO A/D Data Register |

(All registers use 32-bit longwords)

**Figure 4.3 DVME-630 On-board I/O Registers**

### 4.4 82C54 Programmable Timer/Counter

Refer to detailed 82C54 programming information elsewhere in this manual. All registers in the 82C54 are 8 bits wide occupying bits 7 - 0 in the DSP longword. Bits 31 through 8 are not used. Use 7 wait states in the DSP if you are writing your own code to control the 82C54.

Several clocks are used in the A/D section. The A/D conversion clock (also called the Pacer, start clock or sample clock) is the timebase of start pulses which initiate individual A/D conversions. The Pacer may be supplied internally or externally. The internal 8 MHz A/D start clock is derived from the DSP crystal clock at H3 (inverted H1) and then further divided by 82C54 Timer 1. For precision phase tracking applications, external clocking should be considered.

The "trigger" timebase (either internal or external) starts each multi-sample array or frame of A/D data. Each single A/D sample will still be separated by the sample clock interval. If that array also uses sequential channel addressing, it is called a scan. The system may use may use one trigger per scan, one trigger for "N" concatenated scans or only a single trigger to start an indefinite number of samples.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

With these features, the user has full control over the A/D sampling rate, the number of samples taken and the scan trigger rate. By software, the following channel address patterns are examples of what is possible:

TIME ⟶

0123 0123 0123...

012301230123012301230123012301230...
(huge non-stop arrays with selectable sampling rate)

012301230123 012301230123 012301230123...

0 0 0 0 0 0...

0 1 2 3 0 1 2 3 0 1 2 3...
(fixed delay between samples and scans)

0 0 0 0... (simultaneous sampling)
1 1 1 1
2 2 2 2
3 3 3 3

82C54 Counter 0 (Read/Write 804000h)

Timer/Counter 0 of the 82C54 is used to count End of Conversion (EOC) pulses from the A/D converter and controls the number of samples per trigger, up to 65,536 samples. In counter mode 2, when the count is reached, the ACQuire gate is closed and A/D conversion stops until the next trigger. Note that if the ACQuire gate is closed and not retriggered open, a buffer may never get filled. Application programs must carefully select the Counter 0 number of samples and the buffer size.

The count value loaded must be one less than the number of samples desired. To collect 1000 samples, load a count value of 999. The minimum possible number of conversions per trigger is three.

This counter may also be disabled for continuous, non-stop A/D sampling of very large arrays beyond 64K. To do this, program timer 0 for mode 4 (write 0x38 to the control register) then DO NOT write the count register. At least one trigger will be required from the C30 Timer 1 to start this mode. Also, DSP Timer 1 must be set up in pulse mode for continuous non-stop triggering.

An ACQ End of Scan (EOS) local DSP interrupt at INT 1 is optionally available when the count is reached.

82C54 Timer 1 (Read/Write 804001h)

Timer 1 supplies start convert pulses to the A/D. This timer acts as a divider to either the external or the internal timebase. This internal timebase is fixed at 8 MHz and is derived from the DSP's H3 output after dividing by two. An external timebase may be accepted up to 10 MHz.

82C54 Timer 2 (Read/Write 804002h)

Pulses from Timer 1 may start a single A/D conversion or multiple conversions, enabled by the "scan" bit 2 in the command register. Timer 2 selects the number of conversions (the minimum is 3 conversions) in the multiple or "scan" mode. The "scan" mode is normally only used with simultaneous sample/hold (SSH) and sequential channel addressing (increment mode, command bit 3 = 1). In scan mode, EOC pulses are circulated back as A/D start pulses (with a small settling delay of about 400 nanoseconds in U25) until all


Artisan Technology - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

SSH channels are converted. Timer 1 sets the scan rate in the "scan" mode. In the non-scan mode (command 2 = 0), Timer 1 issues A/D start pulses.

82C54 Control Word Register (Read/Write 804003h)

The various modes of the 82C54 are controlled through this register. Refer to the detailed descriptions.

### 4.5 DSP Command Register (Write only to 804004h)

| 31 - 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Not Used | LED On | Not Used | FIFO Rst | Int. Sel | Auto Inc. | Scan Enbl | Conv Enbl | Trig Sel |

Important: The command bit 1 (A/D enable) AND the trigger ACQuire gate (see Section 8) must BOTH be enabled in order for start pulses to reach the A/D converter. In addition, the trigger must be selected (command bit 0) according to the trigger source.

| | |
|---|---|
| Trigger Select (Bit 0) | 1 = External trigger<br>0 = Internal trigger |

The internal trigger is generated by Timer 1 (TCLK1) of the DSP, not the 82C54. TCLK1 must be set up in pulse mode for continuous triggering. For a single trigger, TCLK1 should generate only one pulse at the start of sampling. External triggers are supplied by the user and are 10 MHz maximum.

| | |
|---|---|
| Conversion Enable (Bit 1) | 1 = A/D start pulses accepted<br>0 = A/D inhibited |
| Scan Enable (Bit 2) | 1 = Multiple conversions per 82C54 Timer 1<br>0 = Single conversion per 82C54 Timer 1 |

Bit 2 enables or disables the scan mode (see description above). This bit also sets or clears the SSH control bit to the analog module. The scan mode is normally used only with simultaneous sample and hold. The number of conversions per scan is set by Timer 2 (minimum of 3). The leading edge of the A/D sample clock places all SSH sample/hold amplifiers in hold. The end of the last conversion returns to sample mode.

| | |
|---|---|
| Auto Increment (Bit 3) | 1 = Advance channel address at EOC or start<br>0 = No increment (single channel) |

Channel addressing in increment mode will sequence around from channel 0-15 to 0, modulo 16. Since the 4-channel modules use only channel address bit 1 and 0, this has the effect of addressing channels 0-3 to 0, modulo 4. Note that the address sequences at EOC or just after the A/D start to immediately begin settling on the next channel. If you write your own assembly code, observe settling time minimums before starting the next conversion.

| | |
|---|---|
| Interrupt Select (Bit 4) | 1 = DSP INT1 is generated by the FIFO full or half full flags (jumper selectable).<br>0 = DSP INT1 is generated by the ACQuire status output (82C54 Timer 0 count reached). |

Important: The FIFO half full mode (command bit 4 = 1 and jumper E3) is the normal mode for most A/D sampling. It assures no loss of A/D samples by reading the FIFO before it overflows. See the DMA_INTRPT function. Both bit 4 and DMA_INTRPT must be enabled to accept FIFO interrupts.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

DSP interrupts are enabled by the internal CPU registers.

| | |
|---|---|
| FIFO Reset | 1 = Reset FIFO, set empty flag, clear half full and full flags. |
| (Bit 5) | 0 = No FIFO reset (normal mode to accept A/D data). |

Normally, bit 5 is toggled 0, 1, 0. All previous data will be lost when the FIFO is reset. If the A/D is still running, the FIFO will not be empty at the next EOC.d

| | |
|---|---|
| LED Lamp | 1 = Lit (front panel lamp) |
| (Bit 7) | 0 = Dark |

Bits 31 - 8 are not used.

### 4.6 Start Channel Address Register (Write only to 804005h)

| D31 – D4 | D3 D2 D1 D0 |
|---|---|
| Not used | Channel Address |

This register controls the analog channel multiplexer address. For 4-channel models, the address range is 0 through 3. For 8- or 16-channel models the address range is 0 - 7 or 0 - 15. Allow sufficient settling time after selecting a new address before starting A/D conversion.

In autoincrement mode (command 3 = 1), this address will advance just before or after each A/D conversion and eventually cycle around through channel zero. The current address will be a modulo count of A/D starts since the channel address register was loaded. Only bits 1 and 0 are used for the 4-channel analog modules with an address wrap around at channel 3. All 4 counter bits are used for the 16-channel DVME-630E with an address wrap at channel 15. Note: On model -630F, dual channel SSH is selected by writing 2 into the Channel Address Register.

