VDSP431

Digital Servo Control
VMEbus Module

PROGRAMMING MANUAL

730-11-000-4000

Revision 1.1
June 1999

ALPHI TECHNOLOGY CORPORATION
6202 S. Maple Avenue #120
Tempe, AZ 85283 USA
Tel : (602) 838-2428
Fax: (602) 838-4477

NOTICE

The information in this document has been carefully checked and is believed to be entirely reliable. While all reasonable efforts to ensure accuracy have been taken in the preparation of this manual, ALPHI TECHNOLOGY assumes no responsibility resulting from omissions or errors in this manual, or from the use of information contained herein.

ALPHI TECHNOLOGY reserves the right to make any changes, without notice, to this or any of ALPHI TECHNOLOGY's products to improve reliability, performance, function or design.

ALPHI TECHNOLOGY does not assume any liability arising out of the application or use of any product or circuit described herein; nor does ALPHI TECHNOLOGY convey any license under its patent rights or the rights of others.

A16 PROGRAMMING

Two types of registers are accessible by the VMEbus in A16 (Short I/O space). The first type is a set of hardware registers whose address is determined by jumpers located on the VDSP431 baseboard. The second type is a VMEbus "Global" mailbox register that is address programmable by software. The VMEbus host can broadcast commands to multiple VDSP431 boards using this register.

A16 SHORT I/O REGISTERS

A16 Offsets	Register	Description	R/W
$0001	MAP24	A24 Address Map Bits	W
$0003	MAP32	A32 Address Map Bits	W
$0005	IVR	Interrupt Vector Register	R/W
$0007	GMBAR	Global Mailbox Address Register	W
$0009	MBDR	Mailbox Data Register	W
$000F	VCTRL	VDSP431 Control Register	W

MAP24 REGISTER

BASE ADDRESS + $01

The MAP24 register sets the location of the VDSP431 shared memory in A24 space. Following a power-on reset, the VDSP431 shared memory is disabled. The MAP24 register must be written to set the VMEbus address of the shared memory. This register is write only and cannot be read back. Software must maintain a shadow copy of the register contents. The MAP24 register is defined below:

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
A23	A22	A21	not used	not used	not used	not used	ENABLE

NOTE: When a bit is set to a "1", the comparator is active if the corresponding address line is equal to a "1". Bit 0 of this register must be set to a "1" to enable to shared memory decoder.

MAP32 REGISTER

BASE ADDRESS + $03

The MAP32 register sets the location of the VDSP431 shared memory in A32 space. Following a power-on reset, the VDSP431 shared memory is disabled. The MAP32 register must be written to set the VMEbus address of the shared memory. This register is write only and cannot be read back. Software must maintain a shadow copy of the register contents. The MAP32 register is defined below:

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
A31	A30	A29	A28	A27	A26	A25	A24

NOTE: When a bit is set to a "1", the comparator is active if the corresponding address line is equal to a "1". Bit 0 of the MAP24 register must be set to a "1" to enable to shared memory decoder.

INTERRUPT VECTOR REGISTER (IVR)

BASE ADDRESS + $05

The Interrupt Vector Register is an 8-bit Read/Write register. The value written to this register will be presented to the VMEbus during an interrupt acknowledge (IACK) cycle. The VDSP431 can be programmed to assert an interrupt request to the VMEbus host following command completion or when a block of data has been processed. When the VDSP431 interrupts the VMEbus, the host CPU will run an IACK cycle in order to fetch an interrupt vector. The IVR register is defined below:

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
IV07	IV06	IV05	IV04	IV03	IV02	IV01	IV00

GLOBAL MAILBOX BASE ADDRESS REGISTER (GMBAR)

BASE ADDRESS + $07

The GMBAR register defines a Global A16 Mailbox address that multiple VDSP431 boards can respond to. The VMEbus can communicate global information to one or more VDSP431 boards. This register is write only and cannot be read back. Software must maintain a shadow copy of the register contents. The GMBAR decoder must be enabled, by setting bit 1 in the *VCTRL* register . The GMBAR register is defined below:

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
A15	A14	A13	A12	A11	A10	A09	A08

MAILBOX REGISTER (MBOX)

BASE ADDRESS + $09

The Mailbox register is a write only register that is used to interrupt the VDSP431.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
-	-	-	-	-	-	-	-

VDSP431 CONTROL REGISTER (VCTRL)

BASE ADDRESS + $0F

The VCTRL register is used to :

- Reset the VDSP431. This register can hold the VDSP431 hardware in a reset state that is independent of the VMEbus SYSRESET. This register is write only and cannot be read back. Software must maintain a shadow copy of the register contents.

- Enable the Global Address Register
- Enable the Master mode for the VDSP431 to generate a VMEbus DTACK.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
-	-	-	-	-	MASTER	GLOBAL ENABLE	RST

RST

When RST = 1, the VDSP431 is held in Reset.
When RST = 0, the VDSP431 is released from Reset.

GLOBAL ENABLE

When set to a ”1”, the VDSP431 will generate a local interrupt if there is an A16 access to the address specified by GMBAR. When set to a “0”, accesses are ignored.

MASTER

If set to a “1”, the VDSP431 module generates a DTACK acknowledge when an access is made to the address specified by GMBAR. Only one VDSP431 module should be programmed with this bit set to a “1”. Additional VDSP431 boards must be programmed with this bit set to a “0”.

