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 *Teledyne Dalsa / Coreco Imaging IMS-150* at our website: *Click HERE*

MVC 150/40

IMS

Software Manual

47-S54002-00

iMAGING Technology

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

IMS Software Manual

Document Number 47-S54002-00

Revised June 1994

Copyright© Imaging Technology Incorporated 1999–1994
All rights reserved.
Printed in the United States of America.

All copyrights in this manual, and the hardware and software described in it, are the exclusive property of Imaging Technology Incorporated and its licensors. Claim of copyright does not imply waiver of Imaging Technology's or its licensor's other rights in the work. See the following Notice of Proprietary Rights.

NOTICE OF PROPRIETARY RIGHTS

This manual and the related hardware and software are confidential trade secrets and the property of Imaging Technology and its licensors. Use, examination, reproduction, copying, transfer and/or disclosure to others of all or any part of this manual and the related documentation are prohibited except with the express written consent of Imaging Technology.

The information in this document is subject to change without notice. Imaging Technology Incorporated makes no representations or warranties with respect to the contents of this manual and specifically disclaims any implied warranties of merchantability or fitness for a particular purpose. Imaging Technology Incorporated assumes no responsibility for errors or omissions in this document.

Modular Vision Computer, MVC, MVC 15/40, ITEX, and the IMAGING logo are trademarks of Imaging Technology Incorporated.

All other trademarks are the property of their respective owners.

Imaging Technology Incorporated
55 Middlesex Turnpike
Bedford, MA 01730–1421
(781) 275–2700
FAX: (781) 275–9590

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

PREFACE

ITEX is the library of control functions for the MVC 150/40 image processing subsystem. This manual describes the IMS functions including calling sequences, the functions performed, and examples of code containing the calls.

- Chapter 1, “Introduction” contains general information about the IMS, including a comparison with the IMA, and a hardware overview of the IMS. This chapter also gives information about software conventions and restrictions. Chapter 1 concludes with tables listing the IMS functions in functional groups.
- Chapter 2, “IMS Functions” describes all the IMS functions in alphabetical order.
- Appendix A, “Register Tables” summarizes the IMS registers.

Note *The IMS functions are given in alphabetical order in Chapter 2 of this manual. A series of tables in chapter 1 list these functions by functional groups, and give their page references. If you are more used to functional groupings, consult Chapter 1.*

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

DOCUMENT CONVENTIONS

In this manual, the following conventions describe each function:

This paragraph describes the value returned with the symbol INQUIRE or with normal operation.

Error Values

This paragraph describes the values returned because of error conditions, most often incorrect parameter values.

This paragraph lists the registers or bitfields changed by a function. If a bitfield is changed, the register that contains the bitfield may be listed in parentheses. The hardware manual contains more information about registers and bits.

Constants defined in a header file are shown in uppercase; for example, ON. The header file names are enclosed in quotes; for example, “amclr.h.”

Program examples throughout this book use an ellipsis (...) to indicate that insignificant code has been removed from the example or the code fragment, for example:

```
...
BYTE pixarray[512];
...
  ima_rhbline(ADDRESS,200,1,pixarray);
...
```

AOI A hardware timing format, optimized for digital data processing. Faster than camera or display timing. Functions like *x*_**snap**, *x*_**grab** usually use camera timing. Functions like *x*_**aoi_snap**, *x*_**aoi_grab** use AOI timing, usually for board to board processing or transfer.

ROI A software data structure, used as a location (source or destination) for image data. An ROI can by any useable system resource: camera, display monitor, image memory, system memory, or system disk.

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

Table of Contents

Chapter 1. Introduction

Hardware Overview 1–1
Image Managers: IMA and IMS 1–2
Image Memory 1–4
Scan Access 1–5
Zoom Acquisition 1–6
Acquisition Control 1–7
Memory Write Masks 1–7
Interrupts 1–7
Cross–port Switch 1–8
Clock Speed 1–8
Frame Timing 1–9
AOI Timing Generator 1–9
Video Bus Interface 1–11
Module Interfaces 1–11
Acquisition Module 1–12
Computational Module 1–12
Display Module 1–12
Software Introduction 1–13
Argument Types 1–13
IMS Selection 1–14
Returned Values 1–14
Initialization Functions 1–15
Synchronization and Acquisition Functions 1–15
AOI Timing 1–15
ROI versus AOI 1–16
Read and Write Pixel Functions 1–18
Area Functions 1–19
Module Structure Functions 1–20

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

Chapter 2. IMS Functions

ims_acqbits — Set or Return the Acquisition Command Bits 2–2
ims_aoi_1shot — Perform One Clock Cycle 2–3
ims_aoi_grab — Acquire Multiple Images with the AOI Generator 2–4
ims_aoi_grab16 — Acquire Multiple 16–Bit Images with the AOI Generator 2–5
ims_aoi_halt — Stop an AOI Timing Generator 2–6
ims_aoi_snap — Acquire a Single Image with the AOI Generator 2–7
ims_aoi_snap16 —Acquire a Single 16–Bit Image with the AOI Generator 2–9
ims_aoi_start — Start an AOI Timing Generator 2–11
ims_bitmask — Set or Return the Bit Mask 2–12
ims_chbline — Clear a Horizontal Line to a BYTE Value . . . 2–13
ims_chdwline — Clear a Horizontal Line to a DWORD Value 2–14
ims_chwline — Clear a Horizontal Line to WORD Value . . . 2–15
ims_clkspeed — Set or Return the Clock Speed 2–16
ims_clr_area — Clear an Area 2–17
ims_clr_frame — Clear an Frame 2–18
ims_cp_area — Copy an Area 2–19
ims_cvbline — Clear a Vertical Line to a BYTE Value 2–21
ims_cvdwline — Clear a Vertical Line to a DWORD Value . . 2–22
ims_cvline — Clear a Vertical Line to a WORD Value 2–23
ims_dbrd_intr — Enable or Disable the Interrupt on a Daughter Board 2–24
ims_delay_acqen — Begin a Delayed Acquire 2–25
ims_dispmode — Set or Return the Display Mode 2–26
ims_dregs — Display the Register Values 2–27
ims_dump_mem — Display an Area 2–28
ims_dzoom — Zoom Frame B1 for Display 2–29
ims_dzoomxy — Set Frame B1 X and Y Zoom Factors for Display 2–30
ims_ext_intr — Enable or Disable External Trigger Interrupt 2–21
ims_frame_intr — Enable or Disable the Frame Interrupt . . . 2–32
ims_freeze — Stop Acquisition 2–34
ims_get_adrform — Return the Address Format 2–35
ims_get_aoi — Return the AOI Dimensions 2–36
ims_get_mapadr — Return the VME Memory Map Address 2–37
ims_get_mod — Return a Pointer to an IMS 2–38

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

ims_get_rdroam — Return the Read Pan and Scroll Shadow Registers 2–39
ims_get_slot0 — Return a Pointer to a DM 2–40
ims_get_slot1 — Return a Pointer to a CM 2–41
ims_get_slot2 — Return a Pointer to an AM 2–42
ims_get_wrroam — Return the Write Pan and Scroll Shadow Registers 2–43
ims_grab — Acquire Multiple Images 2–44
ims_ien_bits — Enable or Disable Interrupts 2–45
ims_intr_priority — Set or Return the Interupt Priority Level 2–46
ims_intr_vec — Set or Return the Interrupt Vector 2–47
ims_iregs — Initialize the Registers 2–48
ims_mapsize — Set or Return the Memory Map Size 2–49
ims_pixsz — Set or Return the Pixel Size 2–50
ims_psreset — Set the Pan and Scroll Registers 2–51
~~ims_read_image — Read an Image from a File~~
~~ims_read_imagearea — Read Part of an Image from a File~~
ims_rhbline — Read a Horizontal Line of BYTEs 2–52
ims_rhdwline — Read a Horizontal Line of DWORD Values 2–53
ims_rhwline — Read a Horizontal Line of WORD Values . . . 2–54
ims_rpix — Read a Single Pixel Value 2–55
ims_rvbline — Read a Vertical Line of BYTE Values 2–56
ims_rvdwline — Read a Vertical Line of DWORD Values . . 2–57
ims_rvwline — Read a Vertical Line of WORD Values 2–58
ims_sel_slave — Select or Return a Slave to Output to the Video Bus 2–59
ims_set_aoi — Set the AOI Dimensions 2–60
ims_set_cpspath — Set the Cross–port Switch 2–61
ims_set_lineio — Set–up Frame Buffer for Access 2–63
ims_set_mapformat — Set the Memory Access Format and the Memory Base Address 2–65
ims_set_rdroam — Set the Read Pan and Scroll Shadow Registers 2–66
ims_set_wrroam — Set the Write Pan and Scroll Shadow Registers 2–67
ims_snap — Acquire a Single Image 2–68
ims_vbusmode — Set the Video Bus Timing 2–69
ims_wait_notvb — Wait Until Vertical Blank is Past 2–70
ims_wait_vb — Wait for Next Vertical Blank 2–71
ims_waitacq — Wait for Acquisition to Complete 2–72
ims_whbline — Write a Horizontal Line of BYTEs 2–73

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

ims_whdwline — Write a Horizontal Line of DWORD Values 2–74
ims_whwline — Write a Horizontal Line of WORD Values . . 2–75
ims_wpix — Write a Single Pixel Value 2–76
~~ims_write_image — Write an Image to a File~~
ims_wvbline — Write a Vertical Line of BYTE Values 2–77
ims_wvdwline — Write a Vertical Line of DWORD Values . 2–78
ims_wvwline — Write a Vertical Line of WORD Values 2–79

Appendix A. Registers

Index

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

List of Tables

Number	Title	Page
1–1.	Error Messages	1–14
1–2.	Initialization Functions	1–15
1–3.	Acquisition Functions	1–17
1–4.	Read and Write Pixel Functions	1–18
1–5.	Read and Write Line Functions	1–19
1–6.	Area Functions	1–19
1–7.	Module Structure Functions	1–20
A–1.	IMS Symbolic Register Names	A–2

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

List of Figures

Number	Title	Page
1–1.	Video Bus Support	1–3
1–2.	IMS Block Diagram	1–4
1–3.	Scan Access Ports	1–5
1–4.	Sub–Sample and Zoom	1–6
1–5.	Example Path	1–11
A–1.	IMS Registers	A–5

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

CHAPTER 1
INTRODUCTION

The IMS software is a library of functions that control and operate Imaging Technology's IMS (standard image manager) mother board. The IMS software is distributed as an object code library that can be linked to application programs written in C.

Note *Please read the Software Release Notes provided before attempting any set–up or use of this package. For complete software installation information, see the Installation Guide, but examine the Release Notes for any exceptions to the installation guide.*

HARDWARE OVERVIEW

The standard MVC 150/40™ contains an IMS with an acquisition module (AM), a computation module (CM), and a display module (DM). The IMS contains frame memory, timing generators and two cross–port switches. The cross–port switches control the data paths between the IMS, the AM, the CM, and the DM.

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

Image Managers: IMA and IMS

The IMS is one of two mother boards available with the MVC 150/40. The IMS is a subset of the IMA. The following list compares and contrasts the two image managers and Table 1–1 summarizes this information.

- The IMA and IMS both support one AM, one CM, and one DM
- The IMA and IMS have the same video connector pin–out, and use the same video cables
- The IMA and IMS display connectors are different, and use different display cables. The IMA has a 15–pin VGA compatible display connector. The IMS has a 26–pin display connector that also contains the CM Serial port connections.
- The IMA has 4MB image memory and is organized as 4 frames of 1K by 1K by 8 bits. The IMS has 1.5MB or 3MB of image memory and is organized as 3 frames; of 1K by 512 by 8 bits, or 1K by 1K by 8 bits.
- The IMA frames can be reconfigured as 2K by 512, 4K by 128. The IMS frames cannot be reconfigured.
- The IMA frames A0 and A1 can be tiled together into one frame 1K by 2K, and reconfigured as 2K by 1K. The IMS frames cannot be tiled.
- Only one IMS is allowed in a subsystem. Up to 4 IMAs can be configured in one subsystem.
- The IMA can transmit and receive on the global bus, receive on the pipeline input bus, and output on the pipeline output bus. The IMS can only output data on the global bus, and receive data only on the pipeline input bus. The IMS does not have a pipeline output. Refer to Figure 1–1.

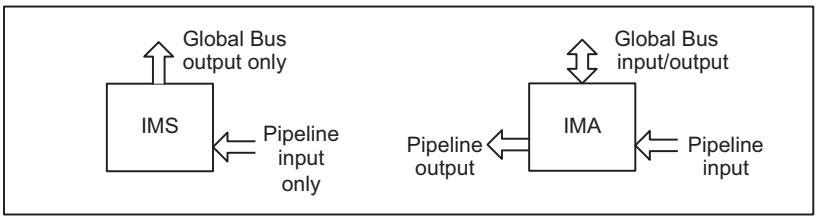


Figure 1–1. Video Bus Support

- The IMA accepts up to 24–bits from the AM. The IMS accepts up to 8–bits from the AM.

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

Table 1–1. Feature Comparison

Feature	IMA	IMS	comments
Memory			
size	4MB	3MB(1.5MB)	frame size: 1K by 1K(512) by 8–bits
organization	4 frames	3 frames	
configurable	yes	no	
tilable	yes	no	
AMs	1	1	pin–compatible 44–pin video
AM input	24–bits	8–bits	input connector
DMs	1	1	different display connectors
CMs	1	1	
Video Bus			Up to 4 IMA on video bus, but
global bus	24–bits I/O	24–bits out	only one IMS on video bus
pipeline in	24–bits in	24–bits in	
pipeline out	24–bits out	no connection	

The IMA and IMS do not contain an ALU or LUTs. LUTs are located on the AMs, DM and some CMs. All acquisition, including camera set–up and selection, is done through the AM family. Figure 1–2 shows the functional blocks of the IMS.

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

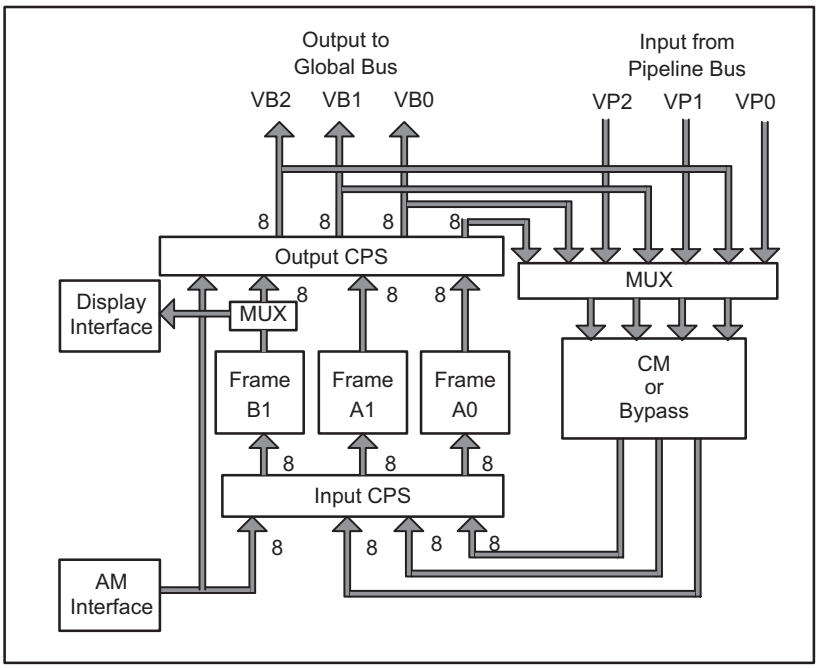


Figure 1–2. IMS Block Diagram

IMAGE MEMORY

The IMS contains either 3MB (three Megabytes) or 1.5MB of memory, logically partitioned into three 8–bit frames: A0, A1, and B1. Frame A0 and A1 can be accessed together as one 16–bit frame.

Before an application writes to frame memory B1, the function **ims_dispmode** must be called to turn off the display. Once the application is done writing, use **ims_dispmode** to turn the display back on. Any application that has a destination region of interest (ROI) in frame memory B1 writes to B1. ROIs are described in the *System–Level Software Manual*.

Scan Access

Each frame has two associated scan address generators, one for read access and one for write access. The output (read) address generators for all three frames are synchronized. This means that the size and format of the output images are all the same, only the addresses (source) are different, as illustrated in Figure 1-2.

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

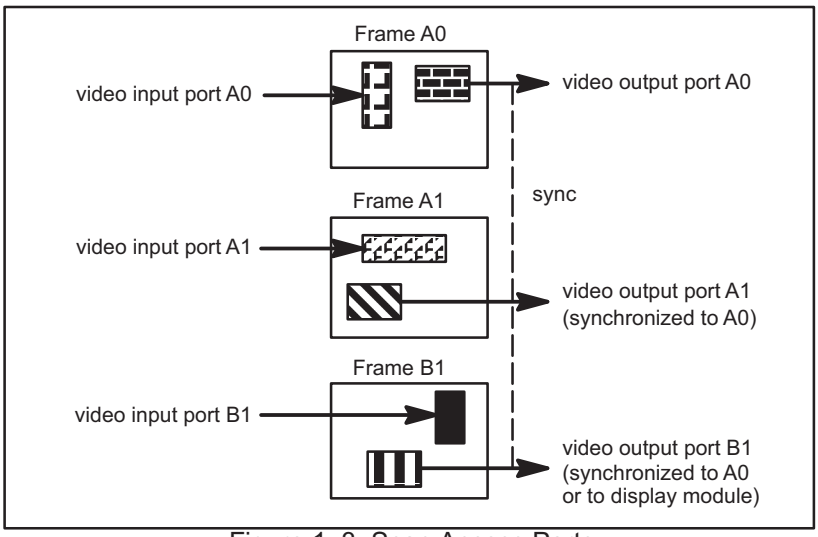


Figure 1–3. Scan Access Ports

The IMS supports pan and scroll on single–pixel boundaries. Each eight–bit frame contains two 10–bit pan registers for read and write access and two 10–bit scroll registers for read and write access.