### 4.7 FIFO Data Register (Read only at 804007h)

| 31 – 16 | 15 | 14 | 13 | 12 | 11 | ---- | 0 |
|---|---|---|---|---|---|---|---|
| Not used | S | S | S | S | MSB | 12-bit A/D data | LSB |
| | S | S | MSB | | | 14-bit A/D data | LSB |

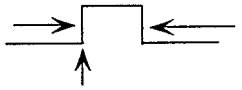
When the FIFO Empty Flag (EF) is not true, this register contains right-justified A/D data. Each read of the FIFO register fetches the next A/D sample from internal first-in, first-out (FIFO) memory with the oldest samples delivered first. The FIFO may be simultaneously filled by the A/D and emptied by the DSP. Data will not be lost if the 3 flags are observed and the aggregate rate of FIFO reads is faster than the A/D sample rate. If the FIFO is read by the DSP while the Empty Flag (EF) bit is set, the data field will be all one's. If the FIFO is not read when the Full Flag (FF) is set, further A/D samples written to the FIFO will be lost. The FIFO Half Full (HF) flag is set when at least half the FIFO contains data. Starting from the HF signal, count the number of samples read to prevent reading down past empty. The HF bit is set at one half the FIFO plus one (either 513 or 2049 samples).

For bipolar input ranges, "S" bits are sign-extended from either bit 11 (12 bit A/D's) or bit 13 (14-bit A/D's). For unipolar ranges, S = 0. Note that the SIGN_EXTEND function is required for full 32-bit sign extension. For bipolar ranges, the MSB is used to indicate polarity, where S = 1 for negative inputs with two's complement coding.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

## 4.8 Miscellaneous DSP Register Bits

The following signals are available as bits in various DSP internal registers. The trailing asterisk (*) means low true. Note that some INTx interrupt bits may be programmed as interrupts or as status flags. Many of these bits may be programmed as either inputs or outputs. Most require CPU register instructions for access.

| Name | Function | Direction | DSP bit |
|---|---|---|---|
| EXT INT* | External interrupt to DSP from connector P1-45 | Input | INT3 |
| EF* | FIFO empty flag interrupt to DSP. 0 = empty, 1 = not empty | Input | INT2 |
| FF/EOS INT (or half full) | DSP interrupt set by the command register and jumpers E2, E3. If cmd 4 = 0, the interrupt is from the ACQuire flag (timer 0), also called End of Scan (EOS). If cmd 4 = 1, the interrupt is from FF* (E2) or HF* (E3-default).<br><br>Important: The FIFO HF bit is set at 513 or 2049 samples for 1K or 4K FIFO's respectively (half full +1). | Input<br><br>(The interrupt occurs on the falling edge of HF*) | INT1 |
| DSPINTREQ* | DSP interrupt set by the VME host I/O cmd 5 = 1. | Input | INT0 |
| XF0 | Oversampling flag set if the A/D is retriggered before EOC is true. XF0 is reset by disabling the A/D. (Write 804004h bit 1 = 0.) | Input | XF0 |
| XF1 | ACQuire flag which is set when the timer 0 A/D sample count is reached. ACQ is true while sampling. A/D conversion will stop when ACQ goes false. | Input | XF1 |
| BINTREQ | Interrupt request to the VME host gated through the interrupt enable register and VME bus IRQ lines. | Output | TCLK0 |
| TCLK0 |  250 nS to 1 µS<br>VME bus interrupt occurs on rising edge. | Output | |
| INTTRIG* | Internal timebase from the DSP timer 1 for local A/D scan triggers. | Output | TCLK1 |


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

# SECTION 5

# AFB PROGRAMMING

## 5.0 DVME-630 General Control Methods

There are several general methods to control the DVME-630. They are:

1. Use the Application Function Block (AFB) scheduler and built-in function library.

2. Add your own user-written functions to the existing library and call them from a downloaded AFB.

3. Write your own code (either in "C" and/or 320C30 assembly). Compile, link, and download it to the DPR then execute. Do not use the AFB Exec system.

Most of this manual will discuss method 1 which uses AFB's. It is the easiest to use, requiring no local programming. It retains most of the speed of the C30 DSP. Method 2 (which requires writing local code and partially rebuilding the system) should be considered if none of the built-in library functions exactly fit your application AND you still want AFB-style operation. Method 3 starts from scratch and offers the highest speed and control over the DVME-630. You will have to take care of everything - vectors, interrupts, stacks, memory allocation, etc.

## 5.1 Software Introduction

The DVME-630 operates as a high speed intelligent slave computer to the VME host. The DVME-630 is not a VME bus master and cannot control the bus. After initialization and program loading, the VME host starts the DVME-630 running, responds to data ready signals, and retrieves processed data blocks. These blocks then are sent to disk, memory or display or they are further processed. Together, the VME host and DVME-630 achieve true simultaneous coprocessing.

The DVME-630 Executive software system has been designed to optimize three competing objectives:

Easy to Use (no local programming)
Fast
Powerful (access to full DSP library)

To achieve these mutually exclusive goals, a high speed command list form of control is used. The Application Function Block (AFB) is a short list calling local library functions. No local programming is needed. The user writes the AFB file with any text editor and it is then converted on the VME host side to an internal binary form. The converted AFB is then downloaded to DVME-630 Dual Port RAM (DPR) and executed. The AFB is powerful because of full access to the local DSP library and because repeating functions may be looped. These loops in turn may be nested. Loops can run with a loop count or "forever" until stopped.

Unlike a slow ASCII interpreter, the AFB runs at the full speed of the 320C30 DSP with minimal overhead. And to accept fast A/D data without sample loss, the DVME-630 uses a fully integrated hardware/software system to handle the bandwidth. This hardware system consists of local FIFO A/D memory, local FIFO interrupts, and a local Direct Memory Access (DMA) controller inside the DSP which runs in background. FIFO interrupts cause program data block transfers.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

### 5.2 Executive Package

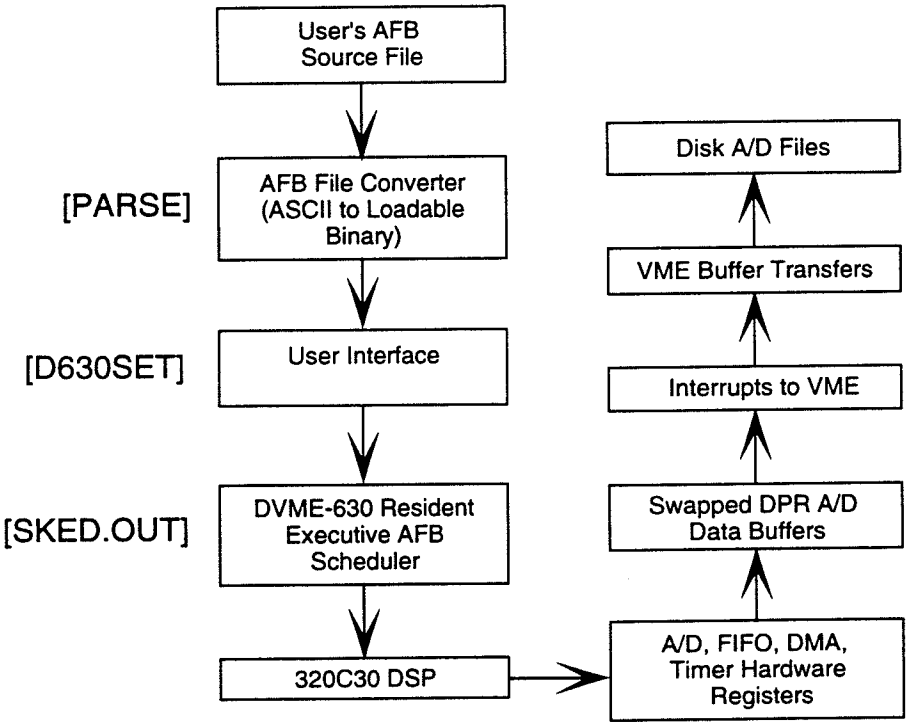
The complete Executive software package is an integrated environment for full control of the DVME-630. It includes programs which run on both the VME host and the DVME-630. The Exec package consists of:

The AFB ASCII to binary file converter (PARSE).
A menu shell.
The VME Host/User Interface . } (D630SET)
The Executive scheduler, DSP library, boot code, vectors, and full local DVME-630 system. (SKED.OUT)

To use this environment, the user simply converts the AFB text file to a binary file, downloads this to the DVME-630 through the EXEC and retrieves data files. These operations are largely automated through the resident library and the menu shell.

### 5.3 Software Hierarchy

The relationship between software in the VME host and in the DVME-630 is illustrated in this diagram. Control flows downward from the user's AFB and A/D data flows upward.