A32/A24 Memory Access

The VDSP431 contains a dual access memory that is shared between the VDSP431 CPU and the VMEbus. The 'C' structure of this memory is shown below:

Data Types

Data Type	Description
IEEE_FLOAT	IEEE-754 32-bit single precision float
UINT32	32-bit unsigned long integer
UINT8	8-bit unsigned integer

VDSP431 Channel Register Template

```
typedef struct host_chan
{
      /* analog feedback variables */
IEEE_FLOAT  Fbk     [MAX_CHAN];     /* current values of feedback */
IEEE_FLOAT  Fbk1    [MAX_CHAN];     /* current values of feedback 1 */
IEEE_FLOAT  Fbk2    [MAX_CHAN];     /* current values of feedback 2 */
IEEE_FLOAT  FbkCyc  [MAX_CHAN];     /* feedback cyclic values */
IEEE_FLOAT  FbkAvg  [MAX_CHAN];     /* current average values */
IEEE_FLOAT  Cmd     [MAX_CHAN];     /* current value composite cmd */
IEEE_FLOAT  Err     [MAX_CHAN];     /* current error */
IEEE_FLOAT  Vout    [MAX_CHAN];     /* current valve output */

      /* Dc, Ac, Freq, and Phase reference controls */
IEEE_FLOAT  DCOff  [MAX_CHAN];      /* DC EndPoint to Ramp to */
IEEE_FLOAT  ACamp  [MAX_CHAN];      /* AC amplitude */
IEEE_FLOAT  ACphs  [MAX_CHAN];      /* AC phase */

      /* control loop constants */
IEEE_FLOAT  Kp [MAX_CHAN];          /* Proportional gain constant */
IEEE_FLOAT  Ki [MAX_CHAN];          /* Integral gain constant */
IEEE_FLOAT  Kd [MAX_CHAN];          /* Derivative gain constant */
IEEE_FLOAT  Il [MAX_CHAN];          /* Integration limit */
UINT32      Ds [MAX_CHAN];          /* Derivative sample period */

      /* calibration control */
UINT32      CalRelay        [MAX_CHAN];  /* close/open cal relay */
IEEE_FLOAT  Cond1Exc        [MAX_CHAN];  /* set bridge excitation */
IEEE_FLOAT  Cond1Gain       [MAX_CHAN];  /* set Conditioner 1 gain */
IEEE_FLOAT  Cond1Offset     [MAX_CHAN];  /* set Conditioner 1 offset */
UINT32      Cond1UseDCHI    [MAX_CHAN];  /* 1=DC High Level, 0=LVDT */
IEEE_FLOAT  Cond2Exc        [MAX_CHAN];  /* set bridge excitation */
IEEE_FLOAT  Cond2Gain       [MAX_CHAN];  /* set Conditioner 2 gain */
IEEE_FLOAT  Cond2Offset     [MAX_CHAN];  /* set Conditioner 2 offset */
UINT32      Cond2UseDCHI    [MAX_CHAN];  /* 1=DC High Level, 0=LVDT */

      /* valve dither control */
IEEE_FLOAT  DitherAmp   [MAX_CHAN];    /* sets dither amplitude */

      /* valve range control */
UINT32      VlvRange    [MAX_CHAN];
IEEE_FLOAT  VlvGain     [MAX_CHAN];

      /* polarity controls */
UINT32      FbkPol1     [MAX_CHAN];     /* FeedBack Polarity 1 */
UINT32      FbkPol2     [MAX_CHAN];     /* FeedBack Polarity 2 */
UINT32      FbkSel      [MAX_CHAN];     /* Feedback Select */
UINT32      VlvPol[MAX_CHAN];           /* Valve Polarity */

      /* servo enable for channel */
UINT32      PidEnable[MAX_CHAN];        /* PID channel enable */
IEEE_FLOAT  VlvOffset[MAX_CHAN];        /* static valve offset */
CTRL_STAT   CtrlStat[MAX_CHAN];         /* Status bit fields */
```

```
      /* Square wave tuning controls */
UINT32      TuneEnable[MAX_CHAN];         /* Tune Enable 1=On,0=Off */
IEEE_FLOAT  TuneAmplitude[MAX_CHAN];      /* Amplitude in volts */
IEEE_FLOAT  TunePeriod[MAX_CHAN];         /* Period of 1 cycle */

UINT32      FilterOn[MAX_CHAN];         /* Filter Enable */
IEEE_FLOAT  FilterFreq[MAX_CHAN];       /* Filter Cut-Off Frequency */

UINT32      TraceCmdOffset[MAX_CHAN];   /* Offset to Command Buffer */
UINT32      TraceFbkOffset[MAX_CHAN];   /* Offset to Feedback Buffer */

IEEE_FLOAT  CoefA[MAX_CHAN];
IEEE_FLOAT  CoefB[MAX_CHAN];

UINT32      NFilterCoef[MAX_CHAN];
IEEE_FLOAT  FilterCoef[MAX_CHAN][ MAX_COEF ];

      /* Limitit Checking Variables */
IEEE_FLOAT  AlarmLimit[MAX_CHAN];       /* Alarm Limit */
UINT32      AlarmFilter[MAX_CHAN];      /* Alarm Limit Filter */
IEEE_FLOAT  CriticalLimit[MAX_CHAN];    /* Critical Limit */
UINT32      CriticalFilter[MAX_CHAN];   /* Critical Limit Filter */
IEEE_FLOAT  FbkLowerLimit[MAX_CHAN];    /* Feedback Lower Limit */
IEEE_FLOAT  FbkUpperLimit[MAX_CHAN];    /* Feedback Upper Limit */
UINT32      FbkLimitFilter[MAX_CHAN];   /* Feedback Limit Filter */

} HCHAN;
```

VDSP431 AC Power/Interlock Status Register Template

```
typedef struct interlock
{
    UINT32  InterlockStatus;    /* Register 0 */
    UINT32  ACPowerStatus;      /* Register 1 */
    UINT32  EnableStatus;       /* Register 2 */
    UINT32  FaultLimitStatus;   /* Register 3 */
    UINT32  KillACEnable;       /* Register 4 */
    UINT32  InterlockLatch;     /* Latched version of Register 0 */
} INTLOCK;
```