Three 8–bit images can be written simultaneously at independent addresses. Frame A0 may acquire an interlaced image from the camera, while frame A1 receives the result from a non–interlaced AOI process, and frame B1 receives an image from the pipeline video bus. The IMS can be storing three asynchronous images in the frames at three different pan and scroll values.

Figure 1-2 also shows that images may be read from unrelated addresses. One important restriction applies. The outputs of frames A0 and A1 are synchronized and have the same timing. The pan and scroll may be different, but the frame formats are the same. If the display interface is enabled, the DM uses frame B1 as the image display buffer and frame B1 is synchronized to the DM timing. If the display interface is disabled, frame B1 is synchronized to frame A0.

The IMS internally sets up the frame buffers to receive and transmit data during each horizontal blanking period. This “set–up time” dictates the minimum AOI horizontal blank time: 40 pixels (1 us).

Zoom Acquisition

The IMS supports pixel decimation (sub–sampling) by two on each frame input. X and Y zoom is controlled independently. Figure 1-3 illustrates sub–sampling.

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

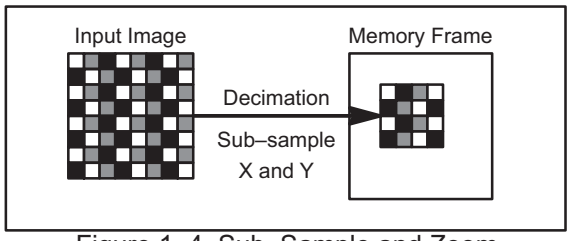


Figure 1–4. Sub–Sample and Zoom

Acquisition Control

Each eight–bit frame has an associated acquisition register. An image is acquired relative to the frame timing. The acquisition begins at the trailing edge of vertical blank and ends at the leading edge of vertical blank. Alternatively, this operation begins upon receiving a trigger pulse from the AM.

New pan and scroll values are loaded near the end of vertical blank.

An interrupt can be programmed to signify the end of the acquisition period for any frame.

Memory Write Masks

Each eight–bit frame has an associated write mask for bit–plane protection. The write masks are used for *both* scan and host access. The masks are always enabled. The value 1 enables writing the bit–plane; a 0 prevents writing. The mask registers are undefined at power–up.

Interrupts

The IMS interrupts the host on one of seven (switch–selectable) request lines. When an interrupt occurs, the vector address to the appropriate interrupt service routine is supplied. The Reset On Register Access (RORA) protocol is supported. An interrupt can be programmed to occur upon detection of one of the following events:

- Camera vertical blank
- Video bus vertical blank
- End of acquisition
- External Trigger
- Module interrupt

CROSS–PORT SWITCH

The IMS contains two dynamically reconfigurable cross–port switches (CPS) for data path control. The input CPS has eight 8–bit inputs and three

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

8–bit outputs for switching data from camera, CM outputs, and memory into the three frame memory write ports. The output CPS has five 8–bit inputs and four 8–bit outputs for switching the frame memory read ports, camera, and a constant (K) to the CM inputs and the video bus.

The two CPS appear to software as 8–bit path registers and one control register. The control register defines the bus width. The path registers select which input drives a particular output.

Keep the following points in mind when setting the CPS.

- The application must set the paths with the cross–port switch (source to destination) before an acquisition or processing event is started.
- The read and write pan and scroll registers for the frame buffers have to be initialized in the vertical blank period preceding the process.
- When using AOI one shot, set the acquisition registers for the target device before releasing the AOI.

The section “Frame Timing” contains an example of setting the CPS.

CLOCK SPEED

The IMS can change the system clock speed. Not all the computational modules (CMs) operate at the 40 MHz nominal rate. The CM–BC, for example, operates at 20 MHz. The choices of clock speed are 10, 20 and 40 MHz. When the clock speed is changed, all video buses within the system change speed immediately. The clock cycle is guaranteed not to glitch during the changeover.

All video buses within the system run at the selected speed. Make sure the system speed exceeds the camera rate. If the camera is operating faster than the system, the FIFO buffer between the AM and the IMS may not get unloaded correctly and lines could be lost.

If the clock speed is only changed during part of the process, it should be changed when the AOI from the previous step is complete and the AOI generator is halted. For example, the clock speed might be changed to perform a correlation using the CM–BC.

FRAME TIMING

A frame is defined as a rectangular region of memory, X pixels by Y lines in size. The IMS has multiple frame timing sources. The AM generates timing based on the camera input. The DM generates timing relative to the display output. The CM generates timing relative to its input. The IMS contains one AOI timing generator for processing, and the video bus can provide asynchronous timing from the pipeline bus.

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

The flexibility of the IMS allows any timing source to drive the various local and external devices (frame stores, CM, global bus). Refer to the function **ims_set_cpspath** (page 2–56) for path restrictions.

AOI Timing Generator

The IMS has a single area–of–interest (AOI) timing generator, AOI0. The IMS drives the global video bus with AOI0 or AM timing. The IMS is always the video bus master. Only one IMS may be installed in a system.

AOI operation defines the duration and blanking intervals. The AOI timing generator uses register values to generate new horizontal and vertical durations, and horizontal and vertical reset signals. The AOI blanking intervals are much shorter than the RS–170 or CCIR standards, reduced to improve processing performance. The AOI size registers define a video image of any size from four pixels by one line up to 4096 pixels by 1024 lines. AOI mode uses non–interlaced addressing.

Set the AOI size and blanking registers before enabling AOI processing. Enabling AOI mode switches the frame to AOI timing immediately. Care should be taken to synchronize switching with vertical blank if acquisition has not completed.

This section describes how to set–up the cross–port switch to send data from the source frame buffer to a CM for processing and then to a destination frame buffer. The AOI timing determines how the cross–port switch is set–up. The examples in this section use the function **ims_set_cpspath**.

To set–up a path from the source ROI to the destination ROI through the CM, the application needs to set–up three connections, as illustrated in Figure 1–5.

- Source frame to video bus
- Video bus to CM
- CM to destination frame

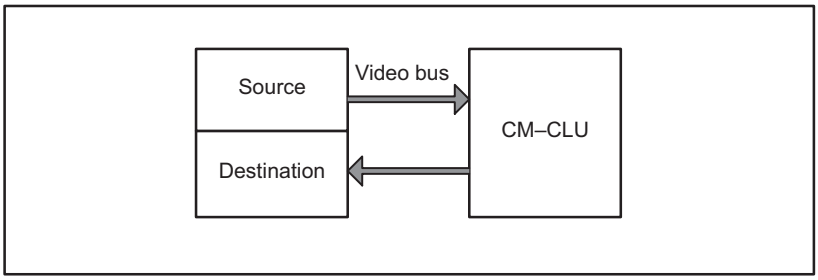


Figure 1–5. Example Path

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

The function **ims_set_cpspath** must be called three times to make this path. Consider this example, the source ROI is located in frame A0; the CM is a CM–CLU; the destination ROI is in frame B1. The code fragment below sets up the data path from source to CM–CLU to destination.

```
...
/* Set the path from source to video bus */
ims_set_cpspath(imsptr, IMS_FA0, IMS_VB0);
/* Set the path from video bus to CM-CLU */
ims_set_cpspath(imsptr, IMS_VB0, IMS_CM0);
/* Set the path from CM-CLU to destination */
ims_set_cpspath(imsptr, IMS_CM0, IMS_FB1);
...
```

The timing for the cross–port switch is determined by the data source.

Finally the application must create source and destination ROIs with the timing shown above.

```
...
source = itx_roi_create(imsptr,X_FA0,0,0,
  256,256, X_VBUSTMG, 0, 8, 0, 0, *opt);
dest = itx_roi_create(imsptr, X_FB1, 0,0,
  256, 256, X_CMTMG, 0, 8, 0, 0, *opt);
...
```

VIDEO BUS INTERFACE

The video bus controls communication between MVC 150/40 boards. This bus has been designed to run at a 40 MHz pixel rate. The video bus has a global bus driven by the IMS and a pipeline bus providing input to the IMS from an adjacent module. The global bus connects to all MVC 150/40 boards on the video bus front–plane. The pipeline bus connects the input of one board to the output of an adjacent board. The pipeline bus transmits 24–bit images to the IMS.

The the IMS drives the global bus with either camera or AOI0 timing. An IMS can drive data from any of its three frame buffers onto any byte of this bus.

A large system can include the IMS; up to four IMA boards (each with an AM, a CM, and a DM); and up to four CMC boards with up to three CMs each.

MODULE INTERFACES

The IMS provides an interface to each of the MVC 150/40 daughter boards. The modules are the same size; but the connectors have a different sizes and location to prevent incorrect installation. The modules are mapped into the

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

host VME memory address space, at the same base address as the image memory. The Page register selects between memory and modules for access.

Acquisition Module

The IMS provides a camera interface for the AM family. The AM contains line FIFOs which are loaded at the camera rate and unloaded at the 40 MHz operating rate of the IMS. The interface supports 8 bits of data, external triggering and interrupts. To access register 0 on the AM, set the bit IMS_MEMEN to ON and IMS_PGKEY to IMS_SLOT2KEY and set read/write memory word offset 0. Both these bits are in the IMS's page register. The AM must be located on slot 2 on the IMS.

Computational Module

The CM interface has 32 data inputs and 24 data outputs. The interface supports any of the MVC 150/40 CM modules, including the CM–PA "master" mode. In master mode, the CM–PA becomes local master of the IMS. It has access to all IMS registers, including modules but excluding memory. Refer to CM–PA manual for details. To access register 0 on the CM, set bit IMS_MEMEN to ON and IMS_PGKEY to IMS_SLOT1KEY and read/write memory word offset 0. Both these bits are in the IMS's page register. The CM must be in slot 1.

Display Module

The DM interface slaves frame B1 to the display output. The DM supports resolutions up to 1K x 768 non–interlaced and 1K by 1K interlaced. To access register 0 on the DM, set bits IMS_MEMEN to ON and IMS_PGKEY to IMS_SLOT0KEY and read/write memory word offset 0. Both these bits are in the IMS's page register. The DM must be in slot 0.

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

SOFTWARE INTRODUCTION

All the functions that operate the IMS begin with the prefix "ims_." To use these functions in an application, include the header files "ims.h" and "itx-core.h." These header files pull in the appropriate system and IMS data structures and symbol definitions.

Argument Types

The argument types used in IMS functions are:

- Signed integers (int)
- Characters (char)
- Long integer (long)
- Floating–point integers (float)
- Double precision floating–point integers (double)
- 16–bit value (short)

On a Sun system, an integer value is 32 bits. On a DOS system, an integer value is 16 bits. Whenever possible, short and long argument types are used. To ensure that the proper size data element is used, use these additional data types:

- Eight–bit unsigned (BYTE)
- 16–bit unsigned (WORD)
- 32–bit unsigned (DWORD)

For example, a value declared as type WORD is an unsigned 16–bit value. Arrays of these values and pointers to these values are also used.

The IMS software uses three data structures:

- MODCNF defines the general board configuration for the MVC 150/40. It contains all the necessary information about the IMS.
- IMSCNF defines the IMS–specific configuration for the MVC 150/40.
- IFF_FPB describes the format of the image file discussed in the *System–Level Software Manual*. Every image file used in an application must have a pointer to this structure.

IMS Selection

In an application program, pointers to the IMS should be set up immediately after configuring the system. The application calls **itx_get_slot_modcnf** which returns a pointer to a MODCNF structure. Any calls to IMS functions

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

use this MODCNF structure to indicate the IMS to perform on. Each board and module in a MVC 150/40 has a pointer. Because the functions in ITEX IMS receive a pointer to MODCNF, the functions have access to the necessary information about the board.

The IMS software supplies the functions to return a board pointer for a daughter board attached to an IMS. The function **ims_get_slot0** retrieves a pointer to the DM–PC, **ims_get_slot1** to the CM–CLU and **ims_get_slot2** to the AM–VS.

Returned Values

In many cases, functions return a short value. This value is either an error code or a value as defined in the header file "itxerr.h" or "imserr.h." If no error has occurred, the function returns the value IMS_NO_ERROR; otherwise, it returns a negative number that specifies the error. These error codes are noted in the function descriptions as "Returned Values." Table 1–1 lists the error messages.

Table 1–1. Error Messages

Error	*Description*
IMS_NO_ERROR	Operation successfully completed.
IMS_BAD_DMODCNF	Bad daughter board MODCNF pointer
IMS_NOT_MASTER	This IMS is not video bus master
IMS_BAD_ARG	Invalid frame or AOI generator selection

The IMS can also return system errors; these error start with "ITX_" and are described in the MVC 150/40 *System–Level Software Manual*.

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

Initialization Functions

The functions in Table 1–2 initialize and set–up the IMS. The acquisition module (AM) on the IMS performs camera set–up and selection. See the software manual for the AM–VS or AM–DIG for more information.

Table 1–2. Initialization and Set–up Functions

Functions	Descriptions	Page
Pointers		
ims_get_mod	Return a pointer to an IMS	2–36
ims_get_slot0	Return a pointer to a DM	2–38
ims_get_slot1	Return a pointer to a CM	2–39
ims_get_slot2	Return a pointer to an AM	2–40
Initialization		
ims_iregs	Initialize the registers	2–43
ims_dregs	Display the register values	2–28
Timing and Cross–Port Switch		
ims_set_aoi	Set the AOI dimensions	2–55
ims_get_aoi	Return the AOI dimensions	2–34
ims_set_cpspath	Set the cross–port switch	2–56

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

Synchronization and Acquisition Functions

The functions in Table 1–3 synchronize the IMS and acquire images. Synchronization functions coordinate acquisitions with the camera frame timing, to ensure that a complete camera image is acquired.

Before calling the acquisition functions, an application must set–up the cross–port switch with the function **ims_set_cpspath** and the Write pan and scroll registers with **ims_set_wrroam**.

AOI Timing

The register AOIXS0 is loaded at the end of the active line. The register AOIXB0 is loaded at the end of the inactive line. The register AOIYS0 is loaded at the end of the active frame. The register AOIYB0 is loaded at the end of the inactive frame. The register AOIHLT0 is sampled at the end of the inactive frame. The register OSHOT0 is sampled at the end of the inactive frame. The register AOILK0 is sampled at the end of the camera vertical blank.

Since AOIHLT0, OSHOT0 and AOILK0 are sampled when the AOI vertical blank counter reaches zero, this sampling guarantees that any register loaded before setting these bits is effective for the next frame.

ROI versus AOI

An AOI (area–of–interest) is a timing format used to process and image. AOI timing is optimized for digital processing, and for the processing delays on the hardware. AOI timing usually allows faster processing than using camera or display timing. Processing can be slaved to the camera or display, or driven by an AOI timing generator. The video bus or pipeline bus must use AOI timing when they are not slaved to a camera or display.

An ROI (region–of–interest) is a data structure, used as a location (source or destination) for image data. An ROI can be a camera (data source) display monitor (destination) image memory on a MVC 150/40 board, system memory, system disk. An ROI can use any timing source: camera, display, AOI timing generator, pipeline bus timing, video bus timing.

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

Table 1–3. Acquisition Functions

Functions	Descriptions	Page
Synchronization		
ims_wait_vb	Wait for next vertical blank	2–66
ims_wait_notvb	Wait until vertical blank is past	2–65
ims_waitacq	Wait for acquisition to complete	2–67
Acquisition		
ims_snap	Acquire a single image	2–62
ims_snap_async	Acquire a single image asynchronous	2–9
ims_grab	Acquire multiple images	2–42
ims_freeze	Stop acquisition	2–32
ims_psreset	Set the pan and scroll registers	2–46
AOI Acquisition		
ims_aoi_snap	Acquire a single image with the AOI generator	2–8
ims_aoi_snap_async	Acquire a single image with the AOI generator asynchronous	2–63
ims_aoi_grab	Acquire multiple images with the AOI generator	2–5
ims_aoi_snap16	Acquire a single 16–bit image with the AOI generator	2–11
ims_aoi_snap16_async	Acquire a single 16–bit image with the AOI generator asynchronous	2–12
ims_aoi_grab16	Acquire multiple 16–bit images with the AOI generator	2–6
ims_aoi_halt	Stop an AOI timing generator	2–7
ims_aoi_start	Start an AOI timing generator	2–13
ims_aoi_1shot	Perform one clock cycle	2–3
ims_aoi_1shot_async	Perform one clock cycle asynchronous	2–4
low–level command		
ims_acqbits	Set or Return the Acquisition Command Bits	2–2

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

Read and Write Pixel Functions

The functions in Table 1–4 and Table 1–5 read and write pixel values to and from the frame buffers directly. They do not use the region–of–interest (ROI) data structure. The *System–Level Software Manual* describes the ROIs.

Table 1–4. Read and Write Pixel Functions

Function	Description	Page
Pixel Read and Write Functions		
ims_rpix	Read a single pixel value	2–50
ims_wpix	Write a single pixel value	2–71

You must call the function **ims_set_lineio** before calling a function listed in Table 1–5. The configuration file sets the memory size and pixel depth. The functions in Table 1–5 use these values when reading and writing the lines. If an application needs to change those values from the configuration value, use the Module Structure functions in Table 1–7.

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

Table 1–5. Read and Write Line Functions

Function	Description	Page
Set–Up for BYTE, WORD, and DWORD functions		
ims_set_lineio	Set–up frame buffer for access	2–58
BYTE Value		
ims_rhbline	Read a horizontal line of BYTE values	2–47
ims_whbline	Write a horizontal line of BYTE values	2–68
ims_chbline	Clear a horizontal line to a BYTE value	2–15
ims_rvbline	Read a vertical line of BYTE values	2–51
ims_wvbline	Write a vertical line of BYTE values	2–72
ims_cvbline	Clear a vertical line to a BYTE value	2–23
WORD Value		
ims_rhwline	Read a horizontal line of WORD values	2–49
ims_whwline	Write a horizontal line of WORD values	2–70
ims_chwline	Clear a horizontal line to WORD value	2–17
ims_rvwline	Read a vertical line of WORD values	2–53
ims_wvwline	Write a vertical line of WORD values	2–74
ims_cvwline	Clear a vertical line to a WORD value	2–25
DWORD Value		
ims_rhdwline	Read a horizontal line of DWORD values	2–48
ims_whdwline	Write a horizontal line of DWORD values	2–69
ims_chdwline	Clear a horizontal line to a DWORD value	2–16
ims_rvdwline	Read a vertical line of DWORD values	2–52
ims_wvdwline	Write a vertical line of DWORD values	2–73
ims_cvdwline	Clear a vertical line to a DWORD value	2–24

Area Functions

The functions in Table 1–6 operate on areas and frames: copy, clear, and display. These functions do not use ROIs.