### 5.4 Simultaneous Concurrent Operation and DPR Access

Read/write requests from either side of the DPR are arbitrated by high speed addressing logic. If either side is using the DPR and the opposite side requests access, the opposite side will wait briefly until the current word transfer is completed. The DVME-630 will delay the DTACK* RDY VME bus line (pin A10) briefly with each access if the DSP has control of the DPR. Simultaneous block transfer requests will result in asynchronous interleaving of DPR access by either side. The accesses are probabilistic meaning that one side may finish before the other. When the VME host has control, the DSP briefly executes a dynamic hold with each cycle during attempted simultaneous access.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Note that both the memory reference portion of an instruction and the actual instruction opcode fetch is considered a DPR access. Many DSP instructions complete the opcode fetch and operand reference in one cycle. The DSP is making DPR accesses even while it is looping on a polled DPR flag. To stop the DPR requests, either place the DSP in hold momentarily or make the DSP execute only from its internal instruction cache loop and only from non-DPR memory. This may offer slightly faster access from the VME host side.

If a very large block transfer is requested by the VME host and minimum VME bus time is required, write your program to put the DSP in hold during the block. The VME host may of course access other parts of non-DPR VME host memory at full bandwidth while the DSP is making DPR access. The DSP runs at full speed anytime there is no access attempt by the VME host.

### 5.5 Power Up Sequence

The VME host establishes hold mode on the DVME-630 so that DPR testing can be done by the VME host. With the DSP in reset, the VME host downloads the initial Executive system from a binary COFF (Common Object File Format) disk file called SKED.OUT. This file includes all vectors, startup code, the local DSP library, tables, and all system code. The VME host causes the DSP to transition out of hold and run mode is started. The DVME-630 performs selftest and flags success using a status word (EXEC_OK_ADDR, offset 4C2h). The VME host confirms that the startup succeeded.

The DVME-630 begins polling the AFB/User flag (PROG_FLAG), also at a fixed offset (4C0h) from the BASE.

If an AFB array is downloaded and the AFB/User flag is switched to AFB mode, the scheduler will attempt to run the sequence of instructions in the AFB list. If AFB processing causes a DSP lockup or undefined operation, the DSP may be reset by the VME host, the DPR rewritten and the process restarted. Using these techniques, extremely powerful, general purpose control of the DVME-630 is offered. In fact, a series of different AFB's may be conveniently stored on disk and downloaded when needed.

After successful processing of an AFB, the scheduler will return to polling the PROG_FLAG. This occurs after the last outermost master END flag is passed. A new AFB may be downloaded and started without having to completely reload the full Exec.

### 5.6 Dual Port Random Access Memory Mapping

The memory map shown is maintained by the Executive system. If some other control method is used for the DVME-630, the map shown does not apply. Note the use of 32-bit DSP addresses as described previously.

At power up the DSP begins execution in the vectors area. The Monitor/Debugger area is reserved for the optional Debugger package. The book-keeping area is for control and status. The full DSP library, A/D, timer buffer control routines are loaded into the library area. The AFB scheduler is also located here.

A very large portion of the memory map is available for data buffers. These may include raw input A/D buffers or floating point output buffers.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The boundaries of the buffers may be varied by adjusting control variables and rebuilding the system from the source code.

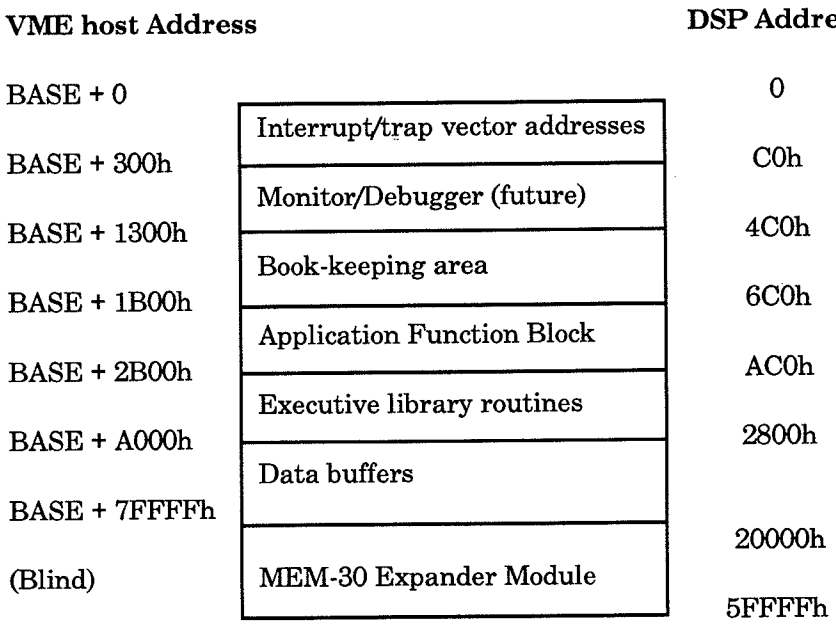
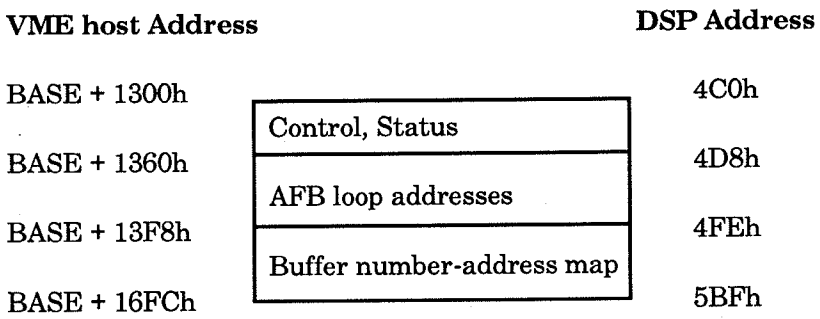



**Figure 5.0 Dual Port RAM Memory Map**

### 5.7 Book-keeping Area

This region controls AFB execution, buffer allocation and contains status flags. The sub-map is shown:





Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

### 5.8 Local Control and Status Flags

Note: ALL address offsets and variables listed in this section are 32-bit doubleword offsets from DSP location zero. To find the byte offset, multiply by four and add the BASE address in the VME host. All flags shown are 32-bit objects. Addresses and values are shown in hexadecimal. Variable names and labels correspond to those found in the source code.

Symbol: PROG_FLAG
Offset from BASE: 4C0 hex
Name: AFB/User program flag.
Description: The DVME-630 scheduler starts polling this flag after EXEC_OK_ADDR is set to -1. PROG_FLAG is initialized to zero. The VME host writes a value into this location to start an AFB or a block of User code, either of which must have been previously downloaded.
Values: AFB_EXEC = E111111E hex - run the AFB.
MAIN_EXEC = 1EEEEEE1 hex - run the User program (presently not used).

Symbol: EXEC_ERR_ADDR (This flag is set by the DEFxBUF functions)
Offset from BASE: 4C1 hex
Name: Executive error flag.
Description: The DVME-630 scheduler sets this flag when it detects a procedure error. The causes may include, but are not limited to, attempted used of undeclared buffers and function codes not found. Note that a DSP lockup will not necessarily set this flag. The VME host should test this flag with every buffer transfer.
Values: FFFFFFFF hex = error detected
FFFFFFFDh = insufficient buffers left
FFFFFFFEh = no memory left

Symbol: EXEC_OK_ADDR
Offset from BASE: 4C2 hex
Name: Executive okay flag.
Description: This flag is written by the DVME-630 scheduler after successful initialization. The VME host should wait one second after starting the EXEC before testing this flag. Do not run an AFB if this test fails. Do not access the DPR (including polling this flag) during the self-test. After setting EXEC_OK_ADDR to -1, the scheduler polls PROG_FLAG.
Value: FFFFFFFF hex = initialization succeeded.

Symbol: NUM_FLAG
Offset from BASE: 4C3 hex
Name: AFB Go/Abort flag
Description: This is set by the VME host and tested by the DVME-630 scheduler with each pass through the master loop. This flag tells the DVME-630 to process or abort the AFB. It can be set at any time. The flag defaults to one (Go) at initial Exec download. If the AFB is stopped, the Exec should be reloaded and started again.
Value: 1 = Enable AFB processing.
0 = Disable AFB processing.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Symbol: BUFF_SIZE_FLAG
Offset from BASE: 4C4 hex
Name: Block buffer size flag
Description: This location contains the half full size of the installed A/D FIFO memory devices. The possible values are from 512 to 8192 samples. The DSP reads this flag to determine the count to be loaded into the DSP counter for each FIFO ready block transfer. The VME host must initialize this flag for values other than 512 samples (the default) before a FIFO interrupt is sent to the DSP. Make sure the proper value matches the installed FIFO's. The SET_IBUF input buffers fed by the FIFO interrupt must be an integer multiple of this flag.
Value: 512 - default or set by VME host.