Vdsp431 host module template

```
typedef struct host_module
{
UINT32      AvailChan;        /* Number of available channels */
IEEE_FLOAT  MasterSpan;       /* span control all channels */
IEEE_FLOAT  ACPeriod;         /* period for AC waveforms */
IEEE_FLOAT  DCPeriod;         /* period for DC transitions */
UINT32      FpChan;           /* front panel channel selection */
UINT32      DigIn;            /* Digital Input Image */
UINT32      DigOut;           /* Digital Output Image */
UINT32      RampType;         /* 0 = Linear, 1 = HaverSine */
UINT32      TraceEnable;      /* 1 = Record, 0=Off */
UINT32      TraceNdx;         /*  Current position in Trace Buffers */
INTLOCK     IntLock;          /* AC Power Interlock Registers */
UINT32      TrigMode;         /* INTERNAL or EXTERNAL_TRIGGER */
UINT32      CmdNdx;           /* Waveform Command Index 0-255 */
IEEE_FLOAT  AuxSetpoint[ MAX_CHAN ];      /* Setpoint Knob Values */
UINT32      DispChan[4];      /* Readout Display Channel */
UINT32      DispParm[4];      /* Readout Display Parameter */
UINT32      LimitStatus;      /* 32-bit Limit Status Word */
UINT32      LimitControl;     /* 32-bit Limit Control Word */
UINT32      ExtNullPace;      /* */
UINT32      ILockEnable[4];   /* Kill AC Interlock Enable Bits */

} HMOD;
```

Boot Control

```
typedef struct _boot_ctrl
{
UINT32 Command;
UINT32 Status;
UINT32 Size;
UINT8  *Image;
} BOOT_CTRL;
```

Dual-Access RAM Structure

```
typedef struct _dport
{
      /* BootRom control structure */
BOOT_CTRL   BootCtrl;

      /* Host Communication Registers */

UINT32     Revision[REV_SIZE];  /* Version String */
UINT32     CmdCode;             /* Command code to execute */
UINT32     CmdStat;             /* Command status */
UINT32     ErrCode;             /* Command error code */
UINT32     IrqEnable;           /* Interrupt Enable word */
UINT32     GlobalCommand;
UINT32     FirstChan;           /* First channel affected by command */
UINT32     LastChan;            /* Last channel affected by command */

HMOD       hmod;                /* Module variable list */
HCHAN      hchan;               /* Channel variable list */

} DPORT;
```

VDSP431 Boot Control

Following a power-on or hardware reset, the VDSP431 Bootrom gains control and accepts commands from either the VMEbus or a serial console located on the front panel. The VMEbus host can send commands to the VDSP431 to boot a program from flash, shared memory, or to program the flash.

Boot Control Structure BootCtrl

VMEbus Address Offset	DSP Address Offset	Register Name
$00000000	$00000000	Command
$00000004	$00000001	Status
$00000008	$00000002	Size
$0000000C	$00000003	Image
$00000010	$00000004	Image Data
….	….	….

The first 4 longwords are occupied by the BOOT_CTRL structure. For example, if the base address of the VDSP431 A32 space is $40000000 the Image Data Is located at $40000010.

Storing a program into flash

The VMEbus host must first download a binary image to the VDSP431 Image Data area. Next, the following registers must be written to:

Register	Value	NOTES
Status	0	Clear for loader
Size	LoadSize	Size of the Loaded Flash Image in bytes
Image	0x00500004	Points DSP to first byte of image
Command	4	Program “Flash” Image Command
		Delay for .1 Second
Command	0	No Command

Starting a program stored in flash

Register	Value	NOTES
Status	0	Clear for loader
Size	0	DSP Loader determines size
Image	0x00480000	DSP Local Address to Flash Memory
Command	2	Load “Flash” Image Command

Starting a program stored in RAM

The VMEbus host must first download a binary image to the VDSP431 Image Data area. Next, the following registers must be written to:

Register	Value	NOTES
Status	0	Clear for loader
Size	0	DSP Loader determines size
Image	0x00500004	DSP Local Address to Flash Memory
Command	1	Load “RAM” Image Command

Revision

This location contains an 80 character string that describes the VDSP431 firmware revision. Please note that each character occupies 4 bytes.
The string has the following format:

VDSP431 Revision: REV Last Build: DATE TIME

```
Where REV is a numeric revision number
DATE is the date of the last firmware build
TIME is the time of the last firmware build
```

CmdCode

Writing a command code to this register will instruct the VDSP431 to execute a command. The VDSP431 polls this register at regular intervals, checking for a non- zero value. The VDSP431 will clear this register when command processing begins.

The data parameters associated with a command code must be written by the VMEbus host before writing to this register. Some command codes can affect more than one VDSP431 channel. The range of channels to which the command code should be applied are determined by the **FirstChan** and **LastChan** registers. Valid command codes are 1 - 19. Currently supported command code are described below :

VDSP431 COMMAND CODE TABLE

COMMAND	CODE	FUNCTION
CAL_RELAY	1	Sets Calibration Relay States
FBK_CONFIG	2	Sets Feedback Parameters
VLV_CONFIG	3	Sets Valve Parameters
PID_GAINS	4	Sets Loop Gains
PID_ENABLE	5	Sets Loop Enable
FP_CHANNEL	*6*	*RESERVED NOT USED*
DC_OFFSET	7	Sets DC Setpoint
AC_AMPLITUDE	8	Sets AC Span
AC_PHASE	9	Sets AC Phase
MASTER_SPAN	10	Sets Master Span All Channels
AC_PERIOD	11	Sets AC Period All Channels
DC_PERIOD	12	Sets DC Ramp Time All Channels
TRIGGER_STATE	13	RESERVED NOT USED
DIGITAL_OUTPUTS	*14*	*RESERVED NOT USED*
RAMP_TYPE	15	Sets Linear or Haver-Sine Ramp
TUNE_ENABLE	16	Sets Tune Enable
ERROR_LIMIT	17	Sets Inner Loop Error Limit
NULLP_LIMIT	18	RESERVED NOT USED
CONFIG_FILTER	19	Sets LowPass Filter Parameters
ACK_LIMIT	20	Acknowledge Critical Limit
FAULT_CLEAR	21	Clear Interlock Fault
FAULT_ENABLE	22	Enable AC Interlocks

CmdStat

This register indicates the status of the last command written to the **CmdCode** register. This register will remain zero until a command completes.