Table 1–6. Area Functions

Function	Description	Page
ims_clr_frame	Clear an frame	2–20
ims_clr_area	Clear an area	2–19
ims_cp_area	Copy an area	2–21
ims_dump_mem	Display an area	2–29

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

Module Structure Functions

The functions in Table 1–7 return and change the values in the IMS data structure MODCNF, and write the values to the appropriate registers. This structure is created at system initialization, with default values from the configuration file.

Table 1–7. Module Structure Functions

Function	Description	Page
Host Interface		
ims_mapsize	Set or return the memory map size	2–44
ims_set_mapformat	Set the memory map format and the memory base address	2–59
ims_get_adrform	Return the address format	2–33
ims_get_mapadr	Return the VME memory map address	2–35
Video Mode		
ims_pixsz	Set or return the pixel size	2–45
ims_vbusmode	Set the video bus timing	2–64
ims_sel_slave	Select or Return a slave to output to the video bus	2–54
ims_clkspeed	Set or return the clock speed	2–18
ims_dispmode	Set or return the display mode	2–27
ims_bitmask	Set or return the bit mask	2–14
ims_delay_acqen	Begin a Delayed Acquire	2–26
Pan and Scroll		
ims_set_rdroam	Set the read pan and scroll registers	2–60
ims_get_rdroam	Return the read pan and scroll registers	2–37
ims_set_wrroam	Set the write pan and scroll registers	2–61
ims_get_wrroam	Return the write pan and scroll registers	2–41
ims_dzoom	Zoom frame B1 for display	2–30
ims_dzoomxy	Set frame B1 X and Y zoom factors for diplay	2–31
Interrupt		
~~**ims_intr_vec**~~	~~Set or return the interrupt vector~~	
~~**ims_ien_bits**~~	~~Enable or disable the interrupt~~	
~~**ims_dbrd_intr**~~	~~Enable or disable the interrupt on a daughter board~~	
~~**ims_ext_intr**~~	~~Enable or disable the external trigger interrupt~~	
~~**ims_frame_intr**~~	~~Enable or disable the frame interrupt~~	
~~**ims_intr_priority**~~	~~Set or return the interupt priority level~~	

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

CHAPTER 2
IMS Functions

The IMS functions are organized in alphabetical order in this chapter. Chapter 1 discusses their functional relationships, and contains tables of functional groupings for: initialization, synchroniatation, acquisition, pixel read and write, line read and write, image read and write, area operations, and module structure functions.

All IMS function prototypes, and global defines specific to the IMS, are found in the file “ims.h” and all register names are defined in the file “ims-reg.h”.

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

ims_acqbits — Set or Return the Acquisition Command Bits

Syntax short **ims_acqbits**(MODCNF ***mod*, short *frame* short *mode*)

ims_ptr Pointer to the IMS.

frame Frame to acquire into:
IMS_FRAMEA0 — Frame A0.
IMS_FRAMEA1 — Frame A1.
IMS_FRAMEB1 — Frame B1.

mode Acquisition mode:
IMS_FREEZE — Stop acquisition
IMS_SNAP — Acquire a single image frame
IMS_GRAB — Acquire images continuously
INQUIRE — Return the pending command (current state of command bits)

Description

The function **ims_acqbits** writes an acquisiton command to, or returns the state of, the acquistion command bits.

A new acquisition begins at the leading edge of a vertical blank, and terminates at the trailing edge of a vertical blank. A new command can be written at any time. Retruning the status of the acquisition command bits shows the next command pending, not the current acquisiton.

Returned Values

Pending acquisiton command (*not* the current acquisition).

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_ABTSA0, IMS_ABTSA1, IMS_ABTSB1

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

ims_aoi_1shot — Perform One Clock Cycle

Syntax short **ims_aoi_1shot**(MODCNF **ims_ptr,* short *aoisel*)

ims_ptr Pointer to the IMS.

aoisel AOI timing generator:
IMS_AOISEL0 — Select AOI generator Zero.

Description

This function enables AOI mode, waits for the AOI status bit to indicate the AOI generator is halted, sets the pan and scroll registers (ims_psreset), writes to the one–shot bit for the AOI generator to complete a single clock cycle. Finally it waits for the AOI status bit to indicate the AOI generator is again halted (single cycle completion). This function does not perform an acquisition.

Error Values

IMS_NO_ERROR — Operation successfully completed.
ITX_TIMEOUT — AOI status bits are not ready or are stuck and the acquisition could not be completed.

Hardware Modified

IMS_AOIMODE, IMS_OSHOT0, all pan and scroll registers.

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

ims_aoi_1shot_async — Perform One-Shot Asynchronously

Syntax short **ims_aoi_1shot_async**(MODCNF ***ims_ptr*, short *aoisel*)

ims_ptr Pointer to the IMS.

aoisel AOI timing generator:
IMS_AOISEL0 — Select AOI generator Zero.

Description

After the AOI generator is halted, the function **ims_aoi_1shot_async** waits for the AOI ready bits to be set, sets the pan and scroll registers, writes the one–shot bit for the AOI generator to complete a clock cycle. This function reteruns immediately, without waiting for the 1shot cycle to complete. This function does not set the acquisition command bits, or perform an acquisition.

Note ims_aoi_1shot *starts the acquisiton and polls until the acquisition is complete.* ims_aoi_1shot_async *starts the acquisition and returns immediately, without waiting for the acquisition to complete.*

Error Values

IMS_NO_ERROR — Operation successfully completed.
ITX_TIMEOUT — AOI status bits are not ready or are stuck and the acquisition could not be completed.

Hardware Modified

IMS_AOIMODE, IMS_OSHOT0, all pan and scroll registers.

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

ims_aoi_grab — Acquire Multiple Images with the AOI Generator

Syntax short **ims_aoi_grab**(MODCNF **ims_ptr,* short *aoisel*, short *frame*)

ims_ptr Pointer to the IMS.

aoisel AOI timing generator:
IMS_AOISEL0 — Select AOI generator Zero.

frame Frame to acquire into:
IMS_FRAMEA0 — Frame A0.
IMS_FRAMEA1 — Frame A1.
IMS_FRAMEB1 — Frame B1.

Description

This function enables AOI mode and halts the AOI generator, waits for a vertical blank in *frame*, sets the pan and scroll registers (ims_psreset), and sets the acquisition bits for *frame* to acquire multiple images. It then releases the AOI generator to free–running mode. Multiple eight–bit images are acquired.

Before calling this function, an application must set–up the cross–port switch with the function **ims_set_cpspath** and the pan and scroll shadow registers with **ims_set_wrroam**.

The function **ims_aoi_grab** uses the AOI generator that can be started and stopped to begin the acquisition; whereas, the function **ims_grab** uses a continuously running clock.

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid *frame* or invalid *aoisel*.
ITX_TIMEOUT — VB bit stuck for *frame* or can't set acquisition to "IMS_FREEZE" status.

Hardware Modified

IMS_AOIMODE, IMS_AOIHLT0, and
IMS_ABTSA0 or IMS_ABTSA1 or IMS_ABTSB1, all pan and scroll registers.

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

ims_aoi_grab16 — Acquire Multiple 16–Bit Images with the AOI Generator

Syntax short **ims_aoi_grab16**(MODCNF **ims_ptr,* short *aoisel*, short *frame*)

ims_ptr Pointer to the IMS.

aoisel AOI timing generator:
IMS_AOISEL0 — Select AOI generator Zero.

frame Frame to synchronize:
IMS_FRAMEA — Frame A.

Description

This function enables AOI mode and halts the AOI generator, waits for a vertical blank in FRAMEA, sets the pan and scroll registers, (ims_psreset) and sets the acquisition bits for FRAMEA to acquire multiple 16–bit images. It then releases the AOI generator to free–running mode. Multiple 16–bit images are acquired.

Before calling this function, an application must set–up the cross–port switch with the function **ims_set_cpspath** and the pan and scroll shadow registers with **ims_set_wrroam**.

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid *frame* or invalid *aoisel*
(must be IMS_FRAMEA and IMS_AOISEL0).
ITX_TIMEOUT — VB bit stuck for *frame*.

Hardware Modified

IMS_AOIMODE, IMS_AOIHLT0, IMS_ABTSA0, IMS_ABTSA1, all pan and scroll registers.

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

ims_aoi_halt — Stop an AOI Timing Generator

Syntax short **ims_aoi_halt**(MODCNF **ims_ptr,* short *aoisel*)

ims_ptr Pointer to the IMS.

aoisel AOI timing generator:
IMS_AOISEL0 — Select AOI generator Zero.

Description

This function stops the AOI timing generator and any associated processing. The AOI generator remains in vertical blank until a one–shot, or until released. This function does *not* put the IMS in AOI mode.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_AOIHLT0

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

ims_aoi_snap — Acquire a Single Image with the AOI Generator

Syntax short **ims_aoi_snap**(MODCNF **ims_ptr,* short *aoisel*, short *frame*, short *mode*)

ims_ptr Pointer to the IMS.

aoisel AOI timing generator:
IMS_AOISEL0 — Select AOI generator Zero.

frame Frame to acquire into:
IMS_FRAMEA0 — Frame A0.
IMS_FRAMEA1 — Frame A1.
IMS_FRAMEB1 — Frame B1.

mode AOI timing generation period:
IMS_AOIOSHOT — Set the one–shot start bit to start a single AOI cycle acquisition
IMS_AOISTART — Release the AOI generator to free–running

Description

This function enables AOI mode and halts the AOI generator, waits for a vertical blank in *frame*, and sets the pan and scroll registers (ims_psreset) and the acquisition bits for *frame* to acquire a single image. It then either releases the AOI generator or fires a one–shot (single AOI cycle) based on *mode*. Finally it checks that the acquisition bits are set to "IMS_FREEZE" status. A single eight–bit image is acquired.

Before calling this function, an application must set–up the cross–port switch with the function ims_set_cpspath and the pan and scroll shadow registers with ims_set_wrroam.

The function ims_aoi_snap uses the AOI generator that can be started and stopped to begin the acquisition; whereas, the function ims_snap uses a continuously running clock.

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid *frame* or invalid *aoisel*.
ITX_TIMEOUT — VB bit stuck for *frame* or can't set acquisition to "IMS_FREEZE" status.

Hardware Modified

IMS_AOIMODE, IMS_AOIHLT0, IMS_OSHOT0, all pan and scroll registers, and IMS_ABTSA0 or IMS_ABTSA1 or IMS_ABTSB1

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

ims_aoi_snap_async — Acquire 8-bit AOI Asynchronously

Syntax short **ims_aoi_snap_async**(MODCNF **ims_ptr*, short *aoisel*, short *frame*, short *mode*)

ims_ptr Pointer to the IMS.

aoisel AOI timing generator:
IMA_AOISEL0 — Select AOI generator Zero.

frame Frame to acquire into:
IMS_FRAMEA0 — Frame A0.
IMS_FRAMEA1 — Frame A1.
IMS_FRAMEB1 — Frame B1.

mode AOI timing generation period:
IMS_AOIOSHOT — Set the One–Shot bit to start a single AOI frame.
IMS_AOISTART — Release the AOI generator to free–running mode.

Description

The function **ims_aoi_snap_async** halts the selected AOI generator, waits for a vertical blank in *frame*, and sets the pan and scroll registers and the acquisition bits for *frame* to acquire a single image. It then starts either a single or a continuous acquisition, based on *mode*. This function returns immediately, without waiting for the acquisition to complete. A single eight–bit image is acquired.

Before calling this function, an application must set–up the cross–port switch with the function ims_set_cpspath and the pan and scroll shadow registers with ims_set_wrroam.

The function ims_aoi_snap uses the AOI generator that can be started and stopped to begin the acquisition; whereas, the function ims_snap uses a continuously running clock.

Note ims_aoi_snap *starts the acquisiton and polls until the acquisition is complete.* ims_aoi_snap_async *starts the acquisition and returns immediately, without waiting for the acquisition to complete.*

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid acquisition buffer.
ITX_TIMEOUT — VB bit stuck for *frame* or can't set acquisition to "IMA_FREEZE" status.

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

Hardware Modified

IMS_AOIMODE, IMS_AOIHLT0, IMS_OSHOT0, all pan and scroll registers, and IMS_ABTSA0 or IMS_ABTSA1 or IMS_ABTSB1

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

ims_aoi_snap16 —Acquire a Single 16–Bit Image with the AOI Generator

Syntax short **ims_aoi_snap16**(MODCNF **ims_ptr,* short *aoisel*, short *frame*, short *mode*)

ims_ptr Pointer to the IMS.

aoisel AOI timing generator:
IMS_AOISEL0 — Select AOI generator Zero.

frame Frame to acquire into:
IMS_FRAMEA — Frame A.

mode AOI timing generation period:
IMS_AOIOSHOT — Set the One–Shot bit to start a single AOI frame.
IMS_AOISTART — Release the AOI generator to free–running mode.

Description

This function enables AOI mode and halts the AOI generator, waits for a vertical blank in frame A0, sets the pan and scroll registers (ims_psreset), and the acquisition bits for 16–bit FRAMEA to acquire a single image. It then either releases the AOI generator or fires a one–shot (single AOI cycle) based on *mode*. Finally this function checks that the acquisition bits are set to "IMS_FREEZE" status. A single 16–bit image is acquired.

Before calling this function, an application must set–up the cross–port switch with the function ims_set_cpspath and the pan and scroll shadow registers with ims_set_wrroam.

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid *frame*, or invalid *aoisel* (must be IMS_FRAMEA and IMS_AOISEL0).
ITX_TIMEOUT — VB bit stuck for *frame* or can't set acquisition to "IMS_FREEZE" status.

Hardware Modified

IMS_AOIMODE, IMS_AOIHLT0, IMS_OSHOT0, IMS_ABTSA0, IMS_ABTSA1, all pan and scroll registers.

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

ims_aoi_snap16_async —Acquire 16-bit AOI Asynchronously

Syntax short **ims_aoi_snap16_async**(MODCNF **ims_ptr*, short *aoisel*, short *frame*, short *mode*)

ims_ptr Pointer to the IMS.

aoisel AOI timing generator:
IMS_AOISEL0 — Select AOI generator Zero.

frame Frame to acquire into:
IMS_FRAMEA — Frame A.

mode AOI timing generation period:
IMS_AOIOSHOT — Set the One–Shot bit to start a single AOI frame.
IMS_AOISTART — Release the AOI generator to free–running mode.

Description

The function **ims_aoi_snap16_async** halts the AOI generator, waits for a vertical blank in *frame*, and sets the pan and scroll registers and the acquisition bits for *frame* to acquire a single image. It then starts either a single or a continuous acquisition, based on *mode*. This function returns immediately, without waiting for the acquistion to complete. A single 16–bit image is acquired.

Before calling this function, an application must set–up the cross–port switch with the function ims_set_cpspath and the pan and scroll shadow registers with ims_set_wrroam.

Note ims_aoi_snap16 *starts the acquisiton and polls until the acquisition is complete.* ims_aoi_snap16_async *starts the acquisition and returns immediately, without waiting for the acquisition to complete.*

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid acquisition buffer.
ITX_TIMEOUT — VB bit stuck for *frame* or can't set acquisition to "IMA_FREEZE" status.

Hardware Modified

IMS_AOIMODE, IMS_AOIHLT0, IMS_OSHOT0, IMS_ABTSA0, IMS_ABTSA1, all pan and scroll registers.

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

ims_aoi_start — Start an AOI Timing Generator

Syntax short **ims_aoi_start**(MODCNF **ims_ptr,* short *aoisel*)

ims_ptr Pointer to the IMS.

aoisel AOI timing generator:
IMS_AOISEL0 — Select AOI generator Zero.

Description

This function releases the (halted) AOI generator to free–running mode.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_AOIHLT0

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

ims_bitmask — Set or Return the Bit Mask

Syntax short **ims_bitmask**(MODCNF **ims_ptr*, WORD *reg*, short *mode*, WORD **value*)

ims_ptr Pointer to the IMS.

value	Pointer to the value returned or the value to set the bit mask.
reg	Mask register to set or to return: IMS_MASKA — Bit mask register A. IMS_MASKB — Bit mask register B.
mode	Indicates if *value* is set or returned: ITX_SET — Set *reg* to *value*. ITX_GET — Return *value* from *reg*.

Description

This function sets or returns *val* to or from *reg*, depending on *mode*. If *mode* is ITX_SET, this function changes the IMS module structure from its previous value.

Returned Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — *reg* is invalid.

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

ims_chbline — Clear a Horizontal Line to a BYTE Value

Syntax void **ims_chbline**(DWORD *address*, short *num_ops*, short *freq*, BYTE *value*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to clear.
freq	Frequency with which to clear, from 0 to 32767:
	0 Clear the same location.
	1 Clear every location.
	2 Clear every other location.
	3 Clear every third location.
	. . .
	32,767 Clear every 32,768th location.
value	Value to clear pixels to.

Description

This function clears a horizontal line to a BYTE value. You must select values for *num_ops* and *freq* that do not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

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

rims_chdwline — Clear a Horizontal Line to a DWORD Value

Syntax void **ims_chdwline**(DWORD *address*, short *num_ops*, short *freq*, DWORD *value*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to clear.
freq	Frequency with which to clear, from 0 to 32767:
	0 Clear the same location.
	1 Clear every location.
	2 Clear every other location.
	3 Clear every third location.
	. . .
	32767 Clear every 32,768th location.
value	Value to clear pixels to.

Description

This function clears a horizontal line to a DWORD value. You must select values for *num_ops* and *freq* that do not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

The function **ims_pixsz** defines the pixel size. Each DWORD cleared by **ims_chdwline** contains either two horizontally adjacent 16–bit pixels or four horizontally adjacent 8–bit pixels.