Symbol: BUF_RDY_FLAG
Offset from BASE: 4C8 hex
Name: Buffer ready flag
Description: This flag address is used to indicate that the data buffer is ready to be transferred and the handshake acknowledgement that the buffer was transferred. This flag is loaded by the DVME-630 SET_OBUF function with the BUF_FULL flag when that buffer is ready to be transferred to the VME host. After the buffer transfer, the VME host must write BUF_EMPTY into this flag. If the DVME-630 attempts to start filling a new buffer and does not detect the BUF_EMPTY acknowledgement in this buffer ready flag, the scheduler sets the buffer over-flow error flag.
Value: -1L (FFFFFFFF hex) = BUF_EMPTY, buffer is empty.
+1L (00000001 hex) = BUF_FULL, buffer is full.

Symbol: BUF_ADDR
Offset from BASE: 4C9 hex
Name: Buffer Address
Description: This location contains the doubleword address offset of the data buffer to be uploaded by the VME host. The SET_OBUF routine writes this address into BUF_ADDRESS just before sending the buffer ready interrupt to the VME host.

Symbol: BUF_COUNT
Offset from BASE: 4CA hex
Name: Buffer count
Description: This location gives the length of the buffer to be uploaded by the VME host. The SET_OBUF routine writes this length into BUF_COUNT just before sending the buffer ready interrupt to the VME host.

Symbol: AFB_DONE_FLAG
Offset from BASE: 4CB hex
Name: AFB done flag
Description: AFB_DONE_FLAG is set to zero before running AFB's. This flag is posted by the scheduler to indicate that the AFB has completed processing. Use a second final "END" statement to be sure AFB_DONE_FLAG stays set at +1.
Value: 1 = The AFB is done.
0 = The AFB is not done.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Symbol: PAUSE_FLAG
Offset from BASE: 4CC hex
Name: Pause flag
Description: This flag is set to +1 by the PAUSE routine. The DSP will then poll the flag, waiting for the VME host to reset it to zero. When 0 is received, the DSP resumes AFB execution. If you wish to be notified that PAUSE occurred, load a -1 (0xFFFFFFFF) from the host side before running the AFB.
Value: +1 = written by the DSP upon entering PAUSE.
0 = written by the VME host to terminate the PAUSE and resume execution.

Symbol: VERSION
Offset from BASE: 4CDh
Description: This is the revision level of SKED.OUT. This flag is not valid until EXEC_OK_ADDR is true (-1). Use the XDUMP utility to view this flag (xdump 4cd).

Symbol: INTERRUPT_FLAG
Offset from BASE: 4CF hex
Name: Interrupt enable flag
Description: The VME host sets this flag to allow interrupts from the DVME-630 scheduler. This is a software flag only and an interrupt enabled by this flag may be overridden by the command register. Note that SET_OBUF and INT2PC host are the only functions presently sending interrupts. Downloaded user programs may send an interrupt for any purpose and are not affected by this flag.
Value: 1 = Interrupt enabled.
0 = Interrupt disabled.

Symbol: INPUT_OVERFLOW_FLAG
Offset from BASE: 4D0 hex
Name: Input buffer overflow flag
Description: This flag is set by the DVME-630 scheduler for local block transfers started by A/D FIFO half full local interrupts. If the AFB loop does not transfer data out of the input buffer fast enough, this flag will be set. The IBUF_RELEASE function indicates that an input buffer has been transferred and is available for the next block transfer from the FIFO.

The DVME-630 scheduler will attempt to continue using the buffer even if an overflow condition is detected. The VME host must reset this flag in order to detect it the next time. Or the VME host may restart the Exec.
Value: 1 = Overflow occurred.
0 = No overflow error.

Symbol: OUTPUT_OVERFLOW_FLAG
Offset from BASE: 4D1 hex
Name: Output buffer overflow flag
Description: This flag is set by the DVME-630 scheduler if the VME host does not transfer the output buffer fast enough. After each output buffer block transfer, the VME host must reset the BUF_RDY_FLAG to -1. The SET_OBUF function tests the BUF_RDY_FLAG before filling the buffer. If BUF_RDY_FLAG does not equal -1, the OUTPUT_OVERFLOW_FLAG is set.
Value: 1 = Overflow occurred.
0 = No overflow error.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Symbol: OVERSAMPLE_FLAG
Offset from BASE: 4D2h
Name: Oversample flag
Description: This flag echoes the state of the A/D oversample flag at DSP bit XF0. This indicates that another start conversion command occurred while the A/D was still converting. Data was lost. XF0 may be cleared by disabling the A/D using SCOMMREG. OVERSAMPLE_FLAG is linked only by software and is typically posted once with each buffer fill.
Value: +1 = oversample occurred
0 = no oversample error

Symbol: POST_TRIG
Offset from BASE: 4D3h
Name: CIRBUF trigger sample
Description: See the CIRBUF function. This location stores the absolute address (not the offset) where the trigger sample occurred. The offset equals the contents of POST_TRIG minus the start of the buffer, which is available from the buffer table at 4FEh.

### 5.9 Local Pointers and Buffer Addresses

Symbol: TOP_OF_AFB
Offset from BASE: 6C0 hex
Description: The AFB starts here and extends to 0xABF hex.

Symbol: OFFSET_TABLE
Offset from BASE: 4FE hex
Description: A map of allocated buffer address offsets and their lengths starts here. Lengths follow each buffer. The map format is: BUF0, LEN0, BUF1, LEN1, BUF2, LEN2,. . . The map is reallocated when the Exec system is reset.

Symbol: END_OFFSET_TABLE
Offset from BASE: 5BF hex
Description: The buffer address offset and length map ends here.

Symbol: AFB_PTR
Offset from BASE: 4C5 hex
Description: This address contains the address of the start of the AFB.

### 5.10 User Extended Library (presently not supported)

This section of the DPR is available to download user-written programs written either in "C" or 320C30 assembly. Once the Exec is started, these "User" programs may be called by switching the AFB/User flag to the User state. User programs are executed by calling the main program with the instruction:

```
CALL _main
```

When control returns to the calling program, the system will loop forever until reset.

```
forever:
    br forever
```

If the User program accesses functions in the Executive library, the User code must use the calling protocol in the library code and both systems must be linked together to resolve labels. User code is an alternative to rebuilding the entire system.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

## 5.11 Application Function Block Structure

The AFB is loaded at a fixed offset from the bottom (BASE) address of the DPR. The AFB is executed sequentially up from the start of the AFB in ascending memory. (This goes down the page in AFB source listings.) The maximum size of the AFB is 1024 32-bit words long. The AFB consists of a list of subroutine tokens with each token followed by the number of parameters required then a list of input parameters to that subroutine. The tokens refer to a table of pointer addresses to the actual subroutines in the local downloaded library. The scheduler executes the subroutines from the pointer table addresses. All items must be in 32-bit format, even for very small integers.

To retain the highest speed, the scheduler does no error checking. The subroutine token, the number of parameters and the parameter limits (if any) must be exact for execution to succeed. The user is responsible for writing the AFB in a logically correct sequence. A few functions will return an Executive error flag if certain preconditions are not met (for example, using an array before buffers are allocated).

The AFB may have two sections. An initial portion executes only once. This portion is usually reserved for buffer declarations, timer setup, channel address initialization, etc. The next section is an outer master loop, typically used for A/D sampling and DSP math. The list of functions is executed sequentially then they are repeated back to the beginning of the loop. Inner loops may also be nested (up to 16 loops deep) within the outer master loop. The loops are marked with BEGIN and END 32-bit tokens with the BEGIN token followed by a loop COUNT variable. A loop may be run once, N times, or "forever" until stopped by the VME host. The loops are used for repetitive operations such as A/D sampling, FFT's, DSP math, or matrix operations. After the AFB finishes, it sets an AFB_DONE flag.