BIT	FUNCTION
0	**1 = Ready for Command / OK, 0 = BUSY**
1	**1 = Command Error**

ErrCode

This register will contain an error code for commands that did not complete properly. The following codes are currently defined:

Code		Meaning
0	ERR_NONE	No Error, OK
1	ERR_INVALID_CHAN	The specified channel range was invalid
2	ERR_INVALID_CMD	The specified command code was invalid
3	ERR_RANGE	The data value(s) are out of range

IrqEnable

NOT USED AT THIS TIME

GlobalCommand

NOT USED AT THIS TIME

FirstChan

This register specifies the first channel to be used for commands that operate on a range of channels. This register, in combination with the **LastChan** register, can be used to specify a range of channels to be processed. For example, FirstChan = 0 and LastChan = 0 specifies only channel 0. Setting FirstChan = 0 and Lastchan = 5 specifies all 6 channels.

LastChan

This register specifies the last channel to be used for commands that operate on a range of channels. This register, in combination with the **LastChan** register, can be used to specify a range of channels to be processed. For example, FirstChan = 0 and LastChan = 0 specifies only channel 0. Setting FirstChan = 0 and Lastchan = 5 specifies all 6 channels.

MODULE VARIABLES

REGISTER NAME	DATA TYPE	R/W	PURPOSE
AvailChan	UINT32	R	Number of available channels
MasterSpan	IEEE-754	R/W	Master amplitude control for AC waveforms
ACPeriod	IEEE-754	R/W	Period in seconds for one AC waveform cycle
DCPeriod	IEEE-754	R/W	Period in seconds to complete a DC transition
FpChan	*UINT32*	*R/W*	*RESERVED NOT USED*
DigIn	*UINT32*	*R*	*RESERVED NOT USED*
DigOut	*UINT32*	*R/W*	*RESERVED NOT USED*
RampType	UINT32	R/W	Selects Ramp Type for DC transtions
TraceEnable	UINT32	R/W	Enables Trace Buffer logging
TraceNdx	UINT32	R/W	Current Index into Trace Buffer. Incremented by VDSP431
IntLock	INTLOCK	R/W	Interlock Status Word
TrigMode	UINT32	R/W	Select Internal or External Trigger Mode
CmdNdx	UINT32	R	Current Sinusoid Index Pointer (0-255)
AuxSetpoint	IEEE-754 ARRAY [6]	R	Setpoint panel knob values
DispChan	UINT32 ARRAY [4]	R	Current Readout Display 1-4 Channel
DispParm	UINT32 ARRAY [4]	R	Current Readout Display 1-4 Parameter
LimitStatus	UINT32	R	Limit Checking Status Word
LimitControl	UINT32	R/W	Limit Checking Control Word
ExtNullPace	UINT32	R	External Null Pace Indicator
IlockEnable	UINT32 ARRAY [5]	R/W	AC Interlock Enable for Interlock Inputs 1-5

AvailChan

This register indicates the number of active VDSP431 channels.

MasterSpan

This register controls the output amplitude of all sinewave generators. The output from each sinewave generator is multiplied by the value in this register forming a master span or amplitude control for all channels. This register is useful for soft starts to ramp up the AC level and for soft stops to ramp down the AC level. The range for MasterSpan is 0.0 to 1.0

ACPeriod

This register determines the period, in seconds, for one AC waveform cycle. The range of ACPeriod is .020 to 20.0 seconds. This range allows an AC frequency range of .05 to 50 Hz. The ACPeriod register is common to all VDSP431 channels.

DCPeriod

This register determines the period in seconds to complete a DC transition. The range of DCPeriod is .020 to 20.0 seconds. The DCPeriod register is common to all VDSP431 channels.

FpChan

NOT USED AT THIS TIME

DigIn

NOT USED AT THIS TIME

DigOut

NOT USED AT THIS TIME

RampType

This register determines what type of ramp will be used when the VDSP431 transitions from the current DC setpoint to a new DC setpoint. If RampType is a 0, Linear Ramp is selected. If RampType is a 1, Haver-Sine Ramp is selected.

TraceEnable

This register determines if the trace buffer is written to. If TraceEnable = 1, then data is written to the Trace Buffer. If TraceEnable = 0, then no data is written to the Trace Buffer.

TraceNdx

This register is a write pointer into the Trace Buffer. This register is updated by the VDSP431 firmware to show where it has placed data into the buffer. The VDSP431 firmware stores the command and feedback values for all channels into a circular buffer. These values are updated at the loop frequency of the servo.

The following structure located in shared RAM is a copy of hardware registers associated with the AC Power Module

IntLock

```
typedef struct interlock
{
      UINT32      InterlockStatus;    /* Register 0 */
      UINT32      ACPowerStatus;      /* Register 1 */
      UINT32      EnableStatus;       /* Register 2 */
      UINT32      FaultLimitStatus;   /* Register 3 */
      UINT32      KillACEnable;       /* Register 4 */
      UINT32      InterlockLatch;     /* Latched Register 0 */

} INTLOCK;
```

InterlockStatus

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
-	-	-	ILOCK5	ILOCK4	ILOCK3	ILOCK2	ILOCK1

ILOCK1 – ILOCK5

This bits reflect the state of the external interlock inputs. These bits = “1” when an interlock is asserted. These bits = “0” when an interlock is deasserted.

ACPowerStatus

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
PULST1	HIGH_HOLD	LOW_HOLD	ON_HOLD	HW_FAULT	AC_HIGH	AC_LOW	AC_ON

```
AC_ON        1=Power ON at “ON” outlet, 0=Power OFF
AC_LOW       1=Power ON at “LOW” outlet, 0=Power OFF
AC_HIGH      1=Power ON at “HIGH” outlet, 0=Power OFF
HW_FAULT     1=AC Hardware Fault detected, 0=OK
ON_HOLD      1=Hold Relay for “ON” Active, 0=Inactive
LOW_HOLD     1=Hold Relay for “LOW” Active, 0=Inactive
HIGH_HOLD    1=Hold Relay for “HIGH” Active, 0=Inactive
PULST1       Unused/Reserved
```

Example ACPowerStatus values read for different conditions:

“OFF”	“ON”	“LOW”	“HIGH”
0x70	0x61	0x43	0x07

EnableStatus

The host can read the enabled state of each of the five interlocks. The VDSP431 firmware enables all interlock sources by default. A “1” indicates enabled state.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	1	1	1	1	1

00011111 = 0x1F

FaultLimitStatus

The host can read the current fault and limit state of the other VDSP-431s in the system. A low signal represents a fault or limit condition:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Limit, VDSP-431 #4	Limit, VDSP-431 #3	Limit, VDSP-431 #2	Limit, VDSP-431 #1	Fault, VDSP-431 #4	Fault, VDSP-431 #3	Fault, VDSP-431 #2	Fault, VDSP-431 #1

KillACEnable

The host can read the enabled state of each of the interlock circuits in terms of removing all AC power to the switched circuits. By specification, each interlock can be programmed to remove all AC power to the switched AC circuits. A high bit indicates that the interlock will result in a loss of power to all AC circuits. A low bit indicates that it will not. Additionally, if any VDSP-431 in the system reports a fault or a limit condition, that condition can also result in shutting down the AC power circuits. The VDSP431 firmware enables all interlocks to remove AC power. Is is possible to enable or dislabed each interlock via software to "Bypass" the interlock.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
N/A	Kill AC VDSP-431 Limit	Kill AC VDSP-431 Fault	Kill AC Interlock 5	Kill AC Interlock 4	Kill AC Interlock 3	Kill AC Interlock 2	Kill AC Interlock 1

InterlockLatch

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
-	-	-	ILATCH5	ILATCH4	ILATCH3	ILATCH2	ILATCH1

ILATCH1 – ILATCH5

These bits reflect the latched state of the InterLockStatus register
These bits = "1" when an interlock is latches. These bits are reset to "0" following a FAULT_RESET command code.

TrigMode

This register selects either Internal or External Trigger Mode. This register is associated with the TRIGGER_STATE command.

CmdNdx

Current Sinusoid Index Pointer (0-255). This register will count from 0-255 over one cycle of a sinewave.

AuxSetpoint

Setpoint panel knob values for channels 1-6

DispChan

Each array element represents one VDSP431 Readout Panel.
Channel numbers range from 1-6

DispParm

Each array element represents one VDSP431 Readout Panel.
Parameter numbers range from 1-8

LIMITS CHECKING

ALARM LIMIT

Alarm Limit defines a symmetrical error voltage window. The absolute value of the control loop error signal is compared to this window. Alarm Limit checks are performed at a 1 kHz rate. Bits 0-5 of the Limit Status Word are used to indicate whether or not the control loop error is within the specified limit. The VDSP431 firmware will clear individual status flags if the error comes back within limits. Host intervention is not required. There is also an Alarm Filter specifies the number of consecutive limits that must occur before a limit is generated.

CRITICAL LIMIT

Critical Limit defines a symmetrical error voltage window. The absolute value of the control loop error signal is compared to this window. Critical Limit checks are performed at a 1 kHz rate. Bits 8-13 of the Limit Status Word are used to indicate whether or not the control loop error is within the specified limit. Critical Limit status flags are latched and can only be cleared via a Host acknowledge. There is also an Critical Filter specifies the number of consecutive limits that must occur before a limit is generated.

FEEDBACK LOWER LIMIT

Feedback Lower Limit defines an absolute voltage level to which the control loop feedback voltage is compared. Lower limit checks are performed at a 1 kHz rate. Bits 16-21 of the Limit Status Word are used to indicate whether or not the control loop error is within the specified limit. Feedback Lower Limit flags are latched and can only be cleared via a Host acknowledge. There is also an FbkLimit Filter specifies the number of consecutive limits that must occur before a limit is generated.

FEEDBACK UPPER LIMIT

Feedback Upper Limit defines an absolute voltage level to which to the control loop feedback voltage is compared. Upper limit checks are performed at a 1 kHz rate. Bits 24-28 of the Limit Status Word are used to indicate whether or not the control loop error is within the specified limit. Feedback Upper Limit flags are latched and can only be cleared via a Host acknowledge. There is also an FbkLimit Filter specifies the number of consecutive limits that must occur before a limit is generated.

LIMIT STATUS WORD

The Limit Status Word is a 32-bit word that contains limit flags for up to 6 channels. The host processor can obtain real-time status by simply polling this word in shared memory. This 32-bit word is divided into 4 groups of eight bits each. Each group of six bits corresponds to channels 1 - 6. The format of the Limit Status Word is as follows:

BITS	FUNCTION
0 – 5	Alarm Limit Bits are “1” if Alarm Limit has been exceeded on the corresponding channel
6 – 7	Reserved
8 – 13	Critical Limit Bits are “1” if Critical Limit has been exceeded on the corresponding channel
14 - 15	Reserved
16 – 21	Feedback Lower Limit Bits are “1” if Feedback Lower Limit has been exceeded on the corresponding channel
22 – 23	Reserved
24 – 29	Feedback Upper Limit Bits are “1” if Feedback Upper Limit has been exceeded on the corresponding channel
30 – 31	Reserved

LIMIT CONTROL WORD

The Limit Control Word is a 32-bit word that contains limit enables for up to 6 channels. The host processor can enable or disable individual channels by writing to this word in shared memory. This 32-bit word is divided into 4 groups of six bits each. Each group of six bits corresponds to channels 1 - 6. The format of the Limit Control Word is as follows:

BITS	FUNCTION
0 – 5	**Alarm Limit** a “1” enables Alarm Limit on the corresponding channel a “0” disables Alarm Limit checking
6 – 7	**Reserved**
8 – 13	**Critical Limit** a “1” enables Critical Limit on the corresponding channel a “0” disables Critical Limit checking
14-15	**Reserved**
16 - 21	**Feedback Lower Limit** a “1” enables Lower Limit checks on the corresponding channel a “0” disables Lower Limit checking
22 - 23	**Reserved**
24 – 29	**Feedback Upper Limit** a “1” enables Upper Limit checks on the corresponding channel a “0” disables Upper Limit checking
30 – 31	**Reserved**

LIMIT ASSERTED	ACTION
Alarm Limit	Stops way-point command generation until Alarm Limit is NOT asserted. ***Continues without host intervention***.
Critical Limit	Stops way-point command generation. ***Waits for host to acknowledge limit.***
Feedback Lower Limit	Stops way-point command generation. ***Waits for host to acknowledge limit.***
Feedback Upper Limit	Stops way-point command generation. ***Waits for host to acknowledge limit.***

ExtNullPace

External Null Pace Indicator

IlockEnable

AC Interlock Enable for Interlock Inputs 1-5
Use this register with the FAULT_ENABLE command code

Channel Variables

PARAMETER	TYPE	R/W	DESCRIPTION
Fbk	IEEE-754	R	Current feedback for channels 1–6
Fbk1	IEEE-754	R	Current feedback (LVDT) for channels 1–6
Fbk2	IEEE-754	R	Current feedback (BRIDGE) for channels 1–6
FbkCyc	IEEE-754	R	Feedback cyclic channels 1–6
FbkAvg	IEEE-754	R	Feedback average channels 1–6
Cmd	IEEE-754	R	Composite command volts for channels 1–6
Err	IEEE-754	R	Error value in volts for channels 1–6
Vout	IEEE-754	R	Valve output in volts for channels 1–6
DCOff	IEEE-754	R/W	DC Setpoint in volts for channels 1-6
ACamp	IEEE-754	R/W	AC Setpoint in volts for channels 1–6
ACphs	IEEE-754	R/W	AC Phase in degrees for channels 1–6
Kp	IEEE-754	R/W	Proportional Gain for channels 1-6
Ki	IEEE-754	R/W	Integral Gain for channels 1–6
Kd	IEEE-754	R/W	Derivative Gain for channels 1–6
Il	IEEE-754	R/W	Integration Limit volts for channels 1–6
Ds	UINT32	R/W	Derivative sample period for channels 1-6
CalRelay	UINT32	R/W	Calibration Relay Control for channels 1-6
Cond1Exc	IEEE-754	R/W	Signal Cond 1 Excitation channels 1–6
Cond1Gain	IEEE-754	R/W	Signal Cond 1 Gain channels 1-6
Cond1Offset	IEEE-754	R/W	Signal Cond 1 Offset channels 1-6
Cond1UseDCHI	UINT32	R/W	Signal Cond 1 Use DC High-Level 1-6
Cond2Exc	IEEE-754	R/W	Signal Cond 2 Excitation channels 1–6
Cond2Gain	IEEE-754	R/W	Signal Cond 2 Gain channels 1-6
Cond2Offset	IEEE-754	R/W	Signal Cond 2 Offset channels 1-6
Cond2UseDCHI	UINT32	R/W	Signal Cond 2 Use DC High-Level 1-6
DitherAmp	IEEE-754	R/W	Valve dither amplitude for channels 1-6
VlvRange	UINT32	R/W	Valve Range channels 1-6
VlvGain	IEEE-754	R/W	Valve Gain channels 1-6
FbkPol1	UINT32	R/W	Feedback 1 polarity control channels 1–6
FbkPol2	UINT32	R/W	Feedback 2 polarity control channels 1–6
FbkSel	UINT32	R/W	Feedback Select 1 or 2 channels 1-6
VlvPol	UINT32	R/W	Valve polarity control for channels 1-6
PidEnable	UINT32	R/W	Pid control enable for channels 1–6
VlvOffset	IEEE-754	R/W	Valve offset in volts for channels 1–6
CtrlStat	UINT32	R	Control/Status Bitfield channels 1–6
TuneEnable	UINT32	R/W	Tune squarewave enable channels 1–6
TuneAmplitude	IEEE-754	R/W	Tune square amplitude channels 1–6
TunePeriod	IEEE-754	R/W	Tune square period channels 1–6
FilterOn	UINT32	R/W	Filter On/Off channels 1-6
FilterFreq	IEEE-754	R/W	Filter Frequency channels 1-6
TraceCmdOffset	UINT32	R	Shared Memory offset to Command Traces 1-6
TraceFbkOffset	UINT32	R	Shared Memory offset to Feedback Traces 1-6
CoefA	IEEE-754	R/W	RESERVED DO NOT USE Filter A Coef
CoefB	IEEE-754	R/W	RESERVED DO NOT USE Filter b Coef

PARAMETER	TYPE	R/W	DESCRIPTION
NfilterCoef	UINT32	R/W	RESERVED Channels 1-6
FilterCoef	IEEE-754 [6][128]	R/W	RESERVED Channels 1-6
AlarmLimit	IEEE-754	R/W	Alarm Limit Channels 1-6
AlarmFilter	UINT32	R/W	Alarm Limit Filter Channels 1-6
CriticalLimit	IEEE-754	R/W	Critical Limit Channels 1-6
CriticalFilter	UINT32	R/W	Critical Filter Channels 1-6
FbkLowerLimit	IEEE-754	R/W	Feedback Lower Limit Channels 1-6
FbkUpperLimit	IEEE-754	R/W	Feedback Upper Limit Channels 1-6
FbkLimitFilter	UINT32	R/W	Feedback Limit Filter Channels 1-6

Fbk

This is an array of IEEE-754 registers that contain the current feedback value in volts for servo channels 1-6. The PID control loop will update this array at regular intervals. The range for Fbk is +/- 10.0 volts.

Fbk1

This is an array of IEEE-754 registers that contain the current LVDT feedback value in volts for servo channels 1-6. The PID control loop will update this array at regular intervals. The range is +/- 10.0 volts.

Fbk2

This is an array of IEEE-754 registers that contain the current BRIDGE feedback value in volts for servo channels 1-6. The PID control loop will update this array at regular intervals. The range is +/- 10.0 volts.

FbkCyc

This is an array of IEEE-754 registers that contain the current single amplitude AC feedback value in volts for servo channels 1-6. The PID control loop will update this array at regular intervals. The range is +/- 10.0 volts.

FbkAvg

This is an array of IEEE-754 registers that contain the average feedback value in volts for servo channels 1-6. The PID control loop will update this array at regular intervals. The range is +/- 10.0 volts.

Cmd

This is an array of IEEE-754 registers that contain the current composite command value in volts for servo channels 1-6. The composite command is a summation of the DC and AC setpoint generators. The PID control loop will update this array at regular intervals. The range for Cmd is +/- 10.0 volts.

Err

This is an array of IEEE-754 registers that contain the current error signal for servo channels 1-6. The error signal represents the difference between the FBK and CMD values. The PID control loop will update this array at regular intervals. The range for ERR is +/- 20.0 volts.

VOut

This is an array of IEEE-754 registers that contain the current valve output voltage for channels 1-6. The valve output is a function of the PID calculation, valve offset and dither amplitude. The PID control loop will update this array at regular intervals.

DCOff

This is an array of IEEE-754 registers that determine the DC setpoint or offset for servo channels 1-6. These values are used to command the DC setpoint generator to ramp to a new DC target level. The move to the new DC level will be paced by the value specified in the DCPeriod register. The range for DCOff is +/- 10.0 volts.

ACamp

This is an array of IEEE-754 registers that determine the AC amplitude or span for servo channels 1-6. These values are used to command the AC setpoint generator to output a new AC amplitude. The period of the AC waveform is determined by the value written to the ACPeriod register. The range for ACamp is 0.0 to 10.0 volts. The MasterSpan control can be used to scale the amplitude of all channels.

ACphs

This is an array of IEEE-754 registers that determine the AC phase for servo channels 1-6. These values are used to command the AC setpoint generator to output a new AC phase. The phase values are relative to channel 0. The range for ACphs is 0.0 to 360.0 degrees.

Kp

This is an array of IEEE-754 registers that specify the Proportional Gain (Kp) factor used by the PID control loop for channels 1-6. The range for KP is 0.0 - 100.0.

Ki

This is an array of IEEE-754 registers that specify the Integral Gain (Ki) factor used by the PID control loop for channels 1-6. The range for KI is 0.0 -100.0.

Kd

This is an array of IEEE-754 registers that specify the Derivative Gain (Kd) factor used by the PID control loop for channels 1-6. The range for KD is 0.0 -100.0.

Il

This is an array of IEEE-754 registers that specify the Integration Limit factor used by the PID control loop for channels 1-6. The range for IL is 0.0 -10.0.

Ds

This is an array of integer registers that specify the Derivative Sampling interval used by the PID control loop for channels 1-6. The range for DS is 2 - 32.

CalRelay

This is an array of integer registers that specify the state of the calibration relays for servo channels 1-6. 1 = Relay Closed, 0 = Relay Open.

Cond1Exc

This is an array of IEEE-754 registers that specify the DC bridge excitation voltage output for servo channels 1-6. The range for Excitation is 0.0 to 10.0 volts.

Cond1Gain

This is an array of IEEE-754 registers that specify the gain range for programmable instrumentation amplifiers on channels 1-6.

Cond1Offset

This is an array of IEEE-754 registers that specify a PGA offset voltage to be applied to channels 1-6. The range for PgaOffset is -10.0 to +10.0.

Cond1UseDCHI

This is an array of integer registers that specify if the LVDT demodulator is to be bypassed to support DC high-level type input. 1=Use DC high-level, 0=LVDT

Cond2Exc

This is an array of IEEE-754 registers that specify the DC bridge excitation voltage output for servo channels 1-6. The range for Excitation is 0.0 to 10.0 volts.

Cond2Gain

This is an array of IEEE-754 registers that specify the gain range for programmable instrumentation amplifiers on channels 1-6.

Cond2Offset

This is an array of IEEE-754 registers that specify a PGA offset voltage to be applied to channels 1-6. The range for PgaOffset is -10.0 to +10.0.

Cond1UseDCHI

This is an array of integer registers that specify if the LVDT demodulator is to be bypassed to support DC high-level type input. 1=Use DC high-level, 0=LVDT

DitherAmp

This is an array of IEEE-754 registers that determine the level of valve dither applied to PID control channels 1-6. The range for DitherAmp is 0.0 to 10.0 volts. The dither frequency is 500 Hz.

VlvRange

This is an array of integer registers that determine the valve gain range for channels 1-6.

VlvGain

This is an array of IEEE-754 registers that determine the valve gain ratio for channels 1-6. A value of 1.0 = 100% of valve driver current setting.

FbkPol1

This is an array of integer registers that determine the feedback polarity for channels 1-6. When FbkPol = 0, polarity = +1.0. When FbkPol = 1, polarity = -1.0. The polarity value is multiplied by the incoming feedback voltage before use

by the PID control. FbkPol can be used to compensate for polarity reversal in field wiring.

FbkPol2

This is an array of integer registers that determine the feedback polarity for channels 1-6. When FbkPol = 0, polarity = +1.0. When FbkPol = 1, polarity = -1.0. The polarity value is multiplied by the incoming feedback voltage before use by the PID control. FbkPol can be used to compensate for polarity reversal in field wiring.

FbkSel

This is an array of integer registers that determine the which feedback signal conditioner will be used as primary feedback for the servo loop.

VlvPol

This is an array of integer registers that determine the feedback polarity for channels 1-6. When VlvPol = 0, polarity = +1.0. When VlvPol = 1, polarity = -1.0. The polarity value is multiplied by the output of the PID control block before being sent to the valve D/A. VlvPol can be used to compensate for polarity reversal in field wiring.

PidEnable

This is an array of integer registers that determine if a PID control channel is enabled on channels 1-6. When PidEnable = 1, the PID control is enabled. When PidEnable = 0, the PID control is disabled. PID channels that are disabled drive the servo error to zero by setting the composite command equal to the feedback signal. Valve dither and valve offset are still applied during this state.

VlvOffset

This is an array of IEEE-754 registers that specify the valve offset in volts for PID control channels 1-6. The range for VlvOffset is -10.0 to 10.0 volts. VlvOffset can be used to electronically balance valves to zero flow.

CtrlStat

TuneEnable

This is an array of integer registers that determine if the Tuning square wave generator is enabled on channels 1-6. When TuneEnable = 1, the generator is enabled. When TuneEnable = 0, the generator is disabled.

TuneAmplitude

This is an array of IEEE-754 registers that specify the square wave generator amplitude in volts.

TunePeriod

This is an array of IEEE-754 registers that specify the square wave generator period in seconds.

FilterOn

This is an array of integer registers that determine if the digital LowPass filter is enabled on channels 1-6. When FilterOn = 1, the filter is enabled. When FilterOn = 0, the filter is disabled.

FilterFreq

This is an array of IEEE-754 registers that specify the filter cutoff frequency in hertz.

TraceCmdOffset

These registers contains an offset to the start of the Trace Command Buffer for channels 1-6. The offset is the number of 32-bit words from the start of the dual-access ram located A32 space.

TraceFbkOffset

These registers contains an offset to the start of the Trace Feedback Buffer for channels 1-6. The offset is the number of 32-bit words from the start of the dual-access ram located A32 space.

CoefA

NOT USED CURRENTLY

CoefB

NOT USED CURRENTLY

NfilterCoef
RESERVED

FilterCoef
RESERVED

AlarmLimit
Alarm Limit Channels 1-6

AlarmFilter
Alarm Limit Filter Channels 1-6

CriticalLimit
Critical Limit Channels 1-6

CriticalFilter
Critical Filter Channels 1-6

FbkLowerLimit
Feedback Lower Limit Channels 1-6

FbkUpperLimit
Feedback Upper Limit Channels 1-6

FbkLimitFilter
Feedback Limit Filter Channels 1-6

SYNCHRONIZING MULTIPLE VDSP431 BOARDS

There is an internal timer that each VDSP431 uses to control the time between points in a sinusoid. When the timer expires, the 431 moves to the next point in a lookup table. In order to synchronize multiple 431 boards, one VDSP431 acts as the Master. The "Master" will be programmed to generate an interrupt when it is time to output the next waveform point. The VMEbus host will then acknowledge this interrupt by writing to a Global Command register located in A16 space. Following the write, every VDSP431 will receive an interrupt at the same time. For a 20 Hertz sinusoid, the VMEbus host would be interrupted at approximately 195 microsecond intervals. The VMEbus host is only required to write one 16-bit word inside of the interrrupt service routine. The other side benefit is that the VMEbus host is also "in-sync" with the control system, providing a mechanism to trigger data acquisition hardware in the system.

Program the GMBAR of each VDSP431 in the system as outlined below:

The GMBAR (A16 BASE ADDRESS + $07) register defines a Global A16 Mailbox address that multiple VDSP431 boards can respond to. The VMEbus can communicate global information to one or more VDSP431 boards. This register is write only and cannot be read back. Software must maintain a shadow copy of the register contents. The GMBAR decoder must be enabled, by setting bit 1 in the VCTRL register . The GMBAR register is defined below:

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
A15	A14	A13	A12	A11	A10	A09	A08

For example, writing the value 0xFF to this register would program the VDSP431 to listen at A16 address 0xFF00.

Program the VCTRL register of each VDSP431 in the system as outlined below:

Pick one of the VDSP431 boards as the “Master”. This is usually the first board at the lowest address but it does not have to be. This is the only board that should have Bit 02 set. This is also the only board that will interrupt the VMEbus host. Enable the 431 to listen to the Global Address by setting Bit 01. Take care to leave Bit 00 clear, otherwise the VDSP431 will be reset. All other VDSP431 boards should be configured as servants by only setting Bit 01.

The VCTRL (A16 BASE ADDRESS + $0F) register is used to :

Reset the VDSP431. This register can hold the VDSP431 hardware in a reset state that is independent of the VMEbus SYSRESET. This register is write only and cannot be read back. Software must maintain a shadow copy of the register contents.

Enable the Global Address Register
Enable the Master mode for the VDSP431 to generate a VMEbus DTACK.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
-	-	-	-	-	MASTER	GLOBAL ENABLE	RST

RST

When RST = 1, the VDSP431 is held in Reset.
When RST = 0, the VDSP431 is released from Reset.

GLOBAL ENABLE

When set to a ”1”, the VDSP431 will generate a local interrupt if there is an A16 access to the address specified by GMBAR. When set to a “0”, accesses are ignored.

MASTER

If set to a “1”, the VDSP431 module generates a DTACK acknowledge when an access is made to the address specified by GMBAR. Only one VDSP431 module should be programmed with this bit set to a “1”. Additional VDSP431 boards must be programmed with this bit set to a “0”.

At this point, you should be able to verify the ability of each VDSP431 to receive a Global Command. Use the VMEbus host to write to the A16 Global Address (0xFF00 in this example). The CMD Led should illuminate on each VDSP431 at the same time.

First, program all VDSP431 servants to select EXTERNAL_TRIGGER. Then, program the “Master” VDSP431 last:

The TRIGGER_STATE command is now used to select either INTERNAL_TRIGGER or EXTERNAL_TRIGGER. When the VDSP431 processes this command, the waveform pointer is reset to 0.

The following define is used when sending the TRIGGER_STATE command:

```
#define TRIGGER_STATE       (13)
```

Set the TrigMode variable equal to one of the following values:

```
#define INTERNAL_TRIGGER    (0)
#define EXTERNAL_TRIGGER    (1)


/* Host module structure */


typedef struct host_module
{

    UINT32      AvailChan;      /* Number of available channels */

    IEEE_FLOAT  MasterSpan;     /* span control all channels */

    IEEE_FLOAT  ACPeriod;       /* period for AC waveforms */
    IEEE_FLOAT  DCPeriod;       /* period for DC transitions */
    UINT32      FpChan;         /* front panel channel selection */
    UINT32      DigIn;          /* Digital Input Image */
    UINT32      DigOut;         /* Digital Output Image */
    UINT32      RampType;       /* 0 = Linear, 1 = HaverSine */
    UINT32      TraceEnable;    /* 1 = Record, 0=Off */
    UINT32      TraceNdx;       /*  Current position in Trace Buffers
*/
      INTLOCK     IntLock;
    UINT32      TrigMode;       /* INTERNAL_TRIGGER or EXTERNAL_TRIGGER
*/
    UINT32      CmdNdx;         /* Waveform Command Index 0-255 */

} HMOD;
```

The variable CmdNdx points to the current waveform point. This variable increments from 0 to 255 as triggers are received.