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

ims_chwline — Clear a Horizontal Line to WORD Value

Syntax void **ims_chwline**(DWORD *address*, short *num_ops*, short *freq*, WORD *value*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to clear.
freq	Frequency with which to clear, from 0 to 32767:
0	Clear the same location.
1	Clear every location.
2	Clear every other location.
3	Clear every third location.
. . .	
32,767	Clear every 32,768th location.
value	Value to clear pixels to.

Description

This function clears a horizontal line to a WORD value. You must select values for *num_ops* and *freq* that do not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

The function **ims_pixsz** defines the pixel size. Each WORD cleared by **ims_chwline** contains either one 16–bit pixel or two horizontally adjacent 8–bit pixels.

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

rims_clkspeed — Set or Return the Clock Speed

Syntax short **ims_clkspeed**(MODCNF **ims_ptr*, short *speed*)

ims_ptr Pointer to the IMS.

speed Clock speed:
IMS_CLK40 — Set this system clock to 40 Mhz.
IMS_CLK20 — Set this system clock to 20 Mhz.
IMS_CLK10 — Set this system clock to 10 Mhz.
INQUIRE — Return the current value.

Description

This function sets the clock speed for the system. Only the master can change the system clock speed. Slave boards can not affect the clock. This function changes the IMS module structure from its previous value.

Returned Values

Current clock speed.

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_NOT_MASTER — This IMS is not a video bus master and can not select a slave IMS. A jumper sets the video–bus master.

Hardware Modified

IMS_CLKSPD

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

ims_clr_area — Clear an Area

Syntax short **ims_clr_area**(MODCNF ***ims_ptr*, short *frame*, short *x*, short *y*, short *dx*, short *dy*, WORD *value*)

ims_ptr Pointer to the IMS.

frame	Frame to clear: IMS_FRAMEA — Frame A, 16–bit frame. IMS_FRAMEA0 — Frame A0, 8–bit frame. IMS_FRAMEA1 — Frame A1, 8–bit frame. IMS_FRAMEB1 — Frame B1, 8–bit frame.
x	Horizontal location of the area in *frame* to clear, in the range of 0 to 1023.
y	Vertical location of the area in *frame* to clear, in the range of 0 to 1023 (0 to 511 on 1.5MB IMS).
dx	Horizontal length of the area to clear, in the range of 1 to 1023.
dy	Vertical length of the area to clear, in the range of 1 to 1023 (0 to 511 on 1.5MB IMS).
value	Value to clear the area to in the range of 0 to 255 for 8–bit frames, 0 to 1023 for 16–bit frame.

Description

This function clears the area in *frame* to *value*. *x* plus *dx* cannot be greater than 1024; *y* plus *dy* cannot be greater than 1024 for 3MB IMS, (512 for 1.5MB IMS).

This function checks the parameters against the size of IMS frame memory, and clips the area to fit. The clear is performed by the host, using **ims_chbline** or **ims_chwline**.

Note *This function internally calls* **ims_set_lineio***, and may change the address value from a previous value.*

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Not a valid *frame*.

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

ims_clr_frame — Clear a Frame

Syntax short **ims_clr_frame**(MODCNF **ims_ptr*, short *frame*, WORD *value*)

ims_ptr Pointer to the IMS.

frame Frame to clear:
IMS_FRAMEA — Frame A, 16–bit frame.
IMS_FRAMEA0 — Frame A0, 8–bit frame.
IMS_FRAMEA1 — Frame A1, 8–bit frame.
IMS_FRAMEB1 — Frame B1, 8–bit frame.

value Value to clear the area to in the range of 0 to 255 for 8–bit frames, 0 to 1023 for 16–bit frame.

Description

This function clears *frame* to *value*. The clear is performed by the host, using **ims_chbline** or **ims_chwline**.

Note *This function internally calls* **ims_set_lineio***, and may change the address value from a previous value.*

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Not a valid *frame*.

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

ims_cp_area — Copy an Area

Syntax short **ims_cp_area**(MODCNF **ims_ptr1*, short *frame1*, short *x1*, short *y1*, short *dx*, short *dy*, MODCNF **ims_ptr2*, short *frame2*, short *x2*, short *y2*)

ims_ptr1	Pointer to the IMS containing *frame1*.
frame1	Source frame:
frame	Frame to clear:
	IMS_FRAMEA — Frame A, 16–bit frame.
	IMS_FRAMEA0 — Frame A0, 8–bit frame.
	IMS_FRAMEA1 — Frame A1, 8–bit frame.
	IMS_FRAMEB1 — Frame B1, 8–bit frame.
x1	Horizontal location for the source area, in the range of 0 to 1023.
y1	Vertical location of the source area, in the range of 0 to 1023 for 3MB IMS, (0 to 511 for 1.5MB IMS).
dx	Horizontal length of the area to clear, in the range of 1 to 1024.
dy	Vertical length of the area to clear, in the range of 1 to 1024 for 3MB IMS, (1 to 512 for 1.5MB IMS).
ims_ptr2	Pointer to the IMS containing *frame2*.
frame2	Destination frame; see *frame1* for options.
x2	Horizontal location for the destination area, in the range of 0 to 1023.
y2	Vertical location of the destination area, in the range of 0 to 1023

Description

This function copies the area in *frame1* defined by *x1*, *y1*, *dx1*, *dy1* to the location in *frame2* defined by *x2*, *y2*. (*x1* plus *dx*) and (*x2* plus *dx*) cannot be greater than 1024. (*y1* plus *dy*) and (*y2* plus *dy*) cannot exceed 1024 for the 3MB IMS, (512 for 1.5MB IMS).

This function checks the parameters against the size of IMS frame memory, and clips the area to fit. The copy process is performed by the host, using horizontal line trread and write functions. The area is not copied using video transfers.

Note *This function internally calls* **ims_set_lineio***, and may change the address value from a previous value.*

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

Error Values

IMS_NO_ERROR — Operation successfully completed.
ITX_ALLOC_ERROR — Memory allocation error.
IMS_BAD_ARG — Not a valid frame. Check value given to *frame1* and *frame2*.

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

ims_cvbline — Clear a Vertical Line to a BYTE Value

Syntax void **ims_cvbline**(DWORD *address*, short *num_ops*, short *linesz*, BYTE *value*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to clear.
linesz	Horizontal frame size returned from **ims_set_lineio**.
value	Value to clear pixels to.

Description

This function clears a vertical line to a BYTE value. You must select a value for *num_ops* that does not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

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

tims_cvdwline — Clear a Vertical Line to a DWORD Value

Syntax void **ims_cvdwline**(DWORD *address*, short *num_ops*, short *freq*, DWORD *value*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to clear.
linesz	Horizontal frame size returned from **ims_set_lineio**.
value	Value to clear pixels to.

Description

This function clears a vertical line to a DWORD value. You must select a value for *num_ops* that does not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

The function **ims_pixsz** defines the pixel size. Each DWORD cleared by **ims_cvdwline** contains either two horizontally adjacent 16–bit pixels, or four horizontally adjacent 8–bit pixels.

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

ims_cvline— Clear a Vertical Line to a WORD Value

Syntax void **ims_cvwline** (DWORD *address*, short *num_ops*, short *linesz*, WORD *value*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to clear.
linesz	Horizontal frame size returned from **ims_set_lineio**.
value	Value to clear pixels to.

Description

This function clears a vertical line to a WORD value. You must select a value for *num_ops* that does not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

The function **ims_pixsz** defines the pixel size. Each WORD cleared by **ims_cvwline** contains either one 16–bit pixels, or two horizontally adjacent 8–bit pixels.

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

ims_delay_acqen — Begin a Delayed Acquire

Syntax short **ims_delay_acqen**(MODCNF **ims_ptr*)

ims_ptr Pointer to the IMS.

Description

The function **ims_delay_acqen** triggers the delayed acquisition, by writting to the ACQEN register. Each frame (A0, A1, B1) must be prevoiusly enabled for delayed acquisition, by setting the WTAQ bit in the acquisition control register: WTAQA0 in register ACQA0, WTAQA1 in register ACQA1, and WTAQB1 in ACQB1. Use the system level function **write_reg** to set these bits.

Error Values

IMS_NO_ERROR — Operation completed successfully.

ACQEN

Example

Delayed/synchronized snap into frames A1 and B1:

```
   . . .
imsmod= ims_get_mod(0);
   . . .
write_reg(imsmod, WTAQA1, 1);
write_reg(imsmod, WTAQB1, 1);
ims_acqbits(imsmod,IMS_FRAMEA1,
IMS_SNAP);
ims_acqbits(iomsmod,IMS_FRAMEB1,
IMS_SNAP);
   . . .
ims_wait_vb(imsmod);
ims_delay_acqen(imsmod);
```


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

ims_dispmode — Set or Return the Display Mode

Syntax short **ims_dispmode**(MODCNF **ims_ptr*, short *disp*)

ims_ptr Pointer to the IMS.

disp Display value:
IMS_DISP_OFF — Turn off the display.
IMS_DISP_ON — Turn the display on.
INQUIRE — Return the current value.

Description

This functions sets the display mode. The display is enabled at initialization. The option IMS_DISP_OFF prevents image data from being displayed. This function has no effect on overlays. Disabling the display module allows frame B1 to be used for processing. This function changes the IMS module structure from its previous value.

Note *Reset the DM–PC Pixel Read mask before resetting the DMEN bit. This keeps IMS from showing confused screen displays when image display is disabled.*

Returned Values

Display value.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_DMEN

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

ims_dregs — Display the Register Values

Syntax void **ims_dregs**(MODCNF **ims_ptr*)

ims_ptr Pointer to the IMS.

Description

This function displays the current values of the registers on the IMS.

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

ims_dump_mem — Display an Area of Frame Memory

Syntax short **ims_dump_mem**(MODCNF **ims_ptr*, short *frame*, short *x,* short *y*, short *dx*, short *dy*)

ims_ptr Pointer to the IMS.

frame	Frame to display: IMS_FRAMEA — Frame A, 16–bit frame. IMS_FRAMEA0 — Frame A0, 8–bit frame. IMS_FRAMEA1 — Frame A1, 8–bit frame. IMS_FRAMEB1 — Frame B1, 8–bit frame.
x	Horizontal location of the area in *frame* to clear, in the range of 0 to 1023.
y	Vertical location of the area in *frame* to clear, in the range of 0 to 1023 for the 3MB IMS, (511 for the 1.5MB IMS).
dx	Horizontal length of the area to clear, in the range of 1 to 1023.
dy	Vertical length of the area to clear, in the range of 1 to 1023 for the 3MB IMS, (511 for the 1.5MB IMS).

Description

This function displays a block of frame memory in hexidecimal format. Frame memory is displayed in the pixel depth set for *frame*, 8 or 16 bits. This function calls **ims_rhbline** or **ims_rhwline**.

Note *This function internally calls* **ims_set_lineio***, and may change the value of the address pointer from a previous value.*

Error Values

IMS_NO_ERROR — Operation successfully completed.
ITX_ALLOC_ERROR — Memory allocation error.

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

tims_dzoom — Zoom Frame B1 for Display

Syntax short **ims_dzoom**(MODCNF **ims_ptr*, short *zoom*)

ims_ptr Pointer to the IMS.

zoom zoom factor:
1 = normal display (zoom by 1)
2 — Zoom by 2
4 — Zoom by 4
INQUIRE — Return the current zoom factor

Description

The function **ims_dzoom** applies the zoom factor *zoom* to the image in frame B1 in both the X and Y (horizontal and vertical) directions. This function writes the new zoom factor, and then returns the zoom factor. This can be used by the application to verify the zoom factor.

Note *Zoom factor should only be changed during vertical blank.*

Returned Values

Current X Zoom factor.

Hardware Modified

IMS_DISPXZ, IMS_DISPYZ

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

ims_dzoomxy — Set Frame B1 X and Y Zoom Factors for Display

Syntax void **ims_dzoomxy**(MODCNF **ims_ptr*, short *xzoom*, short *yzoom*)

ims_ptr Pointer to the IMS.

xzoom Horizontal zoom factor:
1 = normal display (zoom by 1)
2 — Zoom by 2
4 — Zoom by 4

yzoom Vertical zoom factor:
1 = normal display (zoom by 1)
2 — Zoom by 2
4 — Zoom by 4

Description

The function **ims_dzoomxy** applies the zoom factor *xzoom* to frame B1 in the X (horizontal) direction, and *yzoom* in the Y (vertical) direction. This function does not return zoom value.

Note *Zoom factor should only be changed during vertical blank.*

Hardware Modified

IMS_DISPXZ, IMS_DISPYZ

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

ims_freeze — Stop Acquisition

Syntax short **ims_freeze**(MODCNF **ims_ptr*, short *frame*)

ims_ptr Pointer to the IMS.

frame Frame to stop the acquisition:
IMS_FRAMEA — Frame A, 16–bit frame.
IMS_FRAMEA0 — Frame A0, 8–bit frame.
IMS_FRAMEA1 — Frame A1, 8–bit frame.
IMS_FRAMEB1 — Frame B1, 8–bit frame.

Description

This function stops acquisition into *frame* only. Acquisition into other frames continues. The function **ims_grab** startsa continuous acquisition.

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid acquisition buffer.
ITX_TIMEOUT — Can not stop acquisition for the specified frame.

Hardware Modified

IMS_ABTSA0, IMS_ABTSA1, IMS_ABTSB1

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

ims_get_adrform — Return the Address Format

Syntax short **ims_get_adrform**(MODCNF **ims_ptr*)

ims_ptr Pointer to the IMS.

Description

This function returns the address format for the IMS. The address format is either 24 bits, VME standard, or 32 bits, VME extended. The function **ims_set_mapformat** sets this value.

Returned Values

IMS_EXT_MODE — 32–bit addressing mode.
IMS_STD_MODE — 24–bit addressing mode.

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

ims_get_aoi — Return the AOI Dimensions

Syntax short **ims_get_aoi**(MODCNF **ims_ptr*, short **dx*, short **dy*, short **hblen*, short **vblen*, short *aoisel*)

ims_ptr Pointer to the IMS.

dx	Pointer to the horizontal dimension, a returned value. 0 — Do not return this value.
dy	Pointer to the vertical dimension, a returned value. 0 — Do not return this value.
hblen	Pointer to the horizontal length, a returned value. 0 — Do not return this value.
vblen	Pointer to the vertical height, a returned value. 0 — Do not return this value.
aoisel	AOI generator: IMS_AOISEL0 — Select AOI Zero.

Description

This function returns the AOI active and blanking dimensions set by **ims_set_aoi**. If a parameter points to zero or null, the function does not return that parameter's value. This function converts the actual register values to real pixel and line values.

$$x = 4 \times (AOIXS0 + 1)$$

$$y = AOIYS0 + 1$$

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — "Invalid AOI selection" *aoisel* must be IMS_AOISEL0.

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

ims_get_mapadr — Return the VME Memory Map Address

Syntax short **ims_get_mapadr**(MODCNF **ims_ptr*, DWORD **adr*)

ims_ptr Pointer to the IMS.

adr Pointer to the VME memory map address, a returned value.

Description

This function returns the current VME memory base address. The function **ims_set_mapformat** sets this value.

Error Values

IMS_NO_ERROR — Operation successfully completed.

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

ims_get_mod — Return a Pointer to an IMS

Syntax MODCNF ***ims_get_mod**(short *seqnum*)

seqnum Sequence number of the IMS to return.

Description

This functions returns a pointer to the IMS number *seqnum*. The sequence number indicates a specific module when more than one of that module type is in the system.

Returned Values

Pointer to the IMS.

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

ims_get_rdroam — Return the Read Pan and Scroll Shadow Registers

Syntax short **ims_get_rdroam**(MODCNF **ims_ptr*, short *frame*, short **p*, short **s*)

ims_ptr Pointer to the IMS.

frame Frame to set pan and scroll values:
IMS_FRAMEA — Frame A, 16–bit frame.
IMS_FRAMEA0 — Frame A0, 8–bit frame.
IMS_FRAMEA1 — Frame A1, 8–bit frame.
IMS_FRAMEB1 — Frame B1, 8–bit frame.

p Read pan value in the range of 0 to 1023.

s Read scroll value in the range of:
0 to 1023 for the 3MB IMS,
0 to 511 for the 1.5MB IMS.

Description

This function returns the read pan and scroll shadow reigsters for one frame on the IMS. Each frame on the IMS has read pan and scroll values and write pan and scroll values. The read values are used when pixel values are read from the frame.

Returned Values

IMS_NO_ERROR — Operation successfully completed.

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

ims_get_slot0 — Return a Pointer to a DM

Syntax MODCNF ***ims_get_slot0**(MODCNF **ims_ptr*)

ims_ptr Pointer to the IMS containing the display module (DM).

Description

This function returns a pointer to the DM on the IMS. The DM must be in slot 0 on the IMS.

Returned Values

Pointer to a DM.

Error Values

ITX_BAD_MODCNF
IMS_BAD_DMODCNF

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

ims_get_slot1 — Return a Pointer to a CM

Syntax MODCNF ***ims_get_slot1**(MODCNF **ims_ptr*)

ims_ptr Pointer to the IMS containing the computational module (CM).

Description

This function returns a pointer to the CM on the IMS. The CM must be in slot 1 on the IMS.

Returned Values

Pointer to a CM.

Error Values

ITX_BAD_MODCNF
IMS_BAD_DMODCNF

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

ims_get_slot2 — Return a Pointer to an AM

Syntax MODCNF ***ims_get_slot2**(MODCNF **ims_ptr*)

ims_ptr Pointer to IMS containing the AM.

Description

This function returns a pointer to the AM on the IMS. The AM must be located on slot 2 on the IMS.

Returned Values

Pointer to an AM.

Error Values

ITX_BAD_MODCNF
IMS_BAD_DMODCNF

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

ims_get_wrroam — Return the Write Pan and Scroll Shadow Registers

Syntax short **ims_get_wrroam**(MODCNF **ims_ptr*, short *frame*, short **p*, short **s*)

ims_ptr Pointer to the IMS.

frame Frame to set pan and scroll values:
IMS_FRAMEA — Frame A, 16–bit frame.
IMS_FRAMEA0 — Frame A0, 8–bit frame.
IMS_FRAMEA1 — Frame A1, 8–bit frame.
IMS_FRAMEB1 — Frame B1, 8–bit frame.

p Write pan value in the range of 0 to 1023.

s Write scroll value in the range of:
0 to 1023 for the 3MB IMS,
0 to 511 for the 1.5MB IMS.