The AFB sequence forms a very simple procedural language. Although there is no conditional branching, subroutines or labels, these features are available by writing your own AFB functions. In fact, AFB sequencing may be combined with user-written code by writing your subroutines in the calling protocol used by the AFB scheduler.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Here is the structure of an AFB. Each line is a 32-bit value. The indentation and comments on the right side are shown only for process clarity and are ignored by the file converter. See Section 9 for more details:

```
Function Code                                          ;start of AFB
  Number of parameters = "N"
  Parameter 1
  Parameter 2
                                                       
  Parameter N                                          ;toward higher memory
Function Code
  Number of parameters
  Parameter list
Function Code
  Number of parameters
  Parameter list                                       ;end of single pass section
BEGIN Marker ( = 0FFFFFF0 hex )                        ;start of master loop
LOOP COUNT ( -1 = non-stop execution, "forever" )
  Function Code
      Number of parameters
      Parameter list
  Function Code
      Number of parameters
      Parameter list
  BEGIN Marker                                         ;start of nested loop
  LOOP COUNT
      Function Code
          Number of parameters
          Parameter list
  END Marker ( = F000000F hex )                        ;end of nested loop
END Marker                                             ;end of master loop
[ afb_done ]                                           ;AFB_DONE_FLAG posted
```

**Figure 5.1 AFB Structure**

### 5.12 AFB Markers

The following 32-bit values control the start and end of loops in an AFB.

Symbol: BEGIN_FLAG
Description: This marker defines the start of an AFB loop.
Value: F000000F hex

Symbol: END_FLAG
Description: This marker defines the end of a loop or the end of the whole AFB. There is no harm done using two final END flags. This will force AFB_DONE_FLAG to remain true before running the next AFB.
Value: 0FFFFFF0 hex


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

## 5.13 Double Buffering

Most DVME-630 data functions are organized around memory arrays called buffers. A buffer is a fixed-length area of memory with protected read and write operations controlled through the Executive. Access is referenced by a buffer number (a small integer) which uniquely defines a certain buffer. An internal table maintained by the system relates buffer numbers to absolute addresses. The actual memory address is unimportant. Buffers are allocated sequentially starting at buffer 0. Like all AFB parameters, the buffer numbers must be 32-bit objects. There is no dynamic buffer allocation and the Exec must be restarted to deallocate the buffers. (Dynamic buffering is available by writing your own code.)

A principal objective of the DVME-630 system is concurrent, high-speed non-stop A/D sampling, simultaneous DSP math processing and transfer to the VME host. These three asynchronous processes are periodically resynchronized by buffer transfer controls. Simultaneous operations are facilitated by double buffering methods. While the lower buffer is filling, the upper buffer is being emptied. Typically an input buffer is being filled with FFT data while an output buffer is being block-transferred to the VME host's display, disk or system memory. Software handshake flags are used to control buffer access by the input process. If the output process has not finished using the buffer, an overrun error flag is posted when the input process attempts access. The VME host polls these error flags with each buffer access.

In-place processing algorithms such as FFT's use a single swapped buffer. Two double buffers (an input pair and output pair - four buffers total) may be employed on the DVME-630 for non-inplace algorithms. The input pair collect raw A/D data from the FIFO memory. The FIFO half full local interrupt triggers the DSP to burst half the FIFO to a lower input buffer. Each FIFO burst advances an internal pointer address automatically until the lower input buffer is filled. Then the buffer is automatically switched to the upper input buffer.

While all this is occurring, the raw A/D data is being further processed by the master AFB loop using a pair of output buffers. A SWITCH_BUFFERS function in the loop uses the alternate output buffer. When a buffer is filled, the DVME-630 notifies the VME host with an interrupt. The VME host then examines DVME-630 flag locations (BUF_ADDR) which contains the BASE offset address and length (BUF_COUNT) of the filled buffer to be transferred. Immediately after reading the buffer, the VME host resets the BUF_RDY_FLAG to -1 (data taken), indicating that the buffer may now be used by the DVME-630. The SET_OBUF function tests if the previous outbuffer was taken by the VME host, posts the current buffer address to be read then interrupts the VME host. In this way, a swapped pair of buffers or even a sequence of buffers can be handled. Failure by the VME host to write the -1 acknowledge value will cause an overrun error in the OUTPUT_OVERFLOW_FLAG.

A fundamental concept to avoid loss of data is that the downstream processes must run faster in aggregate than the upstream processes. This concurs with actual operation where block transfers to the VME host are generally faster than DVME-630 FFT's on A/D data blocks.

## 5.14 System Speed Hierarchy

There are many trade-offs when determining total system speed. The system may be run in continuous non-stop mode, may fill one large buffer then stop or may run in repeating blocks with no A/D triggers in between.

With increasing non-stop A/D trigger rates, certain simultaneous operations become progressively less possible. Either the VME host will lag behind or the DSP will be too slow, both indicated by overflow error flags. These flags must be polled with each buffer transfer. With extremely fast triggering, the A/D hardware oversample flag should be tested periodically.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

At a certain A/D trigger rate, complex DSP math operations cannot keep up with the A/D section and buffer errors will be posted. Certain large DSP functions (e.g., a 4K-point FFT) may be slower than the transfer to a fast hard disk. At other times, the DSP will be faster than a slow VME host disk.

The entire system can be run at higher speed if no DSP math is done, only transfers to the VME host. With increasing speeds, simultaneous transfers to only a VME host buffer (not to disk) are still available. At even higher speeds, A/D data can be collected to the DPR and there is not enough time for concurrent VME host block transfers. Data must be saved to disk after A/D sampling stops. At the very top of the speed range, only one FIFO fill can be taken at a time to insure no lost A/D samples.

As speed increases, the total continuously-triggered sample array size is limited by the data destination. At moderate trigger rates and DSP math usage, hard disk storage will keep pace with non-stop triggering. At higher speed, VME host memory can keep up. For the highest speed, only the DPR and finally only the FIFO is available without dropping samples.

A final concept to understand is that total aggregate speeds are what matter, not specific A/D trigger rates, if the A/D is not continuously triggered. For example, the A/D may be triggered at over 1 MHz to a fixed size block. Then the A/D is allowed to wait tens of milliseconds while DSP math is done. This usually allows time for concurrent disk storage.

These speeds and capacities are listed in Figure 5.2. Exponential notation is used. The speeds shown are only a rough indication and a specific application could be faster or slower. The user must test the application to guarantee the speed.