Description

This function returns the write pan and scroll shadow registers for one frame on the IMS. Each frame on the IMS has read pan and scroll values and write pan and scroll values. The write values are used when pixel values are written to the frame.

Returned Values

IMS_NO_ERROR — Operation successfully completed.

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

ims_grab — Acquire Multiple Images

Syntax short **ims_grab**(MODCNF **ims_ptr*, short *frame*)

ims_ptr Pointer to the IMS.

frame Frame to contain the acquisition:
IMS_FRAMEA0 — Frame A0.
IMS_FRAMEA1 — Frame A1.
IMS_FRAMEB1 — Frame B1.

Description

This function continuously acquires eight–bit images into *frame*. Before calling this function, an application must set–up the cross–port switch with the function **ims_set_cpspath** and the pan and scroll registers with **ims_set_wrroam**. The function **ims_freeze** stops this acquisition.

ims_grab waits for a vertical blank, performs a psreset, then starts acquiring image data continuously.

The function **ims_grab** uses a continuously running clock; whereas, the function **ims_aoi_grab** uses the AOI generator that can be started and stopped to begin the acquisition.

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid acquisition buffer.
ITX_TIMEOUT — VB status bit stuck.

Hardware Modified

IMS_ABTSA0, IMS_ABTSA1, IMS_ABTSB1

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

ims_iregs — Initialize the Registers

Syntax void **ims_iregs**(MODCNF **ims_ptr*)

ims_ptr Pointer to the IMS.

Description

This function initializes the IMS registers to a predefined state. Use the function **ims_dregs** to display the initial register values.

Hardware Modified

All the IMS registers

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

ims_mapsize — Set or Return the Memory Map Size

Syntax short **ims_mapsize**(MODCNF **ims_ptr*, short *mapsel*)

ims_ptr Pointer to the IMS.

mapsel Memory map size:
IMS_1MM — Set a one–megabyte memory block.
IMS_2MM — Set a two–megabyte memory block.
IMS_4MM — Set a four–megabyte memory block.
INQUIRE — Return the current value.

Description

This function defines the size of the block mapped into VME memory address space. The minimum map size supported is 1MB. If a memory size less than 4MB is selected, the entire memory can still be accessed using the Page register. This mechanism moves the memory–mapped window through the IMS address space. The Page register PG(1–0) bits substitute for upper memory address bits when the map size is less than 4MB.

The IMS contains 1.5MB or 3MB of image memory and three memory–mapped modules or daughter boards. The modules are accessed through the IMS memory access space. A page select bit, PGSEL, selects between IMS memory and daughter board access.

This function changes the IMS module structure from its previous value.

Returned Values

Current map size.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_MAPSEL

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

ims_pixsz — Set or Return the Pixel Size

Syntax short **ims_pixsz**(MODCNF **ims_ptr*, short *pixsz*)

ims_ptr Pointer to the IMS.

pixsz Pixel size:
IMS_PIX8 — 8–bit pixel size.
IMS_PIX16 — 16–bit pixel size.
INQUIRE — Return the current value.

Description

This function sets the pixel size for the IMS. This function affects frames A0 and A1 only. With an eight–bit pixel, an application can use all three frames: A0, A1, and B1. With a 16–bit pixel, an application can only frame A for 16–bit operations.

This function changes the IMS module structure from its previous value.

Returned Values

Current pixel size.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_PIXSZ

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

ims_psreset — Set the Pan and Scroll Registers

Syntax void **ims_psreset**(MODCNF **ims_ptr*)

ims_ptr Pointer to the IMS.

Description

This function writes the Read and Write pan and scroll shadow registers for all three frames, from the IMS structure to the appropriate registers. Normally these values get written during a vertical blank. Vertical blanks can occur during image acquisition.

Hardware Modified

IMS_RPANA0, IMS_RPANA1, IMS_RPANB1, IMS_RSCRA0, IMS_RSCRA1, IMS_RSCRB1, IMS_WPANA0, IMS_WPANA1, IMS_WPANB1, IMS_WSRA0, IMS_WSCRA1, IMS_WSCRB1

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

ims_rhbline — Read a Horizontal Line of BYTEs

Syntax void **ims_rhbline**(DWORD *address*, short *num_ops*, short *freq*, BYTE **array*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to read.
freq	Frequency with which to read, from 0 to 32767:
0	Read the same location.
1	Read every location.
2	Read every other location.
3	Read every third location.
. . .	
32,767	Read every 32,768th location.
array	Pointer to the array to store the pixel values.

Description

This function reads a horizontal line of BYTE values. You must select values for *num_ops* and *freq* that do not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

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

ims_rhdwline — Read a Horizontal Line of DWORD Values

Syntax void **ims_rhdwline**(DWORD *address*, short *num_ops*, short *freq*, DWORD **array*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to read.
freq	Frequency with which to read, from 0 to 32767:
	0 Read the same location.
	1 Read every location.
	2 Read every other location.
	3 Read every third location.
	. . .
	32767 Read every 32,768th location.
array	Pointer to the array to store the pixel values.

Description

This function reads a horizontal line of DWORD values. You must select values for *num_ops* and *freq* that do not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

The function **ims_pixsz** defines the pixel size. Each DWORD read by **ima_rhdwline** contains either two horizontally adjacent 16–bit pixels, or four horizontally adjacent 8–bit pixels.

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

ims_rhwline — Read a Horizontal Line of WORD Values

Syntax void **ims_rhwline**(DWORD *address*, short *num_ops*, short *freq*, WORD **array*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to read.
freq	Frequency with which to read, from 0 to 32767:
	0 Read the same location.
	1 Read every location.
	2 Read every other location.
	3 Read every third location.
	. . .
	32,767 Read every 32,768th location.
array	Pointer to the array to store the pixel values.

Description

This function reads a horizontal line of WORD values. You must select values for *num_ops* and *freq* that do not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

The function **ims_pixsz** defines the pixel size. Each WORD read by **ims_rhwline** contains either one 16–bit pixel or two horizontally adjacent 8–bit pixels.

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

ims_rpix — Read a Single Pixel Value

Syntax WORD **ims_rpix**(MODCNF **ims_ptr*, short *frame*, short *x*, short *y*)

ims_ptr Pointer to the IMS.

frame	Frame to access: IMS_FRAMEA0 — Frame A0. IMS_FRAMEA1 — Frame A1. IMS_FRAMEB1 — Frame B1.
x	X location to read.
y	Y location to read.

Description

This function reads a single 8–bit pixel value at location (*x,y*) in *frame*. This function uses the current pixel size.

Note *This function internally calls* **ims_set_lineio***, and may change the value of the address pointer from a previous value.*

Returned Values

Eight–bit pixel value.

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

ims_rvbline — Read a Vertical Line of BYTE Values

Syntax void **ims_rvbline**(DWORD *address*, short *num_ops*, short *linesz*, BYTE **array*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to read.
linesz	Horizontal frame size returned from **ims_set_lineio**.
array	Pointer to the array to store the pixel values.

Description

This function reads a vertical line of BYTE values. You must select a value for that does not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

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

ims_rvdwline — Read a Vertical Line of DWORD Values

Syntax void **ims_rvdwline**(DWORD *address*, short *num_ops*, short *linesz*, DWORD **array*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to read.
linesz	Horizontal frame size returned from **ims_set_lineio**.
array	Pointer to the array to store the pixel values.

Description

This function reads a vertical line of DWORD values. You must select a value for *num_ops* that does not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

The function **ims_pixsz** defines the pixel size. Each DWORD read by **ims_rvdwline** contains either two horizontally adjacent 16–bit pixels or four horizontally adjacent 8–bit pixels.

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

ims_rvwline — Read a Vertical Line of WORD Values

Syntax void **ims_rvwline**(DWORD *address*, short *num_ops*, short *linesz*, WORD **array*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to read.
linesz	Horizontal frame size returned from **ims_set_lineio**.
array	Pointer to the array to store the pixel values.

Description

This function reads a vertical line of WORD values. You must select a value for *num_ops* that does not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

The function **ims_pixsz** defines the pixel size. Each WORD read by **ims_rvwline** contains either one 16–bit pixels or two horizontally adjacent 8–bit pixels.

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

ims_sel_slave — Select or Return a Slave to Output to the Video Bus

Syntax short **ima_sel_slave**(MODCNF **ims_ptr*, short *sel*)

ims_ptr Pointer to the IMS.

sel Slave status:
IMS_MASTER — The master outputs to the video bus.
IMS_SLAVE0 — Slave 0 outputs to the video bus.
IMS_SLAVE1 — Slave 1 outputs to the video bus.
IMS_SLAVE2 — Slave 2 outputs to the video bus.
INQUIRE — Return the current value.

Description

The IMS is always the bus master. A system can contain up to three slave IMA modules. Only the master may select a slave to output to the video bus. This function changes the IMS module structure from its previous value.

Returned Values

Current status.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_SLAVSEL

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

ims_set_aoi — Set the AOI Dimensions

Syntax short **ims_set_aoi**(MODCNF **ims_ptr*, short *dx*, short *dy*, short *hblen*, short *vblen,* short *aoisel*)

ims_ptr Pointer to the IMS.

dx	Horizontal dimension from 4 to 41024 in multiples of four.
dy	Vertical dimension: from 1 to 1024 for the 3MB IMS, or from 1 to 512 for the 1.5MB IMS.
hblen	Horizontal blank length from 24 to 256 in multiples of eight.
vblen	Vertical blank length from 2 to 64 in multiples of two.
aoisel	AOI timing generator: IMS_AOISEL0 — Select AOI generator Zero.

Description

This function sets the horizontal and vertical dimensions (active size and blanking lengths) of the specified AOI. If a parameter exceeds its limit, it is set to its maximum value. If a parameter is less than its limit, it is set to its minimum value.

The horizontal active size is programmable on boundaries of four pixels. New horizontal parameters are loaded at the end of the current line. The vertical size is programmable on boundaries of one line. New vertical parameters are loaded at the end of the current frame.

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — "Invalid AOI selection" *aoisel* must be IMS_AOISEL0

Hardware Modified

IMS_AOIXS0, IMS_AOIYS0.

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

ims_set_cpspath — Set the Cross–port Switch

Syntax short **ims_set_cpspath**(MODCNF **ims_ptr*, short *spath*, short *dpath*)

ims_ptr Pointer to the IMS.

spath Source path:

IMS_VB0 — Video bus 0. (Not a valid source for destinations: IMS_CM1, IMS_CM2, IMS_CM3)

IMS_VB1 — Video bus 1. (Not a valid source for destinations: IMS_CM2, IMS_CM3)

IMS_VB2 — Video bus 2. (Not a valid source for destinations: IMS_CM1, IMS_CM3)

IMS_VP0 — Video bus pipeline–in 0.

IMS_VP1 — Video bus pipeline–in 1.

IMS_VP2 — Video bus pipeline–in 2.

IMS_FA0 — Frame A0 output.

IMS_FA1 — Frame A1 output.

IMS_FB1 — Frame B1 output.

IMS_CM0 — CM output port 0. (Not a valid source for destinations: IMS_CM1, IMS_CM2, IMS_CM3)

IMS_CM1 — CM output port 1. (Not a valid source for destinations IMS_CM0, IMS_CM2, IMS_CM3)

IMS_CM2 — CM output port 2. (Not a valid for destinations: IMS_CM0, IMS_CM1, IMS_CM3)

IMS_AM0 — AM output port 0. (IMS has only one 8–bit AM input)

IMS_CLR — Clear the destination to zero. Valid for all destinations.

IMS_SET — Set the destination to 0xFFFF. Valid for all destinations.

IMS_K — Constant value K, programmable from 0 to 0xFF. Valid for all destinations.

dpath Destination path:

IMS_VB0 — Video bus 0.

IMS_VB1 — Video bus 1.

IMS_VB2 — Video bus 2.

IMS_FA0 — Frame A0 output.

IMS_FA1 — Frame A1 output.

IMS_FB1 — Frame B1 output.

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

IMS_CM0 — CM output port 0. (Not a valid destination for these sources: IMS_CM1, IMS_CM2, IMS_CM3, IMS_VB1, IMS_VB2, IMS_VB3.)

IMS_CM1 — CM output port 1. (Not a valid destination for these sources: IMS_VB0, IMS_VB2)

IMS_CM2 — CM output port 2. (Not a valid destination for these sources: IMS_VB0, IMS_VB1)

IMS_CM3 — CM output port 3. (Not a valid destination for these sources: IMS_VB0, IMS_VB1, IMS_VB2, IMS_VP0, IMS_VP1, IMS_VP2, IMS_CM0, IMS_CM1, IMS_CM2, IMS_CM3)

Description

This function sets the two cross–port switches that define the input and output paths and timing through the 150/40 system. The following restrictions apply:

- IMS_FA0, IMS_FA1, IMS_FB1, IMS_SET, IMS_CLR, and IMS_K are the only valid sources for a destination of IMS_CM3.

Note *There is no ims_set_cpstiming function. The function ims_set_cpspath sets the CPS timing to the source spath. There is a non–functioning stub ims_set_cpstiming that returns IMS_NO_ERROR, to prevent software from crashing.*

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid destination path.

Hardware Modified

IMS_PVB0, IMS_PVB1, IMS_PVB2, IMS_PVB3, IMS_FA0SEL, IMS_FA1SEL, IMS_FB1SEL, IMS_CMBPY

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

ims_set_lineio — Set–up Frame Buffer for Access

Syntax short **ims_set_lineio**(MODCNF **ims_ptr*, short *frame*, short *x*, short *y*, short **xsz*, DWORD **offadr*)

ims_ptr Pointer to the IMS.

frame	Frame buffer to access on the IMS: IMS_FRAMEA — Frame A, 16–bit frame. IMS_FRAMEA0 — Frame A0, 8–bit frame. IMS_FRAMEA1 — Frame A1, 8–bit frame. IMS_FRAMEB1 — Frame B1, 8–bit frame.
x	Horizontal location to start, relative to the upper–left corner of the frame.
y	Vertical location to start, relative to the upper–left corner of the frame.
xsz	Pointer to the horizontal–line size, a returned value.
offadr	Pointer to the memory address, a returned value.

Description

This function sets the page register relative to the selected frame plus the memory size enabled in host address space. It then adjusts the X,Y coordinates relative to the page register value and calculates an address to return. This function must be called before any other line read/write/clear function (refer to Table 1–5, page 1–17). This function must also be called when you need to write to a different frame or change the location you are accessing.

Subsequent line access uses *offadr* to specify where data is accessed, and the position can be changed by adding *xsz* to *offadr*. You can add *xsz* to *offadr*, but you cannot subtract.

Note *Each time you use functions that change the host access map, you must call this function again before calling the line read/write/clear functions. For example, any DM-PC or DPX function, or line read/write/clear functions on another module (IMA, IMS, DPX, CM-MEM) changes the host mapping. Failure to call ims_set_lineio will cause line access to the other module, not the IMS.*

Error Values

IMS_NO_ERROR — Operation successfully completed.

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

ims_set_mapformat — Set the Memory Access Format and the Memory Base Address

Syntax short **ims_set_mapformat**(MODCNF **ims_ptr*, short *format*, DWORD *adr*)

ims_ptr Pointer to the IMS.

format Memory map format:
IMS_EXT_MODE — 32–bit addressing mode.
IMS_STD_MODE — 24–bit addressing mode.

adr VME memory address.

Description

The two bits in the configuration register (IMS_CONFIG) select the address size (EXTEN, STDEN) either standard (A24) or extended (A32) addressing. In standard A24 addressing mode, the board responds to address modifiers 3DH and 39H. In extended A32 addressing mode, the board responds to address modifiers 0DH and 09H. These bits are qualified with the MEMEN bit in the Page register.

The function **ims_get_adrform** returns the memory map format. The function **ims_get_mapadr** returns the VME physical memory address.

The addressing format is not defined at power–up. This register must be properly initialized before setting the bit MEMEN in the Page register.

This function changes the IMS module structure from its previous value

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_STDSEL, IMS_EXTSEL, IMS_STDADR, IMS_EXTADR

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

ims_set_rdroam — Set the Read Pan and Scroll Shadow Registers

Syntax short **ims_set_rdroam**(MODCNF **ims_ptr*, short *frame*, short *p*, short *s*)

ims_ptr Pointer to the IMS.

frame	Frame to set pan and scroll values: IMS_FRAMEA — Frame A, 16–bit frame. IMS_FRAMEA0 — Frame A0, 8–bit frame. IMS_FRAMEA1 — Frame A1, 8–bit frame. IMS_FRAMEB1 — Frame B1, 8–bit frame.
p	Read pan value in the range 0 to 1023.
s	Read scroll value in the range 0 to 1023 for the 3MB IMS, or 0 to 511 for the 1.5MB IMS.

Description

This function sets the Read pan and scroll shadow registers for one *frame* on the IMS. Each *frame* on the IMS has Read pan and scroll values and Write pan and scroll values. The Read values are used when pixel values are read from the frame.

If *p* exceeds its limit, the pan value wraps from the right edge of the screen to the left. If *s* exceeds its limit, the scroll value wraps from the bottom to the top of the screen.

The pan and scroll shadow values are written into the hardware registers during a vertical blank or by calling **ims_psreset**. This function changes the IMS module structure from its previous value

Returned Values

IMS_NO_ERROR — Operation successfully completed.
ITX_GEN_ERROR — Pan or scroll value too large.
IMS_BAD_ARG — Invalid *frame*

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

ims_set_wrroam — Set the Write Pan and Scroll Shadow Registers

Syntax short **ims_set_wrroam**(MODCNF **ims_ptr*, short *frame*, short *p*, short *s*)

ims_ptr Pointer to the IMS.

frame	Frame to set pan and scroll values: IMS_FRAMEA — Frame A, 16–bit frame. IMS_FRAMEA0 — Frame A0, 8–bit frame. IMS_FRAMEA1 — Frame A1, 8–bit frame. IMS_FRAMEB1 — Frame B1, 8–bit frame.
p	Write pan value in the range 0 to 1023.
s	Write scroll value in the range 0 to 1023 for 3MB IMS or 0 to 511 for the 1.5MB IMS.

Description

This function sets the Write pan and scroll shadow registers for one *frame* on the IMS. Each frame on the IMS has Read pan and scroll values and Write pan and scroll values. The Write values are used when pixel values are written to the frame.

If *p* exceeds its limit, the pan value wraps from the right edge of the screen to the left. If *s* exceeds its limit, the scroll value wraps from the bottom to the top of the screen.

The pan and scroll shadow values are written into the hardware registers during a vertical blank or by calling **ims_psreset**. This function changes the IMS module structure from its previous value

Returned Values

IMS_NO_ERROR — Operation successfully completed.
ITX_GEN_ERROR — Pan or scroll value too large.
IMS_BAD_ARG — Invalid *frame*

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

ims_snap — Acquire a Single Image

Syntax short **ims_snap**(MODCNF **ims_ptr*, short *frame*)

ims_ptr Pointer to the IMS.

frame Frame to contain the acquisition:
IMS_FRAMEA0 — Frame A0.
IMS_FRAMEA1 — Frame A1.
IMS_FRAMEB1 — Frame B1.

Description

This function acquires a single eight–bit image into *frame*. **ims_snap** waits for a vertical blank, performs a psreset, then starts acquiring a single frame of data. Before calling this function, an application must set–up the cross–port switch with the function **ims_set_cpspath** and the Write Pan and Scroll registers with **ims_set_wrroam**.

The function **ims_snap** uses a continuously running clock; whereas, the function **ims_aoi_snap** uses the AOI generator that can be started and stopped to begin the acquisition.

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid acquisition buffer.
ITX_TIMEOUT — VB status bit stuck.

Hardware Modified

IMS_ABTSA0, IMS_ABTSA1, IMS_ABTSB1

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

ims_snap_async — Acquire a Single 8-bit Image Asynchronously

Syntax short **ims_snap_async**(MODCNF **ims_ptr*, short *frame*)

ims_ptr Pointer to the IMS.

frame Frame to contain the acquisition:
IMS_FRAMEA0 — Frame A0.
IMS_FRAMEA1 — Frame A1.
IMS_FRAMEB1 — Frame B1.

Description

The function **ims_snap_async** acquires a single eight–bit image into *frame*. ims_snap waits for a vertical blank, performs a psreset, then starts acquiring a single frame of data. ims_snap_async returns imediately. Before calling this function, an application must set–up the cross–port switch with the function ims_set_cpspath and the Write Pan and Scroll registers with ims_set_wrroam.

The function ims_snap uses a continuously running clock; whereas, the function ims_aoi_snap uses the AOI generator that can be started and stopped to begin the acquisition.

Note ims_snap *starts the acquisiton and polls until the acquisition is complete.* ims_snap_async *starts the acquisition and returns immediately, without waiting for the acquistiion to complete. Use* ima_snap_async *with the interrupts API.*

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid acquisition buffer.
ITX_TIMEOUT — VB status bit stuck.

Hardware Modified

IMS_ABTSA0, IMS_ABTSA1, IMS_ABTSB1

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

ims_vbusmode — Set the Video Bus Timing

Syntax short **ims_vbusmode**(MODCNF **ims_ptr*, short *modesel*)

ims_ptr Pointer to the IMS.

modesel Video bus timing:

IMS_VBUSAOI — Set video bus timing to AOI generator timing.

IMS_VBUSCAM — Set video bus timing to camera timing.

INQUIRE — Return the current timing.

Description

This functions sets the video bus timing to either the camera or AOI generator timing. This function changes the IMS module structure from its previous value.

Returned Values

Current timing.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_AOIMODE

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

ims_wait_notvb — Wait Until Vertical Blank is Past

Syntax short **ims_wait_notvb**(MODCNF **ims_ptr*, short *frame*, short *vbsync*)

ims_ptr Pointer to the IMS.

frame Frame to wait for not vertical blank:
IMS_FRAMEA0 — Frame A0.
IMS_FRAMEA1 — Frame A1.
IMS_FRAMEB1 — Frame B1.

vbsync Synchronization mode:
ON — Wait for a vertical blank before waiting for not vertical blank.
OFF — Do not wait for vertical blank.

Description

This function waits until *frame* is not in vertical blank.

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid acquisition buffer.
ITX_TIMEOUT — VB status bit stuck.

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

ims_wait_vb — Wait for Next Vertical Blank

Syntax short **ims_wait_vb**(MODCNF **ims_ptr*, short *frame*)

ims_ptr Pointer to the IMS.

frame Frame to synchronize:
IMS_FRAMEA0 — Frame A0.
IMS_FRAMEA1 — Frame A1.
IMS_FRAMEB1 — Frame B1.

Description

This function does a busy wait until the beginning of the next vertical blanking interval for the camera for the selected frame.

Error Values

IMS_NO_ERROR — Operation successfully completed.
IMS_BAD_ARG — Invalid acquisition buffer.
ITX_TIMEOUT — VB status bit stuck.

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

ims_waitacq — Wait for Acquisition to Complete

Syntax short **ims_waitacq**(MODCNF **ims_ptr*, short *frame*)

ims_ptr Pointer to the IMS.

frame Frame to synchronize:
IMS_FRAMEA0 — Frame A0.
IMS_FRAMEA1 — Frame A1.
IMS_FRAMEB1 — Frame B1.

Description

This function waits for completion of the current acquisition in *frame*. If the current acquisition is continuous, the function returns immediately the symbol ITX_TIMEOUT; otherwise, the function returns the symbol ITX_NO_ERROR.

Error Values

ITX_NO_ERROR — Acquisition completed.
ITX_TIMEOUT — Acquisition is continuous.
IMS_BAD_ARG — Invalid value for *frame*.

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

ims_whbline — Write a Horizontal Line of BYTEs

Syntax void **ims_whbline**(DWORD *address*, short *num_ops*, short *freq*, BYTE **array*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to write.
freq	Frequency with which to write, from 0 to 32767:
	0 Write the same location.
	1 Write every location.
	2 Write every other location.
	3 Write every third location.
	. . .
	32,767 Write every 32,768th location.
array	Pointer to the array containing the values to write.

Description

This function writes a horizontal line of BYTE values. You must select values for *num_ops* and *freq* that do not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

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

ims_whdwline — Write a Horizontal Line of DWORD Values

Syntax void **ims_whdwline**(DWORD *address*, short *num_ops*, short *freq*, DWORD **array*)

address The memory address returned from **ims_set_lineio**.

num_ops Number of locations to write.

freq Frequency with which to write, from 0 to 32767:

0	Write the same location.
1	Write every location.
2	Write every other location.
3	Write every third location.
. . .	
32767	Write every 32,768th location.

array Pointer to the array containing the values to write.

Description

This function writes a horizontal line of DWORD values. You must select values for *num_ops* and *freq* that do not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

The function **ims_pixsz** defines the pixel size. Each DWORD written by **ims_whdwline** contains either two horizontally adjacent 16–bit pixels or four horizontally adjacent 8–bit pixels.

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

ims_whwline — Write a Horizontal Line of WORD Values

Syntax void **ims_whwline**(DWORD *address*, short *num_ops*, short *freq*, WORD **array*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to write.
freq	Frequency with which to write, from 0 to 32767:
	0 Write the same location.
	1 Write every location.
	2 Write every other location.
	3 Write every third location.
	. . .
	32,767 Write every 32,768th location.
array	Pointer to the array containing the values to write.

Description

This function writes a horizontal line of WORD values. You must select values for *num_ops* and *freq* that do not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

The function **ims_pixsz** defines the pixel size. Each DWORD written by **ims_whwline** contains either one 16–bit pixel or two horizontally adjacent 8–bit pixels.

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

ims_wpix — Write a Single Pixel Value

Syntax void **ims_wpix**(MODCNF **ims_ptr*, short *frame*, short *x*, short *y*, WORD *pv*)

ims_ptr Pointer to the IMS.

frame	Frame to write in: IMS_FRAMEA0 — Frame A0. IMS_FRAMEA1 — Frame A1. IMS_FRAMEB1 — Frame B1.
x	X location to write.
y	Y location to write.
pv	Eight–bit pixel value to write.

Description

This function writes a single eight–bit pixel value at location (*x,y*) in *frame*. This function uses the current pixel size.

Note *This function internally calls* **ims_set_lineio***, and may change the value of the address pointer from a previous value.*

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

ims_wvbline — Write a Vertical Line of BYTE Values

Syntax void **ims_wvbline**(DWORD *address*, short *num_ops*, short *linesz*, BYTE **array*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to write.
linesz	Horizontal frame size returned from **ims_set_lineio**.
array	Pointer to the array containing the values to write.

Description

This function writes a vertical line of BYTE values. You must select a value for *num_ops* that does not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

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

ims_wvdwline — Write a Vertical Line of DWORD Values

Syntax void **ims_wvdwline**(DWORD *address*, short *num_ops*, short *linesz*, DWORD **array*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to write.
linesz	Horizontal frame size returned from **ims_set_lineio**.
array	Pointer to the array containing the values to write.

Description

This function writes a vertical line of DWORD values. You must select a value for *num_ops* that does not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

The function **ims_pixsz** defines the pixel size. Each DWORD written by **ims_wvdwline** contains either two horizontally adjacent 16–bit pixels or four horizontally adjacent 8–bit pixels.

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

ims_wvwline — Write a Vertical Line of WORD Values

Syntax void **ims_wvwline**(DWORD *address*, short *num_ops*, short *freq*, WORD **array*)

address	The memory address returned from **ims_set_lineio**.
num_ops	Number of locations to write.
linesz	Horizontal frame size returned from **ims_set_lineio**.
array	Pointer to the array containing the values to write.

Description

This function writes a vertical line of WORD values. You must select a value for *num_ops* that does not exceed the size of frame memory. The function **ims_set_lineio** must be called before calling this function.

The function **ims_pixsz** defines the pixel size. Each WORD written by **ims_wvwline** contains either one 16–bit pixel or two horizontally adjacent 8–bit pixels.

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

INDEX

A

acquire, begin delayed, **ims_delay_acqen**, 2–25

acquisition, 1–3
- control, 1–6
- multiple images
  - **ims_aoi_grab**, 2–4
  - **ims_aoi_grab16**, 2–5
  - **ims_grab**, 2–44
- single image
  - **ims_aoi_snap**, 2–7
  - **ims_aoi_snap16**, 2–9
  - **ims_snap**, 2–68
- stopping, **ims_freeze**, 2–34
- waiting for completion, **ims_waitacq**, 2–72
- zoom, 1–6

acquisition command, **ims_acqbits**, 2–2

address format, **ims_get_adrform**, 2–35

ALU, 1–3

AM
- interface, 1–12
- returning pointer to, **ims_get_slot2**, 2–42
- timing, 1–9

AOI
- returning dimensions, **ims_get_aoi**, 2–36
- ROI versus AOI, 1–16
- settting dimension, **ims_set_aoi**, 2–60

AOI timing, 1–16

AOI timing generator, 1–9
- acquiring images
  - **ims_aoi_grab**, 2–4
  - **ims_aoi_grab16**, 2–5
  - **ims_aoi_snap**, 2–7
  - **ims_aoi_snap16**, 2–9
- halting, **ims_aoi_halt**, 2–6
- **ims_vbusmode**, 2–69
- starting, **ims_aoi_start**, 2–11
- waiting for AOI ready bits, **ims_aoi_1shot**, 2–3

area
- clearing, **ims_clr_area**, 2–17
- copying, **ims_cp_area**, 2–19
- displaying, **ims_dump_mem**, 2–28

B

bit mask, **ims_bitmask**, 2–12

BYTEs, clearing
- horizontal line, **ims_chbline**, 2–13
- vertical line, **ims_cvbline**, 2–21

BYTEs, reading
- horizontal line, **ims_rhbline**, 2–52
- vertical line, **ims_rvbline**, 2–56

BYTEs, writing
- horizontal line, **ims_whbline**, 2–73
- vertical line, **ims_wvbline**, 2–77

C

clearing
- area, **ims_clr_area**, 2–17
- frame, **ims_clr_frame**, 2–18

clearing horizontal line
- **ims_chbline**, 2–13
- **ims_chdwline**, 2–14
- **ims_chwline**, 2–15

clearing vertical line
- **ims_cvbline**, 2–21
- **ims_cvwline**, 2–23
- **ims_dvdwline**, 2–22

clock speed, **ims_clkspeed**, 2–16

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

clock spped, 1–8
See also frame timing

CM
interface, 1–12
returning pointer to, **ims_get_slot1**, 2–41
timing, 1–9

compare and contrast, image managers, 1–2

copying, area, **ims_cp_area**, 2–19

CPS, 1–7
ims_set_cpspath, 2–61
setting up, 1–9

cross–port switch. *See* CPS

D

data paths, **ims_set_cpspath**, 2–61

daughter board
enabling interrupts, **ims_dbrd_intr**, 2–24
returning pointer to
ims_get_slot0, 2–40
ims_get_slot1, 2–41
ims_get_slot2, 2–42

delayed acquire, begin, **ims_delay_acqen**, 2–25

display, zoom
ims_dzoom, 2–29
ims_dzoomxy, 2–30

display enable, 1–4

display mode, **ims_dispmode**, 2–26

displaying
area, **ims_dump_mem**, 2–28
registers, **ims_dregs**, 2–27

DM
interface, 1–12
returning pointer to, **ims_get_slot0**, 2–40
timing, 1–9

dumping. *See* displaying

DWORD, clearing
horizontal line, **ims_chdwline**, 2–14
vertical line, **ims_cvdwline**, 2–22

DWORD, reading
horizontal line, **ims_rhdwline**, 2–53
vertical line, **ims_rvdwline**, 2–57

DWORD, writing
horizontal line, **ims_whdwline**, 2–74
vertical line, **ims_wvdwline**, 2–78

E

eight–bit images, acquisition
ims_aoi_grab, 2–4
ims_aoi_snap, 2–7
ims_grab, 2–44
ims_snap, 2–68

external trigger interrupt, **ims_ext_intr**, 2–31

F

features, image managers, compare and contrast, 1–2

frame, 1–4
clearing, **ims_clr_frame**, 2–18
enabling interupts, **ims_frame_intr**, 2–32
setting up for access, **ims_set_lineio**, 2–63
timing, 1–9

freezing. *See* stopping

G

global bus, 1–11

grabbing an image, **ims_grab**, 2–44

H

halting
See also stopping
AOI generator, **ims_aoi_halt**, 2–6

horizontal, zoom factor, **ims_dzoomxy**, 2–30

horizontal line, clearing
ims_chbline, 2–13
ims_chdwline, 2–14
ims_chwline, 2–15

horizontal line, reading
ims_rhbline, 2–52
ims_rhdwline, 2–53
ims_rhwline, 2–54

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

horizontal line, writing
ims_whbline, 2–73
ims_whdwline, 2–74
ims_whwline, 2–75

I

IMA, slave selection, **ima_sel_slave**, 2–59

ima_sel_slave, 2–59

image managers, compare and contrast, 1–2

image memory, 1–4

IMS, returning pointer to, **ims_get_mod**, 2–38

ims_acqbits, set or return acquisition command bits, 2–2

ims_aoi_1shot, 2–3

ims_aoi_grab, 2–4

ims_aoi_grab16, 2–5

ims_aoi_halt, 2–6

ims_aoi_snap, 2–7

ims_aoi_snap16, 2–9

ims_aoi_start, 2–11

ims_bitmask, 2–12

ims_chbline, 2–13

ims_chdwline, 2–14

ims_chwline, 2–15

ims_clkspeed, 2–16

ims_clr_area, 2–17

ims_clr_frame, 2–18

ims_cp_area, 2–19

ims_cvbline, 2–21

ims_cvdwline, 2–22

ims_cvwline, 2–23

ims_dbrd_intr, 2–24

ims_delay_acqen, begin delayed acquire, 2–25

ims_dispmode, 2–26

ims_dregs, 2–27

ims_dump_mem, 2–28

ims_dzoom, zoom display, 2–29

ims_dzoomxy, zoom display, 2–30

ims_ext_intr, 2–31

ims_frame_intr, 2–32

ims_freeze, 2–34

ims_get_adrform, 2–35

ims_get_aoi, return AOI dimensions, 2–36

ims_get_mapadr, 2–37

ims_get_mod, 2–38

ims_get_rdroam, 2–39

ims_get_slot0, 2–40

ims_get_slot1, 2–41

ims_get_slot2, 2–42

ims_get_wrroam, 2–43

ims_grab, 2–44

ims_ien_bits, 2–45

ims_intr_priority, 2–46

ims_intr_vec, 2–47

ims_iregs, 2–48

ims_mapsize, 2–49

ims_pixsz, 2–50

ims_psreset, set/load all pan and scroll registers, 2–51

ims_read_imagearea, relplaced in 2.3 release. *See* system_level software, **itx_read_imagearea**

ims_rhbline, 2–52

ims_rhdwline, 2–53

ims_rhwline, 2–54

ims_rpix, 2–55

ims_rvbline, 2–56

ims_rvdwline, 2–57

ims_rvwline, 2–58

ims_set_aoi, set AOI dimensions, 2–60

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

ims_set_cpspath, 2–61

ims_set_cpstiming, non–functioning stub, see ims_set_cpspath, 2–62

ims_set_lineio, 2–63

ims_set_mapformat, 2–65

ims_set_rdroam, 2–66

ims_set_wrroam, 2–67

ims_snap, 2–68

ims_vbusmode, 2–69

ims_wait_notvb, 2–70

ims_wait_vb, 2–71

ims_waitacq, 2–72

ims_whbline, 2–73

ims_whdwline, 2–74

ims_whwline, 2–75

ims_wpix, 2–76

ims_write_image, replaced in 2.3 release. *See* system–level software, **itx_write_image**

ims_wvbline, 2–77

ims_wvdwline, 2–78

ims_wvwline, 2–79

initializing, registers, **ims_iregs**, 2–48

input path, **ims_set_cpspath**, 2–61

interface, module, 1–11

interrupt, 1–7
- daughter board, **ims_dbrd_intr**, 2–24
- external trigger, **ims_ext_intr**, 2–31
- frame enable, **ims_frame_intr**, 2–32
- priority level, **ims_intr_priority**, 2–46
- setting, **ims_ien_bits**, 2–45
- vector, **ims_intr_vec**, 2–47

L

LUTs, 1–3

M

map address, **ims_get_mapadr**, 2–37

master IMA, **ima_sel_slave**, 2–59

master IMS, 1–11

memory
- image, 1–4
- write masks, 1–7

memory map
- format, **ims_set_mapformat**, 2–65
- size, **ims_mapsize**, 2–49

module interface, 1–11

O

one–shot
- **ims_aoi_1shot**, 2–3
- **ims_aoi_grab16**, 2–5

output path, **ims_set_cpspath**, 2–61

P

pan. *See* read pan and scroll values; write pan and scroll values

pipeline data bus, 1–11

pixel
- reading, **ims_rpix**, 2–55
- size, **ims_pixsz**, 2–50
- writing, **ims_wpix**, 2–76

priority level, interrupt, **ims_intr_priority**, 2–46

R

read pan and scroll values, 1–5
- returning, **ims_get_rdroam**, 2–39
- setting, **ims_set_rdroam**, 2–66
- writing to register, **ims_psreset**, 2–51

read_image, replaced in 2.3 release. *See* system–level software, **itx_read_image**

read_reg, A–1

reading
- bits and registers, A–1
  - *See also* system–level software **read_reg**
- images. *See* system–level software, **itx_read_image**

reading horizontal line
- **ims_rhbline**, 2–52
- **ims_rhdwline**, 2–53
- **ims_rhwline**, 2–54

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

reading pixel, **ims_rpix**, 2–55

reading vertical line
ims_rvbline, 2–56
ims_rvdwline, 2–57
ims_rvwline, 2–58

recalling/restoring, images from files. *See* system–level software
itx_read_image

registers
displaying, **ims_dregs**, 2–27
initializing, **ims_iregs**, 2–48

ROI, ROI versus AOI, 1–16

S

saving, images. *See* system–level software, **itx_write_image**

scan access. *See* read pan and scroll values; write pan and scroll values

scroll. *See* read pan and scroll values; write pan and scroll values

setting up frame, **ims_set_lineio**, 2–63

16–bit images, acquisition
ims_aoi_grab16, 2–5
ims_aoi_snap16, 2–9

slave, selection, **ima_sel_slave**, 2–59

snapping an image, **ims_snap**, 2–68

starting, AOI generator, **ims_aoi_start**, 2–11

stopping
See also halting
acquisition, **ims_freeze**, 2–34

sub–sample acquistion, 1–6

T

timing
frame, 1–9
generators, 1–9
video bus, **ims_vbusmode**, 2–69

V

vertical, zooom factor, **ims_dzoomxy**, 2–30

vertical blank
not waiting for, **ims_wait_notvb**, 2–70
waiting for, **ims_wait_vb**, 2–71

vertical line, clearing
ims_cvbline, 2–21
ims_cvdwline, 2–22
ims_cvwline, 2–23

vertical line, reading
ims_rvbline, 2–56
ims_rvdwline, 2–57
ims_rvwline, 2–58

vertical line, writing
ims_wvbline, 2–77
ims_wvdwline, 2–78
ims_wvwline, 2–79

video bus, 1–11
timing, **ims_vbusmode**, 2–69

VME memory base address, **ims_set_map-format**, 2–65

W

waiting for
acquisition to complete, **ims_waitacq**, 2–72
vertical blank, **ims_wait_vb**, 2–71

WORD, clearing
horizontal line, **ims_chwline**, 2–15
vertical line, **ims_cvwline**, 2–23

WORD, reading
horizontal line, **ims_rhwline**, 2–54
vertical line, **ims_rvwline**, 2–58

WORD, writing
horizontal line, **ims_whwline**, 2–75
vertical line, **ims_wvwline**, 2–79

write masks, 1–7

write pan and scroll values, 1–5
returning, **ims_get_wrroam**, 2–43
setting, **ims_set_wrroam**, 2–67
writing to register, **ims_psreset**, 2–51

write_reg, A–1

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

writing
bits and registers, A–1
See also system–level software, **write_reg**
images. *See* system–level software **itx_write_image**

writing horizontal line
ims_whbline, 2–73
ims_whdwline, 2–74
ims_whwline, 2–75

writing pixel, **ims_wpix**, 2–76

writing vertical line
ims_wvbline, 2–77
ims_wvdwline, 2–78
ims_wvwline, 2–79

Z

zoom, 1–6
display
ims_dzoom, 2–29
ims_dzoomxy, 2–30

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

APPENDIX A
IMS REGISTERS

This appendix lists the registers available on the IMS. Use the system–level functions **read_reg** to read values. Use the appropriate register set up function, or the system–level function **write_reg** to restore a value.

Table A–1 (page A–2) lists the symbolic names for IMS registers. An *indented* register name indicates a field that is within the non–indented name that comes before or above it. Refer to the module structure functions, or the file “ims.h” for defined symbolic values.

Figure A–1 (page A–8) shows the IMS hardware registers and their address offsets. The numbers in parenthesis indicate the number of bits in a field. Field names without numbers are a single bit wide. The IMS symbolic names are similar to the hardware names found in the *IMS Hardware Reference Manual*.

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

Table A–1. IMS Symbolic Register Names

Symbolic Name	Description
IMS_CONF	Configuration register
IMS_MAPSEL	Mapping select
IMS_STDSEL	Standard addressing select
IMS_EXTSEL	Extended addressing select
IMS_STDADR	Standard address
IMS_EXTADR	Extended address
IMS_STATUS	Status register
IMS_CAMVB	Camera vertical blank status
IMS_CAMFLD	Camera field status
IMS_VBUSVB	Video bus vertical blank status
IMS_VBUSFLD	Video bus field status
IMS_EN1SH0	One–shot 0 ready/halt status
AOIVB1	AOI1 Vertical blank status
IMS_AMMOD	AM installed status
IMS_DMMOD	DM installed status
IMS_CMMOD	CM installed status
IMS_MASKA	Frame A mask
IMS_MASKB	Frame B mask
IMS_CONST	Constant register
IMS_AOIXS0	AOI0 horizontal size
IMS_AOIYS0	AOI0 vertical size
IMS_AOIXB0	AOI0 horizontal blank size
IMS_AOIYB0	AOI0 vertical blank size

(continued)

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

IMS Symbolic Register Names

Symbolic Name	Description
IMS_AOICTL0	AOI0 Control register
IMS_AOIHLT0	AOI0 halt
IMS_AOILK0	AOI0 lock
IMS_OSHOT0	AOI0 1shot
IMS_AOIMODE	AOI mode enable
IMS_PAGE	Page register
IMS_PAGADR	Page address
IMS_PGSEL	Page select
IMS_PIXSZ	Pixel size (depth)
IMS_MEMEN	Memory map enable
IMS_CTL	Control register
IMS_SLAVSEL	Slave select
IMS_CLKSPD	Clock speed select
IMS_CLRBM	Clear bit masks
IMS_DMEN	Display enable
IMS_EXTRIG	External trigger
IMS_DISPOE	Dispaly output enable
IMS_IVEC	Interrupt vector
IMS_IEN	Interrupts enable
IMS_FMSEL	Frame select
IMS_FMIMOD	Frame interrupt mode
IMS_FMIEN	Frame interrupt enable
IMS_ETIEN	External trigger interrupt

(continued)

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

IMS Symbolic Register Names (continued)

Symbolic Name	Description
IMS_AMIEN	AM interrupt enable
IMS_DMIEN	DM interrupt enable
IMS_CMIEN	CM interrupt enable
IMS_ISTAT	Interrupt Status
IMS_INTACK	Interrupt Acknowledge
IMS_IPRI	Interrupt Priority
IMS_ACQA0	A0 Acquistion Control
IMS_TMODA0	Trigger mode
IMS_WTAQA0	Wait for acquisition enable (ACQEN)
IMS_NEWAQA0	New acquire status
IMS_VBA0	Vertical blank status
IMS_SMA0	Scan mode status
IMS_ABTSA0	A0 Acquisition command bits
IMS_ACQA1	A1 Acquisition Control
IMS_TMODA1	Trigger mode
IMS_WTAQA1	Wait for ACQEN
IMS_NEWAQA1	New acquire status
IMS_VBA1	Vertical blank status
IMS_SMA1	Scan mode status
IMS_ABTSA1	A1 Acquisition command bits
IMS_ACQEN	Acquisition Enable
IMS_ACQB1	B1 Acquisition Control
IMS_TMODB1	Trigger mode
IMS_WTAQB1	Wait for ACQEN
IMS_NEWAQB1	New acquire status
IMS_ABTSB1	B1 Acquisiton command bits

(continued)

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

IMS Symbolic Register Names (continued)

Symbolic Name	Description
IMS_RPANA0	A0 Read Pan
IMS_RSCRA0	A0 Read Scroll
IMS_WPANA0	A0 Write Pan
IMS_WSCRA0	A0 Write Scroll
IMS_RPANA1	A1 Read Pan
IMS_RSCRA1	A1 Read Scroll
IMS_WPANA1	A1 Write Pan
IMS_WSCRA1	A1 Write Scroll
IMS_CTLA	Frame A Control
IMS_A0BP	A0 bit pack
IMS_A1BP	A1 bit pack
IMS_WYZA0	A0 write y zoom
IMS_WYZA1	A1 write y zoom
IMS_WXZA0	A0 write x zoom
IMS_WXZA1	A1 write x zoom
IMS_RPANB1	B1 Read Pan
IMS_RSCRB1	B1 Read Scroll
IMS_WPANB1	B1 Write Pan
IMS_WSCRB1	B1 Write Scroll
IMS_CTLB	Frame B Control
IMS_DISPXZ	Display x zoom
IMS_DISPYZ	Display y zoom
IMS_WYZB1	B1 write y zoom
IMS_B1BP	B1 bit pack
IMS_WXZB1	B1 write x zoom

(continued)

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

IMS Symbolic Register Names (continued)

Symbolic Name	Description
IMS_PVB2	Channel 2 Input
IMS_PVB2S	Input bus select
IMS_PVB2F	Output function
IMS_PVB2M	(must be 1)
IMS_PVB1	Channel 1 Input
IMS_PVB1S	Input bus select
IMS_PVB1F	Output function
IMS_PVB1M	(must be 1)
IMS_PVB0	Channel 0 Input
IMS_PVB0S	Input bus select
IMS_PVB0F	Output function
IMS_PVB0M	(must be 1)
IMS_PIPECON	Pipeline Control
IMS_PCM0S	PCM0 bus select
IMS_PCM1S	PCM1 bus select
IMS_PCM2S	PCM2 bus select
IMS_CMBYP	CM bypass
IMS_PVB3	Channel 1 Input
IMS_PVB3S	Input bus select
IMS_PVB3F	Output function
IMS_PVB3M	(must be 1)

(continued)

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

IMS Symbolic Register Names (continued)

Symbolic Name	Description
IMS_FA0SEL	Frame A0 Input
IMS_FA0S	Input bus select
IMS_FA0F	Output function
IMS_FA0M	(must be 1)
IMS_FA0FTS	Input timing select
IMS_FA1SEL	Frame A1 Input
IMS_FA1S	Input bus select
IMS_FA1F	Output function
IMS_FA1M	(must be 1)
IMS_FA1FTS	Input timing select
IMS_FB1SEL	Frame B1 Input
IMS_FB1S	Input bus select
IMS_FB1F	Output function
IMS_FB1M	(must be 1)
IMS_FB1FTS	Input timing select
IMS_CPSCTL	CPS Control

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

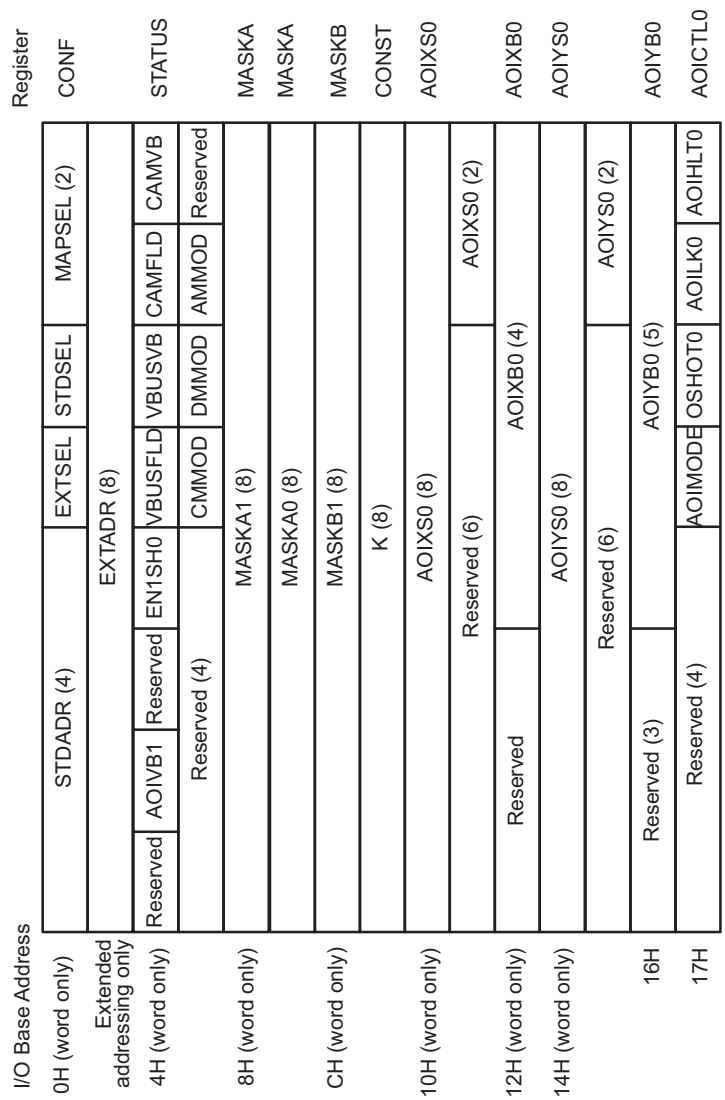


Figure A–1. IMS Registers

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



Figure A–1. IMS Registers (cont.)

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

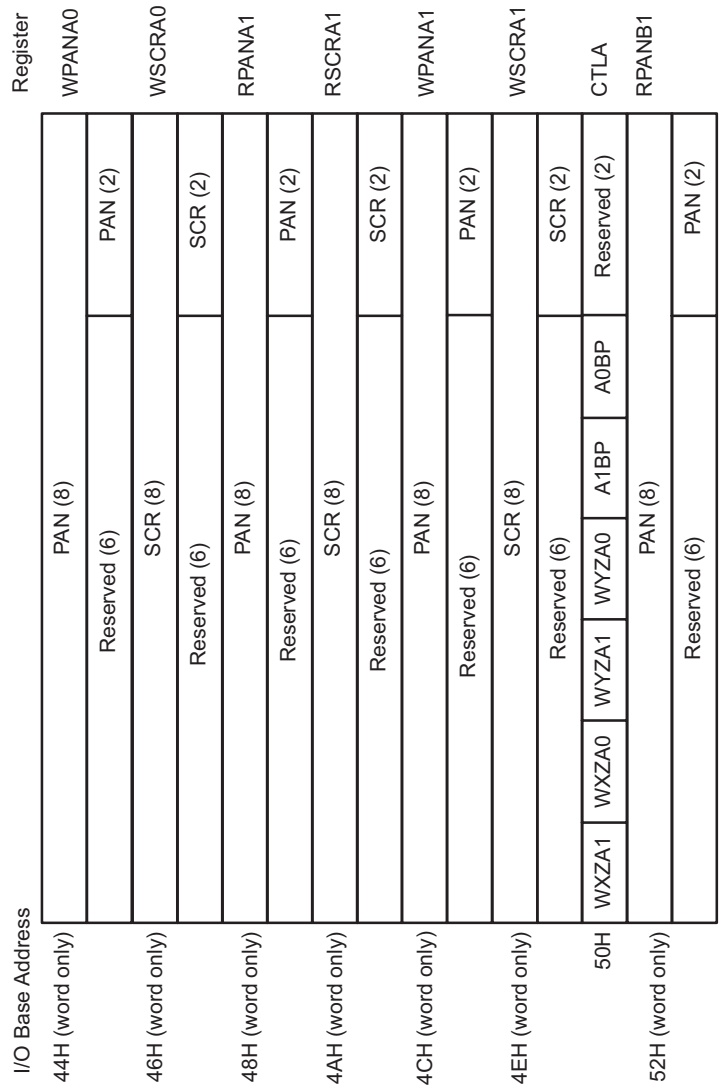


Figure A–1. IMS Registers (cont.)

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

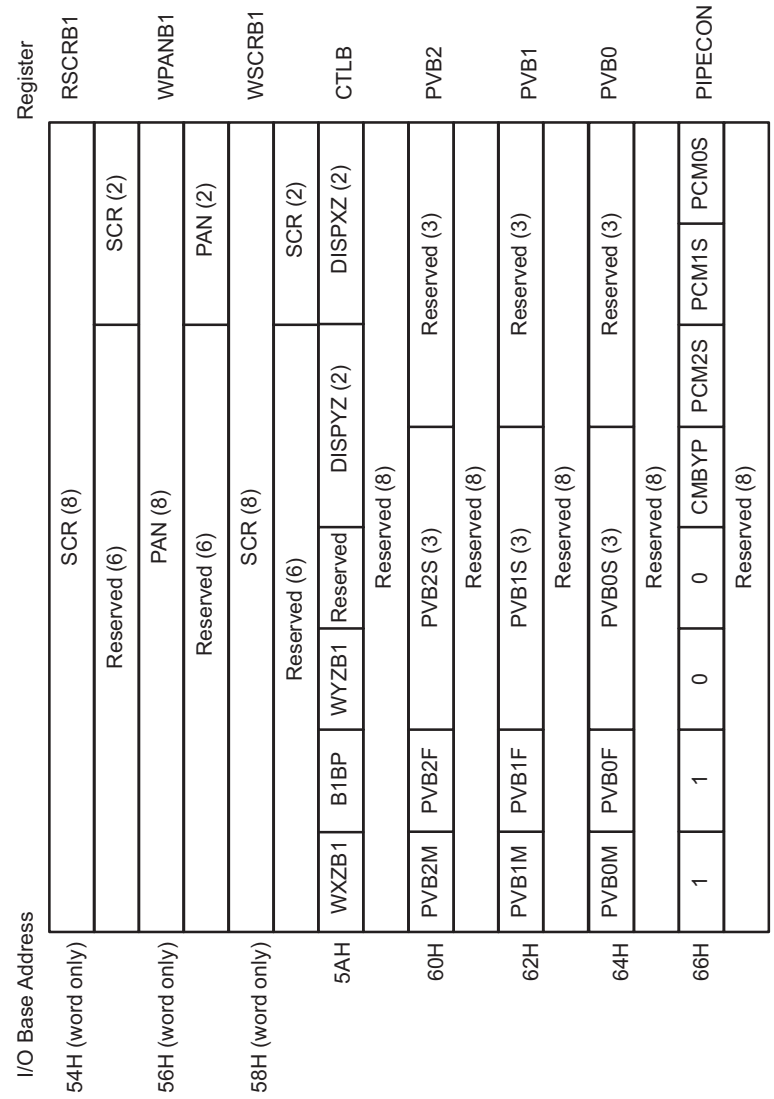


Figure A–1. IMS Registers (cont.)

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

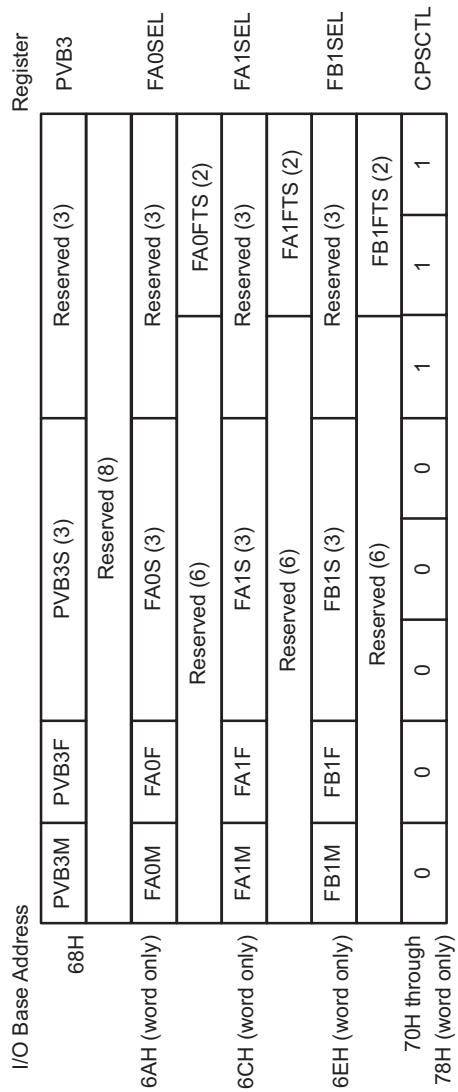


Figure A–1. IMS Registers (cont.)

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

ims_dbrd_intr — Enable or Disable the Interrupt on a Daughter Board

Syntax short **ims_dbrd_intr**(MODCNF **ims_ptr*, short *bit*, short *mode*)

ims_ptr Pointer to the IMS.

bit	Daughter board to set the interrupt: IMS_I_SLOT0 — Set DM interrupts. IMS_I_SLOT1 — Set CM interrupts. IMS_I_SLOT2 — Set AM interrupts.
mode	Value to set: IMS_I_OFF — Disable interrupts. IMS_I_ON — Enable interrupts. INQUIRE — Return the current value.

Description

This functions enables, disables, or returns the interrupt bits on the selected daughter board. This function changes the IMS module structure from its previous value.

Returned Values

IMS_I_OFF — Interrupts disabled.
IMS_I_ON — Interrupts enabled.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_IEN

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

ims_ext_intr — Enable or Disable External Trigger Interrupt

Syntax short **ims_ext_intr**(MODCNF **ims_ptr*, short *mode*)

ims_ptr Pointer to the IMS.

mode Indicates interrupt value to set:
IMS_I_OFF — Disable interrupts.
IMS_I_ON — Enable interrupts.
INQUIRE — Return the current value.

Description

This function sets or returns the IMS's external interrupt bit. This function changes the IMS module structure from its previous value.

Returned Values

0 — Interrupt disabled.
1 — Interrupt enabled.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_IEN

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

ims_frame_intr — Enable or Disable the Frame Interrupt

Syntax short **ims_frame_intr**(MODCNF **ims_ptr*, short *mode*)

ims_ptr Pointer to the IMS.

mode Interrupt to enable:

IMS_I_A0ACQ — Enable an interrupt at the end of acquisition into frame A0.

IMS_I_A1ACQ — Enable an interrupt at the end of acquisition into frame A1.

IMS_I_B1ACQ — Enable an interrupt at the end of acquisition into frame B1.

IMS_I_A0VB — Enable an interrupt at the start of vertical blank in frame A0.

IMS_I_A1VB — Enable an interrupt at the start of vertical blank in frame A1.

IMS_I_B1VB — Enable an interrupt at the start of vertical blank in frame B1.

INQUIRE — Return the current value.

Description

This function sets or returns the current frame interrupt enable values. This function changes the IMS module structure from its previous value.

Returned Values

0 — All interrupts disabled. No interrupts set.
8 — Frame A0 interrupted enabled at the end of acquistion.
9 — Frame A1 interrupted enabled at the end of acquistion.
11 — Frame B1 interrupted enabled at the end of acquistion.
12 — Frame A0 interrupted enabled at the start of vertical blank.
13 — Frame A1 interrupted enabled at the start of vertical blank.
15 — Frame B1 interrupted enabled at the start of vertical blank.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_IEN

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

ims_ien_bits — Enable or Disable Interrupts

Syntax short **ims_ien_bits**(MODCNF **ims_ptr*, short *bits*)

ims_ptr Pointer to the IMS.

bits Interrupt select bits:

IMS_I_SLOT0 — Enable DM interrupts.

IMS_I_SLOT1 — Enable CM interrupts.

IMS_I_SLOT2 — Enable AM interrupts.

IMS_I_EXT — Enable external interrupts.

IMS_I_A0ACQ — Enable an interrupt at the end of acquisition into frame A0.

IMS_I_A1ACQ — Enable an interrupt at the end of acquisition into frame A1.

IMS_I_B1ACQ — Enable an interrupt at the end of acquisition into frame B1.

IMS_I_A0VB — Enable an interrupt at the start of vertical blank in frame A0.

IMS_I_A1VB — Enable an interrupt at the start of vertical blank in frame A1.

IMS_I_B1VB — Enable an interrupt at the start of vertical blank in frame B1.

INQUIRE — Return the current value.

Description

This function enables or disables the interrupt enable bits. The first four values can be ORed with the one of the next six to enable interrupts. This function changes the IMS module structure from its previous value.

Returned Values

Current selected bits.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_IEN

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

ims_intr_priority — Set or Return the Interupt Priority Level

Syntax short **ims_intr_priority**(MODCNF **ims_ptr*, short *ipri*)

ims_ptr Pointer to the IMS.

ipri Interrupt level:
IMS_I_LVL1 — Interrupt level one.
IMS_I_LVL2 — Interrupt level two.
IMS_I_LVL3 — Interrupt level three.
IMS_I_LVL4 — Interrupt level four.
IMS_I_LVL5 — Intterupt level five.
IMS_I_LVL6 — Intterupt level six.
IMS_I_LVL7 — Interrupt level seven.
INQUIRE — Return the current value.

Description

During an interrupt acknowledge cycle, the host broadcasts a three–bit priority code on address lines A03–A01. This code is compared with *ipri*. If this code matches *ipri*, the IMS places the interrupt vector onto the bus. This function changes the IMS module structure from its previous value.

Returned Values

Current level.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_IPRI

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

ims_intr_vec — Set or Return the Interrupt Vector

Syntax short **ims_intr_vec**(MODCNF ***ims_ptr*, short *ivec*)

ims_ptr Pointer to the IMS.

ivec Interrupt vector used to handle interrupts in the range of 0 to 0xFF.
INQUIRE — Return the current value.

Description

This function defines or returns the interrupt vector. This function changes the IMS module structure from its previous value.

Returned Values

Current interrupt vector.

Error Values

IMS_NO_ERROR — Operation successfully completed.

Hardware Modified

IMS_IVEC

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

Series 150/40 ™

ITEX ™

IMS Software

Function Reference Card

47–S54002–01

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

iMAGING Technology

ims_acqbits(MODCNF **mod*, short *frame* short *mode*) Set or Return the Acquisition Command Bits

ims_aoi_1shot(MODCNF **ims_ptr,* short *aoisel*) Perform One Clock Cycle

ims_aoi_1shot_async(MODCNF **ims_ptr,* short *aoisel*) Perform One Clock Cycle Asynchronously

ims_aoi_grab(MODCNF **ims_ptr,* short *aoisel*, short *frame*) Acquire Multiple Images with the AOI Generator

ims_aoi_grab16(MODCNF **ims_ptr,* short *aoisel*, short *frame*) Acquire Multiple 16–Bit Images with the AOI Generator

ims_aoi_halt(MODCNF **ims_ptr,* short *aoisel*) Stop an AOI Timing Generator

ims_aoi_snap(MODCNF **ims_ptr,* short *aoisel*, short *frame*, short *mode*) Acquire a Single Image with the AOI Generator

ims_aoi_snap_async(MODCNF **ims_ptr,* short *aoisel*, short *frame*, short *mode*) Acquire a Single Image with the AOI Generator Asynchronously

ims_aoi_snap16(MODCNF **ims_ptr,* short *aoisel*, short *frame*, short *mode*) —Acquire a Single 16–Bit Image with the AOI Generator

ims_aoi_snap16_async(MODCNF **ims_ptr,* short *aoisel*, short *frame*, short *mode*) —Acquire a Single 16–Bit Image with the AOI Generator Asynchronously

ims_aoi_start(MODCNF **ims_ptr,* short *aoisel*) Start an AOI Timing Generator

ims_bitmask(MODCNF **ims_ptr*, WORD *reg*, short *mode*, WORD **value*) Set or Return the Bit Mask

ims_chbline(DWORD *address*, short *num_ops*, short *freq*, BYTE *value*) Clear a Horizontal Line to a BYTE Value

ims_chdwline(DWORD *address*, short *num_ops*, short *freq*, DWORD *value*) Clear a Horizontal Line to a DWORD Value

ims_chwline(DWORD *address*, short *num_ops*, short *freq*, WORD *value*) Clear a Horizontal Line to WORD Value

ims_clkspeed(MODCNF **ims_ptr*, short *speed*) Set or Return the Clock Speed

ims_clr_area(MODCNF **ims_ptr*, short *frame*, short *x*, short *y*, short *dx*, short *dy*, WORD *value*) Clear an Area

ims_clr_frame(MODCNF **ims_ptr*, short *frame*, WORD *value*) Clear an Frame

ims_cp_area(MODCNF **ims_ptr1*, short *frame1*, short *x1*, short *y1*, short *dx*, short *dy*, MODCNF **ims_ptr2*, short *frame2*, short *x2*, short *y2*) Copy an Area

ims_cvbline(DWORD *address*, short *num_ops*, short *linesz*, BYTE *value*) Clear a Vertical Line to a BYTE Value

ims_cvdwline(DWORD *address*, short *num_ops*, short *freq*, DWORD *value*) Clear a Vertical Line to a DWORD Value

ims_cvline(DWORD *address*, short *num_ops*, short *linesz*, WORD *value*) Clear a Vertical Line to a WORD Value

ims_dbrd_intr(MODCNF **ims_ptr*, short *bit*, short *mode*) Enable or Disable the Interrupt on a Daughter Board

ims_delay_acqen (DWORD *address*, short *num_ops*, short *freq*, BYTE **array*) Read a Horizontal Line of BYTEs— Begin a Delayed Acquire

ims_dispmode(MODCNF **ims_ptr*, short *disp*) Set or Return the Display Mode

ims_dregs(MODCNF **ims_ptr*) Display the Register Values

ims_dump_mem(MODCNF **ims_ptr*, short *frame*, short *x,* short *y*, short *dx*, short *dy*) Display an Area

ims_dzoom(MODCNF **ims_ptr*, short *zoom*) Zoom Frame B1 for Display

ims_dzoomxy(MODCNF **ims_ptr*, short *xzoom*, short *yzoom*) Set Frame B1 X and Y Zoom Factors for Display

ims_ext_intr(MODCNF **ims_ptr*, short *mode*) Enable or Disable External Trigger Interrupt

ims_frame_intr(MODCNF **ims_ptr*, short *mode*) Enable or Disable the Frame Interrupt

ims_freeze(MODCNF **ims_ptr*, short *frame*) Stop Acquisition

ims_get_adrform(MODCNF **ims_ptr*) Return the Address Format

ims_get_aoi(MODCNF **ims_ptr*, short **dx*, short **dy*, short **hblen*, short **vblen*, short *aoisel*) Return the AOI Location and Dimensions

ims_get_mapadr(MODCNF **ims_ptr*, DWORD **adr*) Return the VME Memory Map Address

ims_get_mod(short *seqnum*) Return a Pointer to an IMS

ims_get_rdroam(MODCNF **ims_ptr*, short *frame*, short **p*, short **s*) Return the Read Pan and Scroll Shadow Registers

ims_get_slot0(MODCNF **ims_ptr*) Return a Pointer to a DM

ims_get_slot1(MODCNF **ims_ptr*) Return a Pointer to a CM

ims_get_slot2(MODCNF **ims_ptr*) Return a Pointer to an AM

ims_get_wrroam(MODCNF **ims_ptr*, short *frame*, short **p*, short **s*) Return the Write Pan and Scroll Shadow Registers

ims_grab(MODCNF **ims_ptr*, short *frame*) Acquire Multiple Images

ims_intr_priority(MODCNF **ims_ptr*, short *ipri*) Set or Return the Interupt Priority Level

ims_intr_vec(MODCNF **ims_ptr*, short *ivec*) Set or Return the Interrupt Vector

ims_iregs(MODCNF **ims_ptr*) Initialize the Registers

ims_mapsize(MODCNF **ims_ptr*, short *mapsel*) Set or Return the Memory Map Size

ims_pixsz(MODCNF **ims_ptr*, short *pixsz*) Set or Return the Pixel Size

ims_psreset(MODCNF **ims_ptr*) Set the Pan and Scroll Registers

ims_rhbline(DWORD *address*, short *num_ops*, short *freq*, BYTE **array*) Read a Horizontal Line of BYTEs

ims_rhdwline(DWORD *address*, short *num_ops*, short *freq*, DWORD **array*) Read a Horizontal Line of DWORD Values

ims_rhwline(DWORD *address*, short *num_ops*, short *freq*, WORD **array*) Read a Horizontal Line of WORD Values

ims_rpix (MODCNF **ims_ptr*, short *frame*, short *x*, short *y*) Read a Single Pixel Value

ims_rvbline(DWORD *address*, short *num_ops*, short *linesz*, BYTE **array*) Read a Vertical Line of BYTE Values

ims_rvdwline(DWORD *address*, short *num_ops*, short *linesz*, DWORD **array*) Read a Vertical Line of DWORD Values

ims_rvwline(DWORD *address*, short *num_ops*, short *linesz*, WORD **array*) Read a Vertical Line of WORD Values

ims_sel_slave(MODCNF **ims_ptr*, short *sel*) Select or Return a Slave to Output to the Video Bus

ims_set_aoi(MODCNF **ims_ptr*, short *dx*, short *dy*, short *hblen*, short *vblen,* short *aoisel*) Set the AOI Location and Dimensions

ims_set_cpspath(MODCNF **ims_ptr*, short *spath*, short *dpath*) Set the Cross–port Switch

ims_set_lineio(MODCNF **ims_ptr*, short *frame*, short *x*, short *y*, short **xsz*, DWORD **offadr*) Set–up Frame Buffer for Access

ims_set_mapformat(MODCNF **ims_ptr*, short *format*, DWORD *adr*) Set the Memory Access Format and the Memory Base Address

ims_set_rdroam(MODCNF **ims_ptr*, short *frame*, short *p*, short *s*) Set the Read Pan and Scroll Shadow Registers

ims_set_wrroam(MODCNF **ims_ptr*, short *frame*, short *p*, short *s*) Set the Write Pan and Scroll Shadow Registers

ims_snap(MODCNF **ims_ptr*, short *frame*) Acquire a Single Image

ims_snap_async(MODCNF **ims_ptr*, short *frame*) Acquire a Single Image Asynchronously

ims_vbusmode(MODCNF **ims_ptr*, short *modesel*) Set the Video Bus Timing

ims_wait_acq(MODCNF **ims_ptr*, short *frame*) Wait for Acquisition to Complete

ims_wait_notvb(MODCNF **ims_ptr*, short *frame*, short *vbsync*) Wait Until Vertical Blank is Past

ims_wait_vb(MODCNF **ims_ptr*, short *frame*) Wait for Next Vertical Blank

ims_whbline(DWORD *address*, short *num_ops*, short *freq*, BYTE **array*) Write a Horizontal Line of BYTEs

ims_whdwline(DWORD *address*, short *num_ops*, short *freq*, DWORD **array*) Write a Horizontal Line of DWORD Values

ims_whwline(DWORD *address*, short *num_ops*, short *freq*, WORD **array*) Write a Horizontal Line of WORD Values

ims_wpix (MODCNF **ims_ptr*, short *frame*, short *x*, short *y*, WORD *pv*) Write a Single Pixel Value

ims_wvbline(DWORD *address*, short *num_ops*, short *linesz*, BYTE **array*) Write a Vertical Line of BYTE Values

ims_wvdwline(DWORD *address*, short *num_ops*, short *linesz*, DWORD **array*) Write a Vertical Line of DWORD Values

ims_wvwline(DWORD *address*, short *num_ops*, short *freq*, WORD **array*) Write a Vertical Line of WORD Values

Copyright© 1999, 1998–1993 Imaging Technology, Incorporated.
All rights reserved.
Printed in the United States of America.

ITEX, MVC, MVC 150/40, and the IMAGING logo are trademarks of Imaging Technology Incorporated.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.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