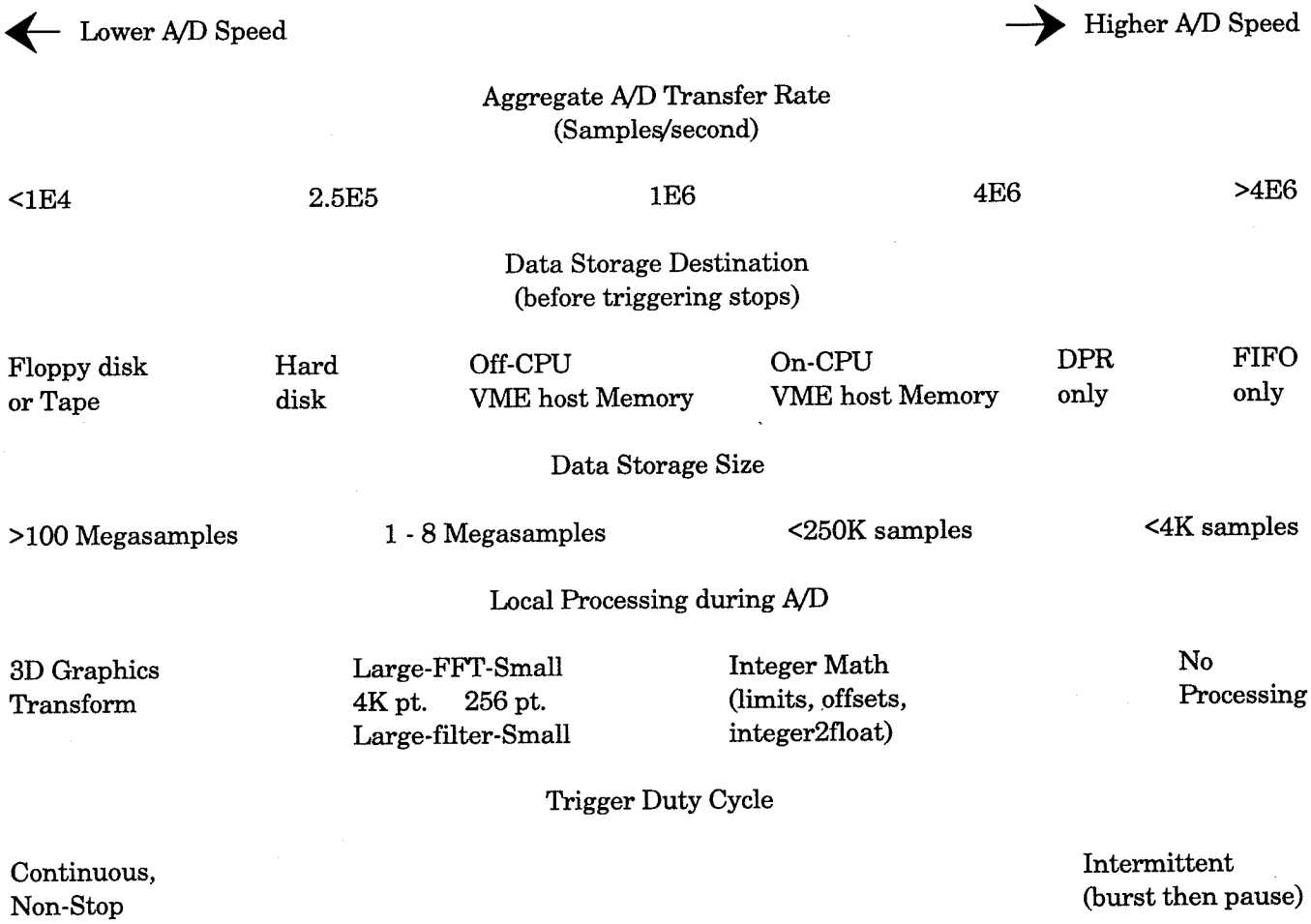



**Figure 5.2 A/D Trigger Rates Versus System Parameters**


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

If you are writing your own local assembly code, test the A/D oversample flag (XF0) for very high sample rates.

### 5.15 System Stack

The stack is located in internal memory. This allows the C30 to access both primary memory and the stack simultaneously in a single cycle without any latency. The stack is 1024 32-bit words. User-written software may relocate the stack.

### 5.16 Buffer Size Considerations

Normally a single input buffer should be sized for the DSP operation required. For a 2048-point FFT, this will require at least 4K 32-bit words (16K bytes) for a combined real and imaginary array. A double switched buffer will be twice this size. However the A/D FIFO is either 1K or 4K 16-bit A/D samples. Therefore, the FIFO is block transferred at the FIFO half-full local interrupt in either 512-word or 2048-word blocks.

To create a 4K buffer from a 1K FIFO for example, eight 512-word transfers from the FIFO are required to fill one of the swapped input buffers. Therefore each FIFO buffer transfer does not cause the input buffers to switch. The switch only occurs when one buffer is full. The FIFO interrupt routines keep track of progressive buffer fills and advance the current fill pointer address while testing for buffer full status. Since the AFB allocates the buffers, the FIFO routines get their buffer base addresses and lengths after the AFB initializes the buffers.

When each A/D sample is written as a 32-bit word in the input buffer, the upper 16 bits are meaningless and cannot be used in DSP computations. The full 32-bit raw A/D word will usually be adjusted later by the SIGN_EXTEND or FLOAT_TRANSFER function to create a valid signed 32-bit integer array.

### 5.17 Cycle Times

For users writing their own local assembly code, the following cycle times should be observed:

Most reads or writes using primary memory (MEMSTRB active) - 1 H1 cycle.

Most reads or writes using expansion memory or the I/O registers or DMA to non-DSP memory (IOSTRB active) - 2 H1 cycles.

An H1 cycle is 62.5 nanoseconds at 32.000 MHz. See the TI documentation for details.

Note that the 8K x 32 Expansion Memory has its own 32-bit data and address buses allowing simultaneous parallel operations with primary memory or internal DSP RAM. This bus is shared with the I/O registers including the FIFO A/D data register. The highest speeds occur when all data is loaded into the internal DSP dual 1K x 32 RAM's and the instruction loop completely fits within the DSP's 64 x 32 execution cache memory.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

### 5.18 Highest A/D Speeds versus Memory

The following are methods to maximize system speed with user-written assembly code.

The DVME-630 may be used in either concurrent simultaneous or non-concurrent serial modes. Concurrent mode collects A/D samples while doing DSP math and arbitrating buffer uploads to the host. Non-concurrent serial mode collects all the A/D samples first, then stops A/D sampling and starts DSP math (optional) then flags the host to upload. Using swapped dual DPR buffers, concurrent mode can sample "forever" with no lost samples. Non-concurrent mode allows the highest A/D sample rates to fill the DPR since no additional time is spent on simultaneous DSP math.

To achieve the very highest speed filling the DPR, all A/D samples must be saved as 32-bit integers since there will not be time for a CONCAT type operation between samples. The DSP can only do 32-bit operations. This limits the effective capacity to 128K samples minus a small amount of space for the system code (most of which could be run out of the expansion memory). Make sure the VME host is making no access to the DPR (including flag polling) while the A/D is filling it. This avoids arbitration delays. Use a final interrupt to the host from TCKL0 to signal that the DPR is filled.

If additional time is allowed for an on-the-fly CONCAT-style function, almost 256K samples may be saved with special local assembly code (see the DMA_INTRPT function, mode 3. It runs over 2 MHz.). Using CONCAT under the AFB system will leave about 220K samples available. If A/D data is truncated to 8 bits, almost 512K samples may be saved. Special code will be needed to pack four 8-bit A/D samples per longword.

To determine if the DSP system is fast enough, set the A/D sample rate to a moderate speed. Test the FIFO half full flag (at DSP INT1) IMMEDIATELY after a half full block transfer. As long as the half full flag has not been set yet, the system is still faster than the A/D rate and no samples have been lost. Count the number of pollings of the half full flag before it is set again. Post this polling count out to an unused DPR location so the host can read it (at low duty cycle please to avoid arbitration delays). This diagnostic step will be removed for an actual high speed application.

Now repeat the test with successively faster A/D sample rates. As the count of HF flag pollings approaches zero, system capacity is reaching its limit. Eventually the FIFO HF will be set right after a transfer. Post an overflow flag to the DPR. Use an A/D start rate that is slower than this maximum rate.

At the very highest A/D speed, another trick to improve rates is to transfer more than half the FIFO at the half full signal. This somewhat improves overall efficiency by having larger transfer block sizes and less frequent half full polling intervals. Since the FIFO continues filling with new A/D samples during a block transfer at the half full signal, more than half the FIFO is available for transfer. However, this amount cannot be directly determined. If the FIFO is transferred down past empty, erroneous "data" (all 1's in bits 15 - 0) will be collected. The empty flag is set only for an instant until the very next A/D sample makes the FIFO non-empty. Therefore the empty flag cannot be reliably polled and should be detected as a local interrupt.

The size of the increased block must be determined empirically. Write a local Interrupt Service Routine to catch the FIFO empty flag on DSP INT2. This interrupt will be hit if the block size is too large to avoid reading past empty. Make this FIFO empty ISR report an error flag out to the DPR or an output bit. Now using a constant A/D clock rate, try successively larger FIFO block sizes while watching for the empty interrupt. Once the block size causes the empty interrupt, reduce the final block size by several samples to prevent reading empty data. Obviously this trick will work for only one A/D sample rate.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

If the block size approaches the total (full) FIFO size, this says that most of the speed of the DSP system is completely absorbed by FIFO transfers (as you would expect).

For the FIFO block transfers, program block moves are faster than local DSP DMA. Another technique to insure highest speeds is to make the looping control code to handle FIFO transfers all fit inside the DSP 64-word internal control cache. A cache "miss" forces the system to refill its instruction queque at the cost of some machine cycles.

All of these methods require assembly language downloaded local DSP code. New AFB functions need to be written to do this under the AFB system.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

# SECTION 6

# APPLICATION FUNCTION BLOCK LIBRARY

This section lists the functions available in an Application Function Block. The user writes the functions in an AFB list to perform A/D conversion, DSP math and buffering operations. All functions follow an identical form. The function code is first, followed by a variable indicating the number of parameters needed by that function. This number is used to push parameters on the stack. After the number of parameters, the input parameters are listed. To retain speed and small memory size, there is no error checking. The user must insure that the AFB is accurate and arranged in a logically correct sequence. The function code is lowest in memory with the parameters following in ascending memory.

The function code is a small integer which refers to a table of absolute subroutine addresses. The actual code number is available in a header file (FUNCDEF.H) in the source code and is also decoded from the function name by the PARSE utility. With future software revisions, the addresses may change but their function codes will remain the same. All items in the AFB list are 32-bit objects. Hexadecimal constant representation is used.

Note: Functions which write bits to the write-only command register (SADTSC, SFIFOI, SADR, RFIFO, and SCOMMREG), save those bits in an internal read/write shadow register to be used for the next command register write. The user should remember which bits were previously set. Otherwise rewrite the whole register using SCOMMREG. The command register and its shadow are both initialized to zero after start up.

Most registers are either 8 or 16 bits whereas all CPU operations are 32-bits. Register bits occupy bits 15-0 or 7-0 in the 32-bit word.

## 6.0 A/D Scan Functions

### 6.0.0 INITAD

Initialize the start channel address.

XXXXX = Function code
00000001h = Number of parameters
XXXXX = Start channel address

This function loads the input multiplexer start channel address register. The current channel address will then stay fixed at the starting channel or will increment, depending on the autoincrement bit in the command register.

### 6.0.1 INITTIM0

Note: For all the 82C54 counter/timer functions listed, please refer to Section 8.

Load 82C54 timer 0.

XXXXX = Function code
XXXXX = Timer 0 control register word
XXXXX = Timer 0 data register word in A/D samples per trigger

This routine initializes 82C54 timer/counter 0 to select the number of A/D samples per trigger. The minimum is three samples per trigger. The maximum number of samples is 65,536. The data value should not exceed 16-bits even though the parameter field is 32-bits.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

For greater than 64K samples per trigger (the "forever" mode), a special technique is used. Mode 4 on the 82C54 is loaded into the control word then the data register is NOT loaded. This is detected automatically in the function for control code 0x38L (mode 4). For mode 4, any value may be written in the data register. The total number of samples is then counted by the user's software.

Important: Load one less than the number of samples per trigger desired. For 1024 samples, load 0x000003FFL.

### 6.0.2 INITTIM1

Load 82C54 timer 1.

XXXXX = Function code
00000002h = Number of parameters
XXXXX = Timer 1 control register word
XXXXX = Timer 1 data register word in conversions per second.

This routine initializes 82C54 timer 1. The data parameter is the A/D conversion rate in samples per second. This timer supplies the A/D start convert clock. The timer divides either an internal 8 MHz clock or an external Pacer clock. The minimum divisor is two, giving a 4 MHz maximum internal start convert rate. The data value should not be less than 122 Hz (8000000/65535).

### 6.0.3 INITTIM2

Load 82C54 timer 2.

XXXXX = Function code
00000002h = Number of parameters
XXXXX = Timer 2 control register word
XXXXX = Timer 2 data register word in SSH samples per A/D clock

This routine initializes 82C54 timer 2. This timer is used only with the DVME-630A in Simultaneous Sample/Hold mode. The timer counts the number of A/D conversions from each start pulse from Timer 1 OUT1, giving multiple A/D conversions per A/D start pulse. Normally no more than four or sixteen samples (i.e., one scan) should be programmed per start clock. The data value should not exceed 16-bits even though the parameter field is 32-bits.

Since each start pulse initiates multiple A/D conversions, the start pulse rate, programmed by Timer 1, should not exceed the specified multichannel A/D sample, including all settling delays. For four channels, this should be 250 KHz or less.

Important: Load one less than the number of SSH samples per A/D clock desired. For four samples, load 0x3L.

Timer 2 should always be configured for mode 2 (control word = 0xB4 or 0xBC). The minimum count data is 2. Therefore, the minimum number of SSH channels acquired is 3 (one more than the count value).


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

### 6.0.4 SADTSC

Select internal or external trigger source.

XXXXX = Function code
00000001h = Number of parameters
XXXXX = 0 for 320C30 internal trigger, 1 for external trigger

This routine selects the data acquisition trigger source (not the A/D start clock) using the command register bit 0. The internal trigger is 320C30 timer 1.

### 6.0.5 SADCR

Select the A/D start clock rate.

XXXXX = Function code
00000001h = Number of parameters
XXXXX = Rate in samples per second (16 bits max.)

This routine sets the 82C54 timer 1 divisor which selects the A/D start clock rate. If an internal start clock timebase is used, the raw clock fed to the 82C54 timer 1 (before the divisor) is 8 MHz. If an external start clock timebase is used, SADCR sets the timer 1 divisor by the formula, divisor = 8 MHz/rate. The divisor may be written directly using INITTIM1.

### 6.0.6 SSTR

Select the internal trigger rate.

XXXXX = Function code
00000001h = Number of parameters
XXXXX = Rate in triggers per second

This function selects the internal scan trigger rate by loading the 320C30 timer 1 with the rate.

The AFB SSTR function has been modified slightly starting with SKED version 21 hex (33 decimal). This function sets the internal DSP trigger timer. The time interval is being changed to accomodate applications which need only one trigger (to start "forever" sampling) and for applications slower than 1 trigger per second.

IMPORTANT: The rate parameter is being revised to TI float format for this change. Previous versions used a 32-bit integer rate. Since the timer divisor register in the DSP is a 32-bit integer, this is simply a notation convenience. In the SSTR AFB function, the float is cast to integer just before loading the timer register. This means that the fundamental operation performed is either N Hertz or 1/N Hertz. The special condition of N = zero is trapped to yield one single trigger only for applications which sample "forever".

If you need a trigger every minute, enter the parameter as 0.01666667 seconds in TI float (1.666667E-2). The internal cast will load the proper integer divisor.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The SSTR AFB may be called as follows:

SSTR (1 parameter, 10.0) - 10 triggers per second (10 Hz)
SSTR (1 parameter, 0.0) - Exactly one trigger pulse only
SSTR (1 parameter, 0.1) - 1 trigger every 10 seconds (0.1 Hz)
SSTR (1 parameter, 0.0166667) - 1 trigger every 60 seconds
SSTR (1 parameter, 0.8) - Not recommended. The cast would round up to 1.0 (1 Hz).

Note that this is the DPR AFB binary interface to SSTR, not the DVME-630EXEC interface.

DVME-630EXEC Version of SSTR

The PARSE.C file in the DVME-630EXEC, was modified in a slightly different way to handle the integer divisor. Rates greater than 1 Hz use a trailing 'L'. Rates less than 1 Hz use a trailing 'F'. Notice that ONLY integer parameters are accepted. They are internally converted to the float format before building the binary AFB.

Write the EXEC AFB source file as follows:

SSTR (10L) - 10 triggers per second (10 Hz)
SSTR (0) - Exactly one trigger pulse only
SSTR (10F) - 1 trigger every 10 seconds (0.1 Hz)
SSTR (60F) - 1 trigger per minute
SSTR (0.5) - Illegal. Parameters must be decimal integers.

### 6.0.7 SADSPC

Select the number of A/D samples per scan.

XXXXX = Function code
00000001h = Number of parameters
XXXXX = Number of A/D samples per scan

In the scan mode only (command bit 2 = 1), this function sets the number of A/D samples to be collected with each trigger. This mode is used primarily for simultaneous sample and hold to automatically scan all held SSH channels. SADSPC loads the 82C54 counter 2 with the count. If more than one sample is requested, SADSPC also sets the autoincrement bit 3 in the command register.

### 6.0.8 STADS

Select the total number of A/D samples.

XXXXX = Function code
00000001h = Number of parameters
XXXXX = Number of A/D samples per session - 1

This function sets the total number of A/D samples to be acquired during a session up to a maximum of 65536 samples per session. 82C54 counter 0 is loaded with the sample count. Load one less than the number of samples required with 3 samples minimum. When the count is reached, A/D conversion will be disabled. For non-stop sessions in excess of 65K samples, set 82C54 counter 0 to mode 4 then do not write to the data register.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

### 6.0.9 SFIFOI

Enable/disable the FIFO interrupt.

XXXXX = Function code
00000001h = Number of parameters
XXXXX = 0 for disable, 1 for enable

This function enables or disables the FIFO interrupt from either the half full or full flag (depending on jumpers). Bit 4 in the command register is set or cleared.

### 6.0.10 SADR

Enable/disable A/D conversion

XXXXX = Function code
00000001h = Number of parameters
XXXXX = 0 for disable, 1 for enable

This function sets or clears bit 1 in the command register. Normally this should be the last operation after setting up the timers and channel addressing.

### 6.0.11 RFIFO

Reset the FIFO memory

XXXXX = Function code
00000000h = Number of parameters

This function flushes the FIFO memory by resetting bit 5 in the command register from 1 to 0. All previous A/D data will be lost and the empty flag will be true. If A/D conversion is still running, the FIFO will start filling again at the next A/D EOC signal. SCOMMREG must have been invoked first before running RFIFO to initialize the internal LAST_COM_REG_VAL.

### 6.0.12 CALAD

Calibrate the A/D converter

XXXXX = Function code
00000002h = Number of parameters
0X0L = Not used (set to zero)
XXXXX = Simultaneous sample/hold flag
0x1L = SSH (DVME-630A)
0x0L = No SSH (DVME-630B-F)

This function reads four A/D samples (channels 0 - 3) from the FIFO and places them in the bottom (offset 2800h) of the buffer area. The function then sets the BUF_RDY_FLAG to BUF_FULL (+1L) in the DPR. The VME host reads the A/D samples and acknowledges by resetting BUF_RDY_FLAG to BUF_EMPTY (FFFFFFFFh or -1L). CALAD polls the BUF_RDY_FLAG until the -1 acknowledgement then fetches the next samples. CALAD will poll for -1 in BUF_RDY_FLAG before the first sample is fetched. Therefore the VME host must set the -1 to start the sampling. The FIFO is reset between each loop of the function and only the first samples in the FIFO are written to the buffer. CALAD will overwrite any previous data at buffer addresses 2800h - 2803h. Buffers do not have to be allocated. If NUM_FLAG (offset 4C3h) is set to zero by the host, CALAD returns and the scheduler starts polling PROG_FLAG for the next AFB. CALAD manages all timers and triggers.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

### 6.0.13 SCOMMREG

Set the A/D command register (see Section 4.6)

XXXXX = Function code
00000001h = Number of parameters
XXXXX = A/D command register word

This function writes the specified word into the command register at 804004h and also copies into the shadow register ("LAST_COM_REG_VAL"). All previous bits are overwritten. Do not inadvertently write bits incorrectly.

### 6.0.14 DMA_INTRPT

Enable FIFO interrupts, select FIFO block transfer mode, perform optional data packing or sign extension.

XXXXX = Function code
00000001h = Number of parameters
XXXXX = Transfer mode
  0 = Program transfer (repeat block - 1 A/D sample per longword)
  +1 = Local DMA block transfer
  +2 = FIFO empty flag collection (reserved for internal usage)
  +3 = Data packing transfer (2 A/D samples per longword)
  +4 = Sign extension transfer (1 A/D sample per longword)

(Modes 3 and 4 are new with SKED.OUT revision 2.5)

This function enables the local interrupt from the FIFO half full signal. Depending on the mode, one half the FIFO of A/D data is moved from the FIFO data register to the SET_IBUF input buffer at each HF interrupt. Command bit 4 also must be enabled. The FIFO BUFF_SIZE_FLAG at 4C4h must be set properly before the transfer. Make sure the buffers are allocated as an integer multiple of half the FIFO.

Each buffer transfer will advance the address pointer in the current input buffer, using multiple transfers if necessary to fill the buffer.

When DMA_INTRPT is executed, the FIFO interrupt system is "armed". DMA_INTRPT must be called once before the A/D start. The A/D should be enabled in the command register after calling DMA_INTRPT.

Transfer modes:

All modes except number one use a program transfer (repeat block) since this is higher speed than DMA (mode one).

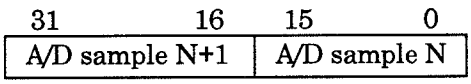
Mode 0: This mode uses program mode for highest speed. The data size is 32-bit A/D samples with hardware sign extension to bit 15. A/D data is right justified.

Mode 1: Identical to mode 0 but uses local DMA.

Mode 2: Reserved for local use by FIR (don't use externally).


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Mode 3: Two successive A/D samples are packed into one 32-bit word on the fly. Bits 31-16 are the later A/D sample and bits 15- 0 are the earlier sample. Both samples include hardware sign extension. This mode is faster than the CONCAT function and uses less memory. The format is:

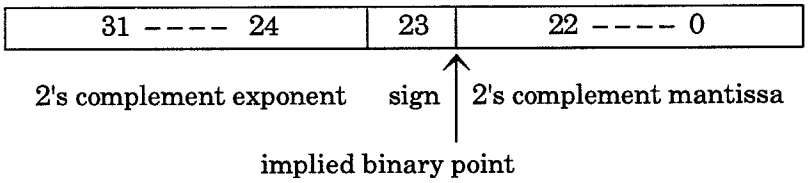



Mode 4: This mode performs on-the-fly sign extension of bit 15 through bit 31. One A/D sample is stored per 32-bit longword. Program transfer is used.

### 6.1 Data Formats

The 320C30 DSP addresses only 32-bit objects. There are no 16-bit or 8-bit modes. In the DSP functions listed below, unless otherwise specified as 32-bit integers, all input data arrays must be in the TI 32-bit single precision floating point format. This format is optimized for DSP hardware speed. For A/D raw binary data, the conversion is performed using the SIGN_EXTEND and INT2FLT or the FLOAT_TRANSFER array functions. Output arrays will also be in this form and must use the DSPIEEE function to convert to IEEE-754 format. Refer to the TI manuals for further details.

The single precision floating point format is reproduced here:




The floating point number x is given by:

x = 01.mantissa * 2^exponent if sign = 0
x = 10.mantissa * 2^exponent if sign = 1
x = 0 if the exponent = -128, sign = 0 and mantissa = 0.

### 6.2 FFT Array Formats

The DVME-630 Executive uses two different formats for real or complex Fast Fourier Transform arrays. Raw A/D data must first be sign-extended and converted to floating point. If the complex FFT is used, the imaginary array is filled with floating point zeroes to generate a DC-referenced FFT output. Both formats require allocation of a 2*N array to process an FFT from N real A/D samples. Both the number of points and the log 2 of the points are passed as parameters to avoid another calculation.

The complex FFT format deletes the redundant Im(0) and Im(N/2) terms since they are present in the Re(0) and Re(N/2) terms respectively.

The format following shows the bipolar 12-bit A/D data before full 32-bit sign extension and floating point conversion. For highest speed (and because the DSP does not have a 16-bit transfer mode), A/D data is always read from the FIFO as a 32-bit longword. Note that sequential 4-channel data (even using the SSH option) will appear in the channel sequence . . . 0, 1, 2, 3, 0, 1 . . . (or . . . 0, 1, 2 . . . 14, 15, 0, 1 . . . for the 16-channel DVME-630E). Sequential multichannel arrays must be split up into single channel arrays by the UNRAV functions before most math processing.


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

If no floating point math is done (including the DSP functions), the raw A/D array in the DPR can be reduced to half its size by using the CONCAT function. The sign extension from bit 11 to 15 is done by A/D hardware. Sign extension from bit 15 to 31 is done by software (SIGN_EXTEND or FLOAT_TRANSFER).

Raw A/D array from FIFO memory (12-bit A/D is shown):

| Data sample | Bit 31 – 16 | 15 – 12 | 11 | 10 – 0 | Array index |
|---|---|---|---|---|---|
| 1 | Not used | sign-extension | A/D MSB (sign) | A/D data | 0 |
| 2 | Same as above | | | | 1 |
| ⋮ | | | | | ⋮ |
| N | Same as above | | | | N -1 |

After 32-bit sign-extension:

| Data sample | Bits 31 – 12 | 11 | 10 - 0 | Array index |
|---|---|---|---|---|
| 1 | sign-extension of MSB | A/D MSB (sign) | A/D data | 0 |
| 2 | Same as above | | | 1 |
| ⋮ | | | | ⋮ |
| N | Same as above | | | N -1 |

Real N-point FFT

(Input array, after A/D floating point conversion but before FFT):

| Data point | | Array index |
|---|---|---|
| 1 | Sign-extended, floating point A/D data | 0 |
| ⋮ | | ⋮ |
| N | | N-1 |
| ↓ higher memory | FFT scratchpad | N |
| | | ⋮ |
| | | 2*N -1 |


Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

**Artisan Technology Group** is an independent supplier of quality pre-owned equipment

## Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

## We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

## Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP