Greenspring IP-DAC-SU

# 16-Channel DAC Simultaneous Update IndustryPack

**In Stock**

**Used and in Excellent Condition**

**Open Web Page**

https://www.artisantg.com/95831-1

# IP-DAC-SU

## 16 Channel DAC
## Simultaneous Update
## IndustryPack®
## Manual

## User Manual


Hardware Revision C
Part # 89002062 Rev. 1.0
20050124

**IP-DAC-SU**

**16 Channel DAC**
**Simultaneous Update**
**IndustryPack®**

**SBS Technologies, Inc.**
**1284 Corporate Center Drive**
**St. Paul, MN 55121**
**Tel (651) 905-4700**
**FAX (651) 905-4701**
**Email: support.commercial@sbs.com**
**http://www.sbs.com**

# Table of Contents

# List of Figures

# Product Description

Each IP–DAC SU provides up to 16 channels of simultaneously updated digital to analog conversion. Crystal Semiconductor CS4333 serial DACs are featured for audio quality 16–bit analog conversion. IP–DAC SU is optimized for high density providing up to 64 DAC channels per 6U VME slot. Multiple IP–DAC SUs can be ganged and updated simultaneously by means of an RS–485 sync signal. Up to 33 IPs, 528 channels, can be updated simultaneously. Output level shifting and offset correction is user programmable by means of a separate DAC channel. The 16 channel IP–DAC SU is a Type II IndustryPack. For maximum channel/slot density, low profile surface mounted components are used on both sides of the IP. A lower cost 8 channel Type I version is also available.

The Crystal Semiconductor CS4333 stereo DACs combine modern audio quality performance in sub–miniature packaging. The DACs provide 16 bit resolution in 2's complement format and are designed with high fidelity audio applications in mind. Crystal specifies 94 dB dynamic range, 0.003% THD, 1–bit D/A conversion and ultra linear analog low pass filtering. IP–DAC SU complements the Crystal DAC's performance with RF filtered analog power, an isolated analog ground plain and hand crafted layout to optimize the analog stages. Linear Technology LT1114 precision picoamp input Op Amps and ratiometrically designed bias circuitry provide uniform, low drift, analog output with sub–miniature surface mount components.

IP–DAC SUs are designed to be ganged for multiple IP simultaneous updating. All analog outputs on the Master IP and Slave IPs are updated simultaneously via an RS–485 sync signal. One Master can drive up to 32 Slave IPs for a total of 33 IPs, or 528 channels. Each IP is software configurable as master or slave. The Master IP can also provide an interrupt after update cycle completion for interrupt driven high throughput applications. IP–DAC SU is capable of sustained throughput rates up to 10 kHz.

IP–DAC SU output levels are user programmable on a per IP basis. Output level and offset are adjustable, on the fly, with an independent dedicated DAC channel. Programmable levels provide standard output ranges of: 0 to +10, ±5 and $-10$ to 0 Volts. Other ranges are available as customer specials. The leveling circuit is designed to automatically track and cancel thermal drift and offsets.

Advanced Xilinx FPGA SRAM cells are used to double buffer the DAC data. A full 16 bit wide, no–wait state data path is provided for efficiency. Texas Instruments SN75LBC176 transceivers provide easy hook–up of the RS–485 sync signal. No null modem or "flipping" required. Additionally, a 2K x 16 user programmable EEPROM is provided with calibration data.

The combination of simultaneous update, expandability to hundreds of channels, high channel/slot density and audio quality analog performance make IP–DAC SU ideal for large simulator, control and waveform synthesis projects.

## IP-DAC-SU Key Features

• 16 Digital to Analog channels per IP.
• Simultaneous Update, all channels, all IPs.
• Expandable to 528 channels.
• Up to 64 channels per 6U slot.
• 16–bit resolution.
• Audio quality DACs.
• On board calibration EEPROM.
• Ideal for control, simulation and waveform synthesis.
• Pacer interrupt for high throughput.
• 10 kHz maximum update rate.
• 16 bit, no wait state, data path.
• All units may be master or slave.
• Low cost per channel.
• Programmable output level

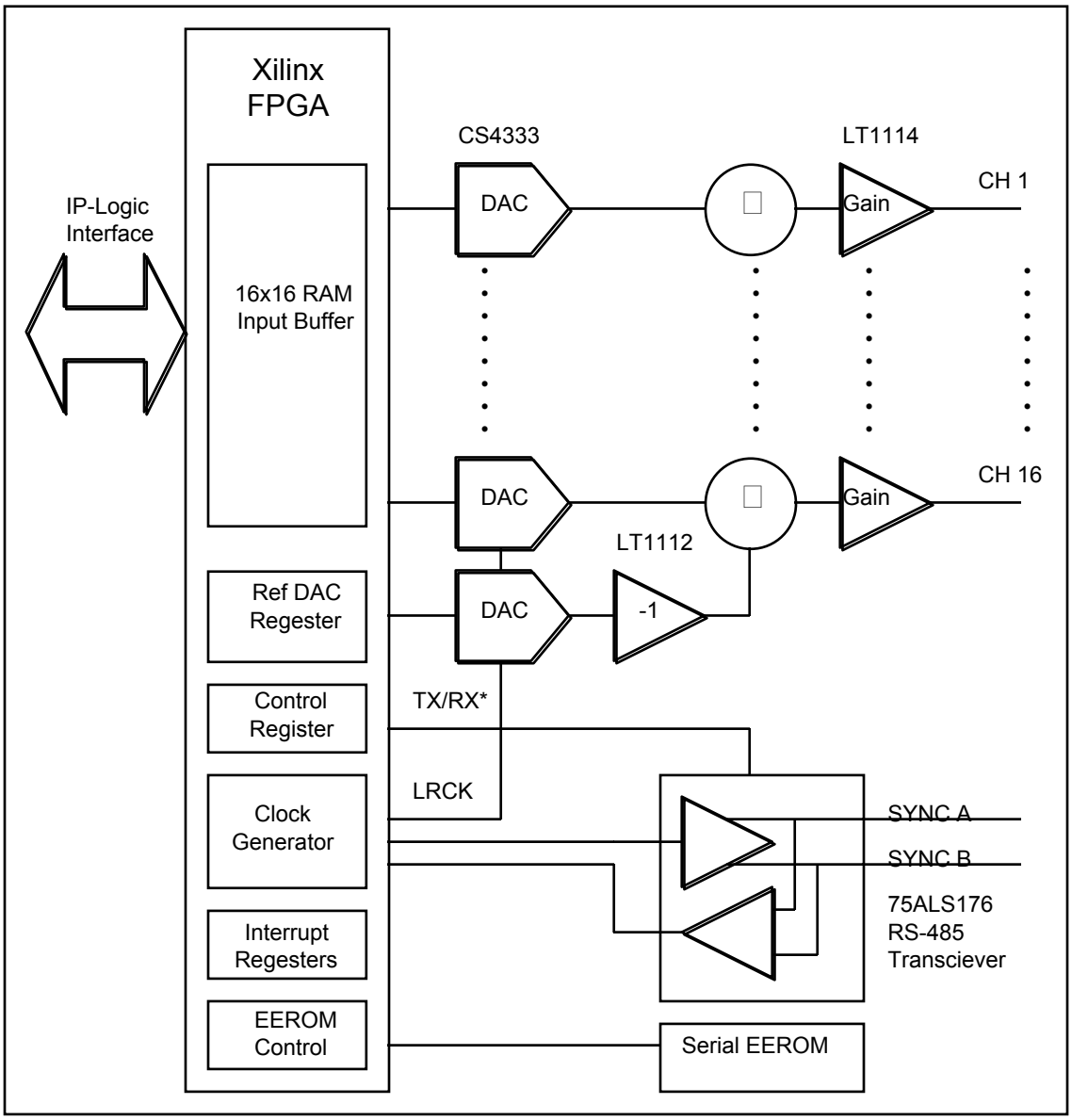


**Figure 1 IP–DAC SU Simplified Block Diagram**

# VME Addressing

All registers are both byte and word accessible. A full 16 bit data path is provided and word access is recommended for maximum throughput. The offsets between registers are on uniform longword boundaries for programming convenience. Generally the power up default value of most registers is 0. However, Xilinx does not guarantee the contents of the SRAM on power up. Therefore initialization of the DAC data is required. Please see the Programmers Model for more detail.

Standard Word I/O Accessing

| Offset (hex) | Name | Register |
|---|---|---|
| Base + 0 | CON | Control |
| Base + 4 | REF | Reference DAC |
| Base + 8 | EED | EEROM Data |
| Base + C | EEC | EEROM Control |
| Base + 10 | INTD | Interrupt Data |
| Base + 14 | INTP | Interrupt Polarity |
| Base + 18 | INTEN | Interrupt Enable |
| Base + 1C | INTCLR | Interrupt Clear |
| Base + 20 | INTPND | Interrupt Pending |
| Base + 24 | VECTOR | Interrupt Vector |
| Base + 40 | CH1 | Channel 1 data |
| Base + 44 | CH2 | Channel 2 date |
| Base + 48 | CH3 | Channel 3 data |
| Base + 4C | CH4 | Channel 4 data |
| Base + 50 | CH5 | Channel 5 data |
| Base + 54 | CH6 | Channel 6 data |
| Base + 58 | CH7 | Channel 7 data |
| Base + 5C | CH8 | Channel 8 data |
| Base + 60 | CH9 | Channel 9 data |
| Base + 64 | CH10 | Channel 10 data |
| Base + 68 | CH11 | Channel 11 data |
| Base + 6C | CH12 | Channel 12 data |
| Base + 70 | CH13 | Channel 13 data |
| Base + 74 | CH14 | Channel 14 data |
| Base + 78 | CH15 | Channel 15 data |
| Base + 7C | CH16 | Channel 16 data |

**Figure 2 VME Word Access**

# NuBus Addressing

All registers are both byte and word accessible. A full 16 bit data path is provided and word access is recommended for maximum throughput. The offsets between registers are on uniform longword boundaries for programming convenience. Generally the power up default value of most registers is 0. However, Xilinx does not guarantee the contents of the SRAM on power up. Therefore initialization of the DAC data is required. Please see the Programmers Model for more detail.

NuBus addressing can calculated from VME addressing:
NuBus address = (VME address x 2)

Standard Word I/O Accessing

| Offset (hex) | Name | Register |
|---|---|---|
| Base + 0 | CON | Control |
| Base + 8 | REF | Reference DAC |
| Base + 10 | EED | EEROM Data |
| Base + 18 | EEC | EEROM Control |
| Base + 20 | INTD | Interrupt Data |
| Base + 28 | INTP | Interrupt Polarity |
| Base + 30 | INTEN | Interrupt Enable |
| Base + 38 | INTCLR | Interrupt Clear |
| Base + 40 | INTPND | Interrupt Pending |
| Base + 48 | VECTOR | Interrupt Vector |
| Base + 50 | CH1 | Channel 1 data |
| Base + 58 | CH2 | Channel 2 date |
| Base + 60 | CH3 | Channel 3 data |
| Base + 68 | CH4 | Channel 4 data |
| Base + 70 | CH5 | Channel 5 data |
| Base + 78 | CH6 | Channel 6 data |
| Base + 80 | CH7 | Channel 7 data |
| Base + 88 | CH8 | Channel 8 data |
| Base + 90 | CH9 | Channel 9 data |
| Base + A0 | CH10 | Channel 10 data |
| Base + A8 | CH11 | Channel 11 data |
| Base + B0 | CH13 | Channel 13 data |
| Base + B8 | CH14 | Channel 14 data |
| Base + C0 | CH15 | Channel 15 data |
| Base + C8 | CH16 | Channel 16 data |

**Figure 3 Nubus Word Access**

# ISA (IBM PC–AT) Addressing

All registers are both byte and word accessible. A full 16 bit data path is provided and word access is recommended for maximum throughput. The offsets between registers are on uniform longword boundaries for programming convenience. Generally the power up default value of most registers is 0. However, Xilinx does not guarantee the contents of the SRAM on power up. Therefore initialization of the DAC data is required. Please see the Programmers Model for more detail.

IP–DAC SU is normally accessed one word at a time in the host's I/O space. Byte ordering is a function of CPU family, bus architecture and IP Carrier. The byte order may be reversed for Intel Processors or ISA bus carriers.

Standard Word I/O Accessing

| Offset (hex) | Name | Register |
|---|---|---|
| Base + 0 | CON | Control |
| Base + 4 | REF | Reference DAC |
| Base + 8 | EED | EEROM Data |
| Base + C | EEC | EEROM Control |
| Base + 10 | INTD | Interrupt Data |
| Base + 14 | INTP | Interrupt Polarity |
| Base + 18 | INTEN | Interrupt Enable |
| Base + 1C | INTCLR | Interrupt Clear |
| Base + 20 | INTPND | Interrupt Pending |
| Base + 24 | VECTOR | Interrupt Vector |
| Base + 40 | CH1 | Channel 1 data |
| Base + 44 | CH2 | Channel 2 date |
| Base + 48 | CH3 | Channel 3 data |
| Base + 4C | CH4 | Channel 4 data |
| Base + 50 | CH5 | Channel 5 data |
| Base + 54 | CH6 | Channel 6 data |
| Base + 58 | CH7 | Channel 7 data |
| Base + 5C | CH8 | Channel 8 data |
| Base + 60 | CH9 | Channel 9 data |
| Base + 64 | CH10 | Channel 10 data |
| Base + 68 | CH11 | Channel 11 data |
| Base + 6C | CH12 | Channel 12 data |
| Base + 70 | CH13 | Channel 13 data |
| Base + 74 | CH14 | Channel 14 data |
| Base + 78 | CH15 | Channel 15 data |
| Base + 7C | CH16 | Channel 16 data |

**Figure 4 ISA (IBM PC–AT) Word Access**

# Getting Started

This section is intended to get first time users started with IP-DAC-SU, quickly. The following steps will verify IP-DAC-SU operation and installation in just a few minutes. The procedure uses a low level debugger to program the IP. The following are screen captures using a Motorola MVME162 CPU and a SBS VIPC610 carrier., both of which are in the factory default configurations. The debugger is Motorola's MVME 162 Bug, which is standard with there CPU. These examples are intended as a guide and other systems will be different.

In the first example an IP-DAC-SU is loaded into Slot A of the VIPC610 and programmed for a 2.5 Volt analog output. The procedure has four steps:

Step 1: Read the IP I/O space and IDPROM to verify correct installation.

Step 2: Set the DAC Data Register value to a mid scale value of 0x3FFF.

Step 3: Set the UPDATE bit to start data conversion..

Step 4: Probe I/O pins 2 & 3 with a voltmeter to verify analog output.



MVME162 Debugger/Diagnostics Release Version 2.1 - 10/19/93
COLD Start

Local Memory Found =01000000 (&16777216)

MPU Clock Speed =25Mhz

```
162-Bug>md ffff6000 ffff6100       /*Display IP Slot A contents, IDPROM is correct*/
FFFF6000  FFF0 FFFF 0000 FFFF  FFFE FFFF FFFE FFFF  ................
FFFF6010  FFFF FFFF FFFF FFFF  FFFE FFFF FFFF FFFF  ................
FFFF6020  FFFF FFFF FF0F FFFF  FFFD FFFF FFFF FFFF  ................
FFFF6030  FFFF FFFF FFFF FFFF  FFFF FFFF FFFF FFFF  ................
FFFF6040  0000 FFFF 0000 FFFF  0000 FFFF 0000 FFFF  ................
FFFF6050  0000 FFFF 0000 FFFF  0000 FFFF 0000 FFFF  ................
FFFF6060  0000 FFFF 0000 FFFF  0000 FFFF 0000 FFFF  ................
FFFF6070  0000 FFFF 0000 FFFF  0000 FFFF 0000 FFFF  ................
FFFF6080  FF49 FF50 FF41 FF43  FFF0 FF4F FFC1 FF00  .I.P.A.C...O....
FFFF6090  FF00 FF00 FF0C FFC4  FF00 FF00 FF00 FF00  ................
FFFF60A0  FF49 FF50 FF41 FF43  FFF0 FF4F FFC1 FF00  .I.P.A.C...O....
FFFF60B0  FF00 FF00 FF0C FFC4  FF00 FF00 FF00 FF00  ................
FFFF60C0  FF49 FF50 FF41 FF43  FFF0 FF4F FFC1 FF00  .I.P.A.C...O....
FFFF60D0  FF00 FF00 FF0C FFC4  FF00 FF00 FF00 FF00  ................
FFFF60E0  FF49 FF50 FF41 FF43  FFF0 FF4F FFC1 FF00  .I.P.A.C...O....
FFFF60F0  FF00 FF00 FF0C FFC4  FF00 FF00 FF00 FF00  ................
162-Bug>m ffff6040  /*Load DAC Data Register*/
FFFF6040 0000? 3fff /*3FFF ≈ 2.5 Volts*/
FFFF6042 FFFF? .
162-Bug>m ffff6000  /*Set UDATE bit*/
FFFF6000 FFF0? 1
**WARNING:NO MATCH**          /*162-Bug Warning normal*/
FFFF6000 FFF0?       /*UPDATE bit automatically reset*/
```

The second example shows multiple IP operation. The master IP is loaded in slot A and the slave IP is loaded in slot B. The procedure has six steps:

Step 1: First connect the RS-485 sync signals. Connect IP A P2.27 to IP B P2.27 and IP A P2.29 to IP B P2.29.

Step 2: Read the IP I/O spaces and IDPROMs to verify correct installation..

Step 3: Put the IP in slot B to slave mode.

Step 4: Load DAC Data registers.

Step 5: Turn on Master IP transmitter and start update cycle.

Step 6: Probe I/O pins 2 & 3 of both IPs with a voltmeter to verify analog output.

```
Copyright Motorola Inc. 1988 - 1993, All Rights Reserved

MVME162 Debugger/Diagnostics Release Version 2.1 - 10/19/93
COLD Start

Local Memory Found =01000000 (&16777216)

MPU Clock Speed =25Mhz

162-Bug>md ffff6000 ffff6200      /*Display Slots A&B and correct result*/
FFFF6000  FFF0 FFFF 0000 FFFF  FFFE FFFF FFFE FFFF  ................
FFFF6010  FFFF FFFF FFFF FFFF  FFFE FFFF FFFF FFFF  ................
FFFF6020  FFFF FFFF FF0F FFFF  FFFD FFFF FFFF FFFF  ................
FFFF6030  FFFF FFFF FFFF FFFF  FFFF FFFF FFFF FFFF  ................
FFFF6040  0000 FFFF 0000 FFFF  0000 FFFF 0000 FFFF  ................
FFFF6050  0000 FFFF 0000 FFFF  0000 FFFF 0000 FFFF  ................
FFFF6060  0000 FFFF 0000 FFFF  0000 FFFF 0000 FFFF  ................
FFFF6070  0000 FFFF 0000 FFFF  0000 FFFF 0000 FFFF  ................
FFFF6080  FF49 FF50 FF41 FF43  FFF0 FF4F FFC1 FF00  .I.P.A.C...O....
FFFF6090  FF00 FF00 FF0C FFC4  FF00 FF00 FF00 FF00  ................
FFFF60A0  FF49 FF50 FF41 FF43  FFF0 FF4F FFC1 FF00  .I.P.A.C...O....
FFFF60B0  FF00 FF00 FF0C FFC4  FF00 FF00 FF00 FF00  ................
FFFF60C0  FF49 FF50 FF41 FF43  FFF0 FF4F FFC1 FF00  .I.P.A.C...O....
FFFF60D0  FF00 FF00 FF0C FFC4  FF00 FF00 FF00 FF00  ................
FFFF60E0  FF49 FF50 FF41 FF43  FFF0 FF4F FFC1 FF00  .I.P.A.C...O....
FFFF60F0  FF00 FF00 FF0C FFC4  FF00 FF00 FF00 FF00  ................
FFFF6100  FFF0 FFFF 0000 FFFF  FFFF FFFF FFFE FFFF  ................
FFFF6110  FFFF FFFF FFFF FFFF  FFFE FFFF FFFF FFFF  ................
FFFF6120  FFFF FFFF FF0F FFFF  FFFD FFFF FFFF FFFF  ................
FFFF6130  FFFF FFFF FFFF FFFF  FFFF FFFF FFFF FFFF  ................
FFFF6140  0000 FFFF 0000 FFFF  0000 FFFF 0000 FFFF  ................
FFFF6150  0000 FFFF 0000 FFFF  0000 FFFF 0000 FFFF  ................
FFFF6160  0000 FFFF 0000 FFFF  0000 FFFF 0000 FFFF  ................
FFFF6170  0000 FFFF 0000 FFFF  0000 FFFF 0000 FFFF  ................
FFFF6180  FF49 FF50 FF41 FF43  FFF0 FF4F FFC1 FF00  .I.P.A.C...O....
FFFF6190  FF00 FF00 FF0C FFC4  FF00 FF00 FF00 FF00  ................
FFFF61A0  FF49 FF50 FF41 FF43  FFF0 FF4F FFC1 FF00  .I.P.A.C...O....
FFFF61B0  FF00 FF00 FF0C FFC4  FF00 FF00 FF00 FF00  ................
FFFF61C0  FF49 FF50 FF41 FF43  FFF0 FF4F FFC1 FF00  .I.P.A.C...O....
FFFF61D0  FF00 FF00 FF0C FFC4  FF00 FF00 FF00 FF00  ................
```

```
FFFF61E0  FF49 FF50 FF41 FF43  FFF0 FF4F FFC1 FF00  .I.P.A.C...O....
FFFF61F0  FF00 FF00 FF0C FFC4  FF00 FF00 FF00 FF00  ................
162-Bug>m ffff6100  /*Put IP in B in Slave Mode*/
FFFF6100 FFF0? 2
**WARNING:NO MATCH**          /*162-Bug Warning normal*/
FFFF6100 FFF2? .
162-Bug>m ffff6040  /*Load IP A DAC reg*/
FFFF6040 0000? 3fff
FFFF6042 FFFF? .
162-Bug>m ffff6140  /*Load IP B DAC reg*/
FFFF6140 0000? 3fff
FFFF6142 FFFF? .
162-Bug>m ffff6000  /*Turn on IP A transmitter*/
FFFF6000 FFF0? 5    /*and start update*/
**WARNING:NO MATCH**          /*162-Bug Warning normal*/
FFFF6000 FFF4?      /*UPDATE bit automatically reset*/
```

# Programming

IP–DAC SU has three register groups; the utility registers, the interrupt control registers and the sixteen DAC data registers. The utility registers control operating modes of the IP, the reference DAC level and calibration EEPROM access. The interrupt group controls interrupts and the vector. All utility registers and interrupt registers are flip–flop based latches. Generally, the power up default value of these registers is 0. The exceptions are the IPEN, INTCLR and the VECTOR registers. The sixteen DAC data registers are implemented with Xilinx FPGA SRAM cells. Xilinx does not guarantee a power up state for the SRAM cells and therefore they must be initialized .

## The Utility Group

### CON: The Control Register

The control register initializes updating and selects the mode of operation. There are four bits in the control register;

| BIT | NAME | ACCESS | FUNCTION |
|---|---|---|---|
| D0 | UPDATE | R | 0 = NO UPDATE REQUESTED |
| | | R | 1 = UPDATE REQUESTED |
| | | W | 0 = NO EFFECT |
| | | W | 1 = UPDATE DACS |
| D1 | MASTER* | R/W | 0 = MASTER MODE |
| | | R/W | 1 = SLAVE MODE |
| D2 | TX/RX* | R/W | 0 = RS–485 RECEIVER MODE |
| | | R/W | 1 = RS–485 TRANSMITTER MODE |
| D3 | CNTR | R/W | 0 = COUNTERS RUN |
| | | R/W | 1 = COUNTERS RESET |
| D4–D15 | R/O | NOT USED | |

**Figure 5 Control Register Bits**

UPDATE: Writing a 1 to this bit starts the DAC data conversion. Once the data conversion has started, the UPDATE bit will be automatically cleared. The UPDATE bit may be reset at anytime during data conversion. Use the INTD register to determine conversion cycle status. In multiple IP operation setting the UPDATE bit of Master IP starts conversion for all IPs. Setting the UPDATE bit an IP in slave mode causes just the slave IP to update. Writing a 0 to this bit has no effect. The power up default value of the UPDATE bit is 0.

MASTER*: This bit selects master or slave mode. Writing a 0 to this bit puts the IP in master mode. Writing a 1 to this bit puts the IP in slave mode. For operation of a single IP master mode is recommended. In multiple IP operation only one IP is allowed be the master. All others must be slaves. When an IP is switched to slave mode it will sync up with the Master IP's clocks after the first update cycle. The TX/RX* Bit must be set to 0 for all slaves. The power up default value of the MASTER* bit is 0.

TX/RX*: This bit sets the data direction of the RS–485 transceiver. Writing a 1 to this bit configures the transceiver as a transmitter. The transmitter mode is used by the Master IP to send the update sync signal. Writing a 0 to this bit configures the transceiver as a receiver. Receiver mode is used by all Slave IPs. Slave IPs must be in receive mode to be synchronized with the master. To avoid possible contention on the RS–485 line, the power up value of this bit is 0 (i.e. receive mode). When an IP is in master mode any RS–485 signals received are ignored. It is acceptable to leave this bit 0 for single IP operation. Using transmitter mode on a Slave IP is meaningless and not supported.

| MASTER* | TX/RX* | MODE |
|---|---|---|
| 0 | 0 | SINGLE IP, TRANSMITTER OFF, DEFAULT |
| 0 | 1 | MASTER IP, TRANSMITTER ON |
| 1 | 0 | SLAVE IP, RECEIVER ON |
| 1 | 1 | AMBIGUOUS |

**Figure 6 RS -485 Transmitter Modes**

CNTR: This is a master reset for the counters used to generate SCLK, LRCK and searializer address sequencing. CNTR is intend as a debug tool only. Writing a 1 to this bit resets all the counters. Caution, DAC data may be corrupted and the DACs will shut down. 5 Volt power consumption for the IP drops from 285 mA to about 65 mA when the counters and DACs are shut down. Writing a 0 to this bit allows the counters to run free. The power up default value of CNTR is 0.

## REF: Reference DAC Register

IP–DAC SU provides an independent DAC channel for level shifting the output voltage. The reference DAC register accepts the same 16 bit 2's complement data format as the main DAC channels. The reference DAC register is a double buffered latch that is automatically transcribed and serialized on every LRCK cycle. This process is transparent to software. The reference DAC is local to the IP. In multiple IP operation each IPs reference DAC is programmed independently.

The reference DAC value is converted to an analog voltage, inverted and summed with the main DAC channels. This feature allows programmable output leveling and offset adjustment on the fly. The reference signal inversion and summation technique cancels out inherent offsets in the CS4333s and automatically compensates for thermal drift. Because of the inversion; *Increasing the reference DAC value will shift the output levels down.* The Reference DAC register is a read/write flip–flop biased 16 bit latch. It powers up with a value of 0x0000. The default value of 0x0000 selects the ±5 Volt range. Setting the reference DAC value *down* to 0x8001 selects the 0 to +10 Volt range. Adjustments for offset variation can be added to these values.

| VALUE (HEX) | OUTPUT RANGE |
|---|---|
| 0X0000 | ±5 VOLTS, DEFAULT |
| 0X8001 | 0 TO +10 VOLTS |
| 0X7FFF | –10 TO 0 VOLTS |

**Figure 7 Reference DAC Values & Output Ranges**

## EED: Calibration EEPROM Data Register

This register is one bit wide and ported to the ATMEL AT93C56 Serial EEPROM. Please refer to the ATMEL Data sheet for details. The SK clock and wait states are automatically generated by the low level interface making data transfer transparent to software. Reading a 1 or 0 from this bit is equivalent to reading the EEPROM DO output pin. Writing a 1 or 0 to this bit is equivalent to writing to the EEPROM DI input pin.

For users who do not want to bother with reading the calibration EEPROM hard copy of the calibration data is supplied with each IP.

## EEC: Calibration EEPROM Control Register

This register is one bit wide and ported to the ATMEL AT93C56 Serial EEPROM. Please refer to the ATMEL Data sheet for details. Its contents are output directly to the EEPROM CS pin.

## TEST: Reserved Test Register

This register is used for factory testing of the RS-485 transceiver. The TR bit indicates the state of the R pin on the transceiver U8.1. TR is read/only. The TD drives the D pin on the transceiver directly U8.4. TD is read/write. The TD bit defaults to 0 for normal operation. This register is intended for factory use only and not supported.

## The Interrupt Group

The interrupt registers are organized in the standard SBS architecture for consistency and interrupt subroutine portability. Sample interrupt subroutine code examples are available in QuickPaks, contact SBS for details. Briefly, the structure consists of four overlaying registers that monitor and qualify interrupt sources. The interrupt sources are ranked in order of precedence from right to left. This allows for efficient bit shift operations in time critical interrupt subroutines. The fifth register is the user programmable interrupt vector.

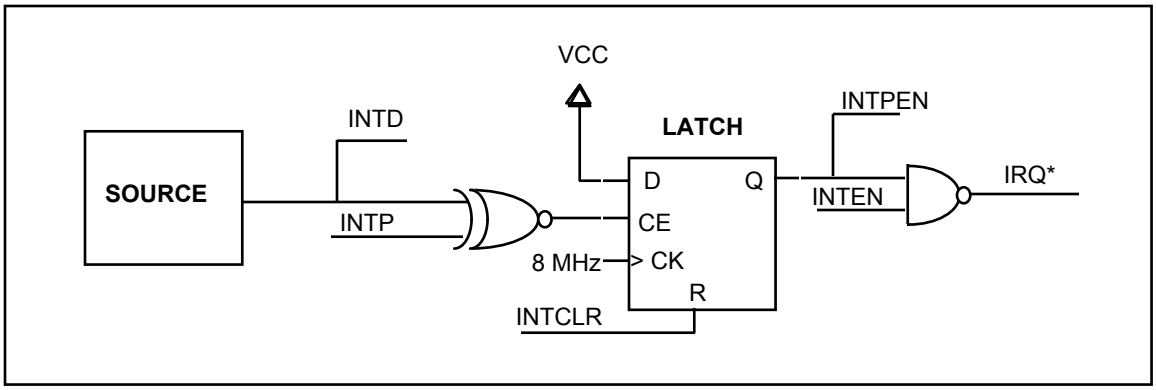


**Figure 8 Interrupt Architecture**

IP–DAC SUs interrupt circuitry is based setting and clearing an interrupt latch. When the seralizer state machine is ready for new data and updating the INTD bit sets the interrupt latch. Once the latch has been set it will remain set until cleared with the INTCLR function. INTPEN indicates the contents of the latch and INTEN masks the interrupt request to the host.

IP–DAC SU has only one interrupt source, the seralizer state machine status. This will generate an interrupt whenever IP–DAC SU is ready for new data and ready to start a new update cycle. Therefore, there is only one bit in each of the interrupt registers; D0.

### INTD/BUSY*: The Interrupt Data Register

This is a one bit read/only register that reads IP–DAC SU serializer state machine status. INTD/BUSY* is recommended for monitoring update cycle status. Reading a 1 from this bit indicates that IP–DAC SU is read for new DAC data and ready to start a new update cycle. Reading a 0 from this bit indicates that the serializer state machine is busy. Either an update cycle has been started or data serialization is in progress. *Do not attempt to access the DAC data registers if INTD/BUSY* is set to 0.* This feature allows for interrupt driven updating paced by the serializer state machine. Note; INTD indicates the state of the seralizer state machine not the interrupt latch. Use the IPEN function monitor the interrupt latch.

### INTP: The Interrupt Polarity Register

This register is not used on IP–DAC SU and hard wired to 1. The memory location is reserved.

## INTEN: Interrupt Enable Register

This is a one bit read/write register that enables the Interrupt Request from the IP to the host system. Reading a 1 from this register indicates that the interrupt is enabled. Reading a 0 from this register indicates that the interrupt is disabled. If the interrupt latch is set to 1, and the latch is set an Interrupt Request will be sent immediately to the host. If INTEN is set to 0, the interrupt request line is disabled regardless of the latch state. The power up default value of this register is 0 (i.e. Interrupts disabled).

## INTCLR: The Interrupt Clear Register

This is a one bit write/only register that clears the interrupt latch. This bit is connected to the reset of the interrupt latch. Writing a 1 to this bit clears the latch. Writing a 0 to this bit has no effect. All reads from this register will be 1. The INTCLR signal is a "one shot" event. That is the INTCLR signal is transitory; when a 1 is written the latch is cleared immediately. On the next clock cycle the latch is ready for a new interrupt. Returning the INTCLR bit to 0 after clearing is not necessary.

## IPEN: The Interrupt Pending Register

This is a one bit read/only register that indicates the contents of the interrupt latch. Reading a 1 indicates that the latch has been set. Reading a 0 from this register indicates that the latch is clear and no interrupt is pending. Note; This register shows the contents of the latch not the interrupt source status. Use INTD to monitor the interrupt source.

## VECTOR: The interrupt Vector Register

This is a read/write 8–bit register. It contains the interrupt vector asserted during interrupt servicing. All 8 bits D0..D7 are user definable. If the user program does not initialize the VECTOR register it defaults to 0x0F, the Motorola 68K family "uninitalized interrupt" value.

## The DAC Data Group

### CH1..CH16: DAC Data Registers

There are sixteen, 16–bit, read/write DAC data registers. After updating, their data is fed to the CS4333 DACs. The CS4333s use 16–bit 2's complement data format. Hence: 0x7FFF is maximum value, 0x8001 is minimum and 0x0000 is mid–scale.

The DAC data registers are biased on Xilinx SRAM cells. These registers are accessible as words or bytes in normal IP I/O space. Low level interface circuitry provides full 16 bit wide, no–wait state, access to the SRAM registers. Byte access is supported but word access is recommended. A two stage pipeline of SRAM is used. The first stage is user accessible. During serialization the data is transferred to a second SRAM FIFO stage. Serialized data in the FIFO is continuously fed to the CS4333 DACs. Xilinx does not guarantee the initial contents of the SRAM cells therefore the all DAC data registers must be initialized and an update must be completed after any system reset.

When the UPDATE bit in the control register is set the DAC data is serialized and loaded into the output FIFO. All 16 words are transferred from the input buffer to the output FIFOs every update cycle. While serialization is in progress accessing the data registers should not be attempted. The INTD register indicates when serialization is in progress. If the INTD bit is 0 serialization is in progress; do not access the data registers.

# Theory of Operation

## Serialization and Data Pipeline

The Crystal Semiconductor stereo CS4333 serial DACs use 16 bit 2's complement data format. The serial data stream is synchronized with three clocks, MCLK, SCLK and LRCK. On IP–DAC SU, MCLK is the 8 MHz IP clock. SCLK defines the Baud rate (bit rate) of the DACs. LRCK separates data words, stereo channels and defines the DAC "sampling" or refresh rate. Both the left and right channels of a DAC refreshed once every LRCK cycle. In audio systems LRCK would be a standard sampling frequency like; 32 kHz or 44.1 kHz. On IP–DAC SU; LRCK is derived from MCLK, the 8 MHz IP Clock:

$$T_{LRCK} = 256 \times T_{MCLK}.$$

$$T_{LRCK} = 256 \times 125\ nS = 32\ \mu S$$

$$F_{LRCK} = 1/T_{LRCK} = 31.25\ kHz$$

The CS4333 DACs use capacitive storage elements for data conversion. For DC operation the serial data must be continuously refreshed. Serialized DAC data is continuously cycled to the DACs by output FIFOs. FIFO Refreshing is automatically handled by the IP.

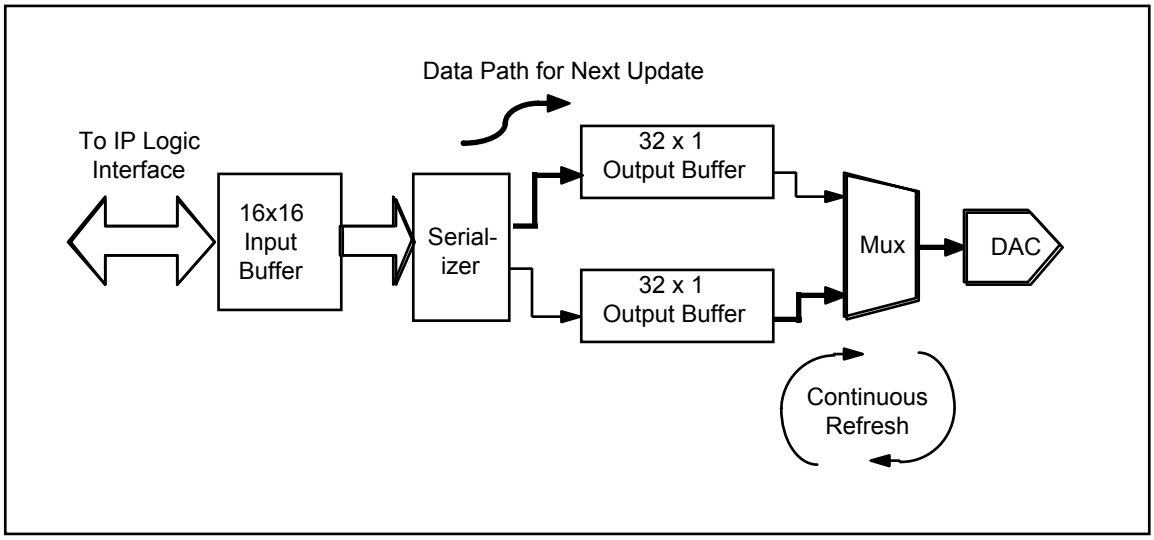


**Figure 9 Data Pipeline**

The DAC SU data pipeline employs a 16x16 bit SRAM based input buffer, a serializer and two parallel 32x8 bit SRAM Based FIFO outputs. Input data is loaded into the 16x16 input buffer at the head of the pipe. The input buffer can be accessed as normal IP I/O space. When updating is desired the UPDATE bit in the control register is set. IP–DAC SU will wait until the next LRCK cycle to begin conversion. The data in the input buffer is serialized and fed to one of the output FIFOs. After the FIFO has been filled, its contents are continuously fed to the DAC serial inputs. For subsequent data conversion new data is serialized and loaded into the other FIFO which is idle. After serialization the FIFOs are swapped. Thus the output FIFOs are "ping ponged" on every update cycle. FIFO swapping is transparent to the user and the DACs. While the serializer is loading FIFOs,

input SRAM addressing is under control of the serializer state machine. Accessing the DAC data registers during this period will have unknown results.

The distinction between Refresh rate and Update rate needs to be made. Refresh rate is the rate at which the FIFO data is fed to the DACs. The Refresh rate is the LRCK rate of 31.25 kHz. The CS4333 takes one LRCK cycle to change analog output level of both channels, regardless of magnitude.

Update rate is the throughput rate of the data seralizer and pipeline. The SRAM pipeline requires two LRCK cycles for serialization and the FIFO swap. And one more LRCK cycle is required to load new data into the DACs. Additionally there is a synchronization delay between setting the UPDATE bit and starting of conversion on an LRCK transition. In all, updating takes a little more than three LRCK cycles making the Update rate approximately 10 kHz.

## Reference DAC and Analog Output Stages

The CS4333s are designed for AC coupled applications. The raw output signal range is 4 Vpp, centered at 2.3 Vdc. IP–DAC SU's reference DAC signal is inverted and added to the raw DAC outputs to cancel out the 2.3 Vdc offset and other undesirable effects like thermal drift.

After leveling, the output signals are boosted with gain amplifiers. The gain amps are ratiometrically biased with resistor networks. This technique takes advantage of the relatively tight ratio tolerance between resisters in the same package and density of surface mount resister networks. For the standard product the signal gain is 3:1. This gain is slightly higher than necessary but guarantees a full 10 volt output swing. Other gain configurations can be special ordered.

# Applications Guide

## Power Up Default, Single IP Operation

IP–DAC SU is designed to run in single IP mode on power up or after reset. The REF Register defaults to ±5 Volt range. Interrupts are disabled. The control register defaults to 0 setting the IP into master mode with the RS–485 transceiver off. These defaults make IP–DAC SU ready for single IP operation right out of the box. Load 16–bit 2's complement data in the DAC Data registers, write a 1 to the UPDATE bit and IP–DAC SU will generate analog output. If a 0 to +10 Volt range is required set the REF register value to 0x8001. For 16, or less, analog channels the default configuration should serve most users needs.

## Multiple IP Operation

For systems requiring more than 16 simultaneously updated channels, multiple IP operation will be necessary. IP–DAC SUs are designed to be ganged by means of an RS–485 update sync signal. With the hardware; Connect all the SYNC A lines together and all SYNC B lines together. No null MODEM or "flipping" is required . In Software; Load DAC data into the desired data registers. Set the UPDATE bit in the Master IP and all IPs update simultaneously.

To configure an IP as a slave write 0x2 to its control register. This puts the IP in slave mode and turns the RS–485 receiver on. This configuration will remain in effect until altered by the program or the system is reset. In slave mode all registers are still accessible. For Slaves the same precautions about accessing DAC data during updates apply; Do not access DAC data during update cycles. The Master IPs INTD register should be used to determine update cycle status of the whole gang.

To configure an IP as a master , write 0x5 to its control register. This will turn on the RS–485 transmitter and start an update cycle. All slaves will sync up with the Master IP's clocks after the first gang update. To change the outputs of any of the IPs; alter the desired DAC data registers and write 0x5 to the Master IP control register again.

Every IP can be configured as either a master or a slave. However, *only one IP in a gang can be a master.* It is recommended that all Slave IPs be configured before turning on the master's RS–485 transmitter.

## Interrupt Driven Operation

For maximum throughput, IP–DAC SU can be interrupt driven by the seralizer state machine. In the interrupt service routine; first load DAC data, then start the new update cycle, while the cycle is in progress clear the interrupt latch, enable it and return to the main program. An interrupt request will be asserted when the update cycle has completed and the IP is ready for new data.

The interrupts work in either Single IP or multiple IP systems. Use the Master IP to drive multiple IP systems. All Slave IP must have their interrupts disabled. IP–DAC SU completes an update cycle in approximately 90µS, this speed and the large number of DAC data registers available may consume considerable processor time. System designers may want to use an external pacer clock for slower data rates.

## Optional ranges and External Power Supplies

IP–DAC SU can be special ordered with ranges outside the ±12 Volt rails. Two fused power lines EXTPWR+ and EXTPWR– are provided for external supplies. Power is routed through two traces E1 and E2 on the back side of the IP. These must be cut if external power is applied.

## Slew Rate and IP-DAC-SU

The CS4333s use builtin, low pass, DSP filters. The filter characteristics are optimized for AC audio performance with high dynamic range and low THD+N. This filter has some peculiar effects on the standard DC specifications engineers are accustom to seeing on DACs. Classically Slew Rate is used to determine output speed on Op-amps and DACs. Additionally the time of conversion to stable output (within 1 LSB) is of interest. These two parameters are obscured by the DSP filters.

The CS4333s perform all output voltage changes in roughly the same amount of time. This is a function of the sampling frequency (i.e. 31.25 kHz) NOT the magnitude of output voltage change. So whether the conversion is full scale, or just one bit, the conversion time is 1/31.25 kHz = 32 uS. To calculate a "Slew Rate" number in the specs we use:

$$\text{Max. Slew Rate} = (\text{Full Scale})/32\ \text{uS} = 10\ \text{Volts}/32\ \text{uS} = 310\ \text{mV/uS}$$

This is slow compared to general purpose Op-amps, however bear in mind the output ramp is constructed by the DSP filter. Additionally the DSP filter adds in damping oscillations on both the trailing edge, and the leading edge of the conversion ramp. In time domain the DAC output appears to be slow and unstable, but in frequency domain it minimizes out of band components which, very effectively, improved passband Dynamic Range and reduce THD+N. The damping oscillations are added to the conversion peroid. The envelopes of the damping oscillations are approximately 32 uS, on each end. If we add the leading oscillation, the conversion ramp and the trailing oscillation we get:

$$\text{Conversion Time to 1 LSB} = \text{Leading Osc.} + \text{Ramp} + \text{Trail Osc.} = 32 + 32 + 32 = 96\ \text{uS}.$$

From a time domain DC stand point this number is abysmal; however the effects on the small signal AC performance in frequency domain are optimal.

## Thermal Drift

The Crystal CS4333 DAC is pushing the envelope on density and is the smallest 16 bit DAC available. Unfrotunatly the small die size has resulted in a relatively large thermal drift coefficent. The final performance of 250 ppm/°C is considerably worse than expected when the design was conceived. For AC coupled applications, like audio, the drift is not a significant problem; however for the DC coupled IP-DAC-SU this problem is a bigger issue. For 16 bit resolution systems one LSB is 15 ppm. If the system can be thermally stabilized to 1°C, 250 ppm/°C is equivalent to 16 counts of drift which effectively reduces IP-DAC-SUs absolute accuracy to 12 bits. IP-DAC-SU should be considered for its high density and simultaneous update features. For applications where more stability is required; IP-16DAC, with 3 ppm/°C, drift is recommended.

# Calibration

Under ideal conditions, an ideal D/A would output the maximum voltage at its maximum D/A value, and the minimum output voltage at its minimum D/A value as shown in Figure 10. For example, IP–DAC SU uses a 16 bit signed integer D/A and a default voltage output of □5 volts. Thus, under ideal conditions, writing 32,768 to the D/A would output exactly +5 volts, –32,768 would output exactly –5 volts, and zero would output zero volts.

However, the actual voltage output differs from the desired voltage due to tolerances in the D/A, op amp and gain resistor components, . Software calibration compensates for these tolerances by adjusting the actual D/A count written to the DAC to achieve the desired output voltage.

Each channel is calibrated by sampling the voltage for a set of DAC inputs as shown in Figure 11. The voltage axis is converted to units of D/A counts by assuming a perfect D/A as shown in Figure 12. A least squares linear curve fit is applied to the data, and actual gain and offset of the D/A channel are computed as shown in Figure 13. The actual gain and actual offset for each channel are stored as the gain correction and offset correction factors. The definition and computation of these correction factors are shown in Figures 14 and 14.

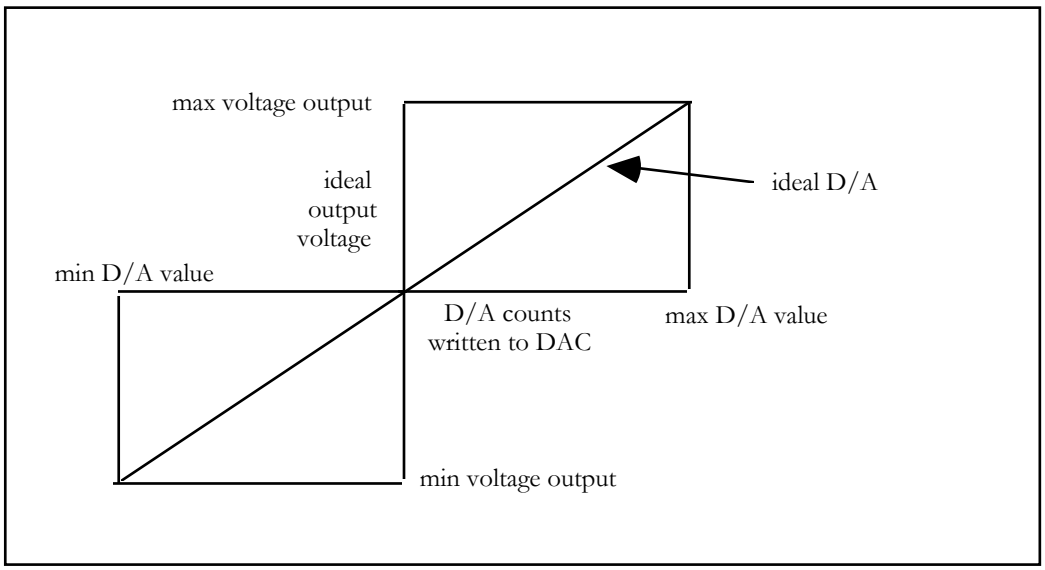


**Figure 10 Ideal D/A Output**

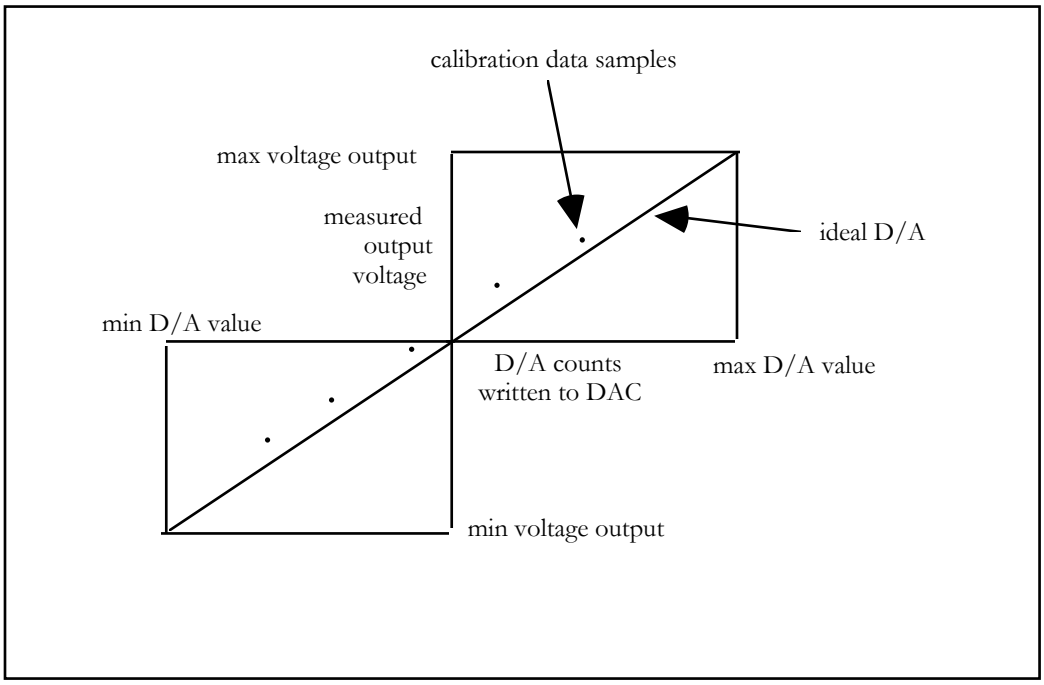


**Figure 11 Raw D/A Calibration Data**

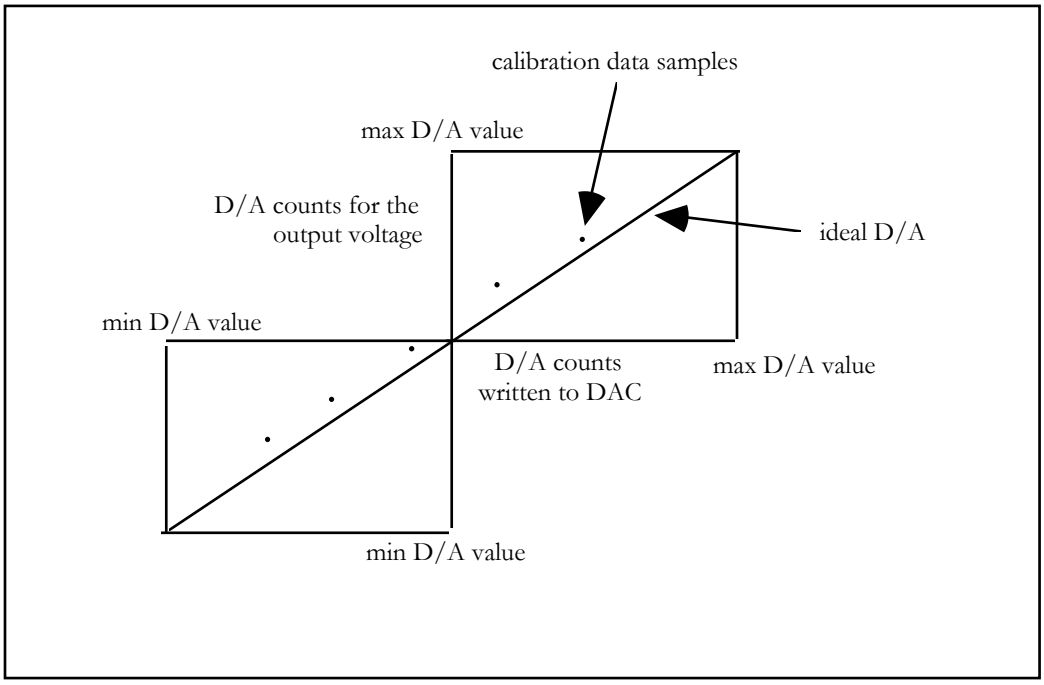


**Figure 12 D/A Calibration Data in Units of Counts**

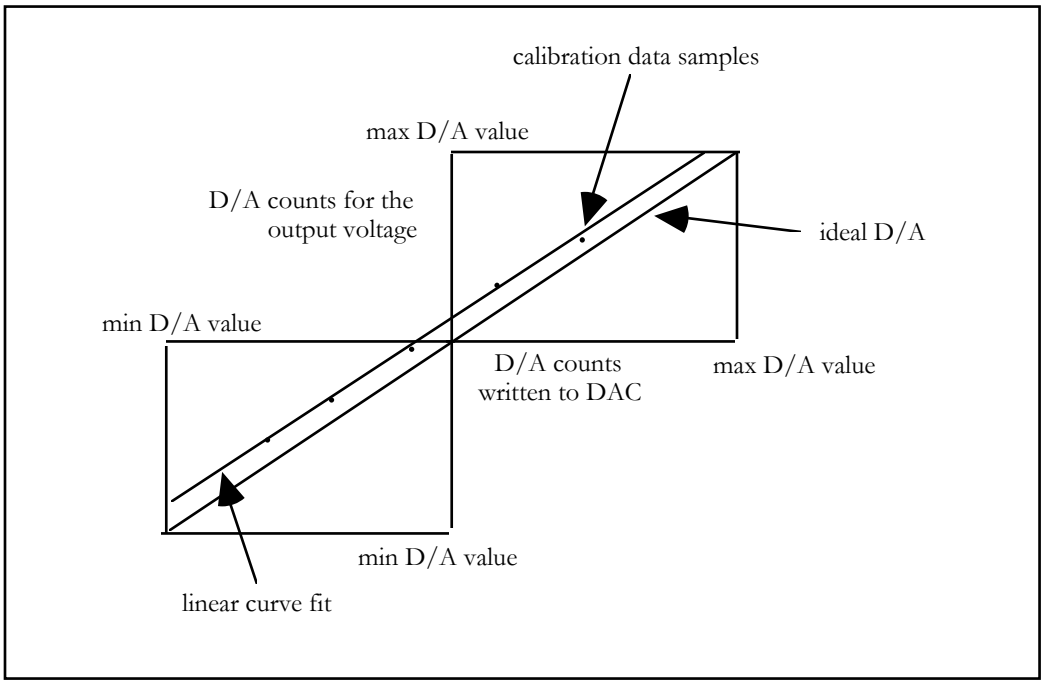


**Figure 13 D/A Calibration Curve**

## Offset Correction

Offset correction is defined as the voltage output of the channel converted to units of D/A counts, when the value zero is written to the D/A as shown in Figure 14. Offset correction is multiplied by a scaling factor to allow it to have a precision of 1/scale counts. The actual offset is recovered by the following equation:

$$actual\_offset = \frac{offset\_correction}{scale}$$

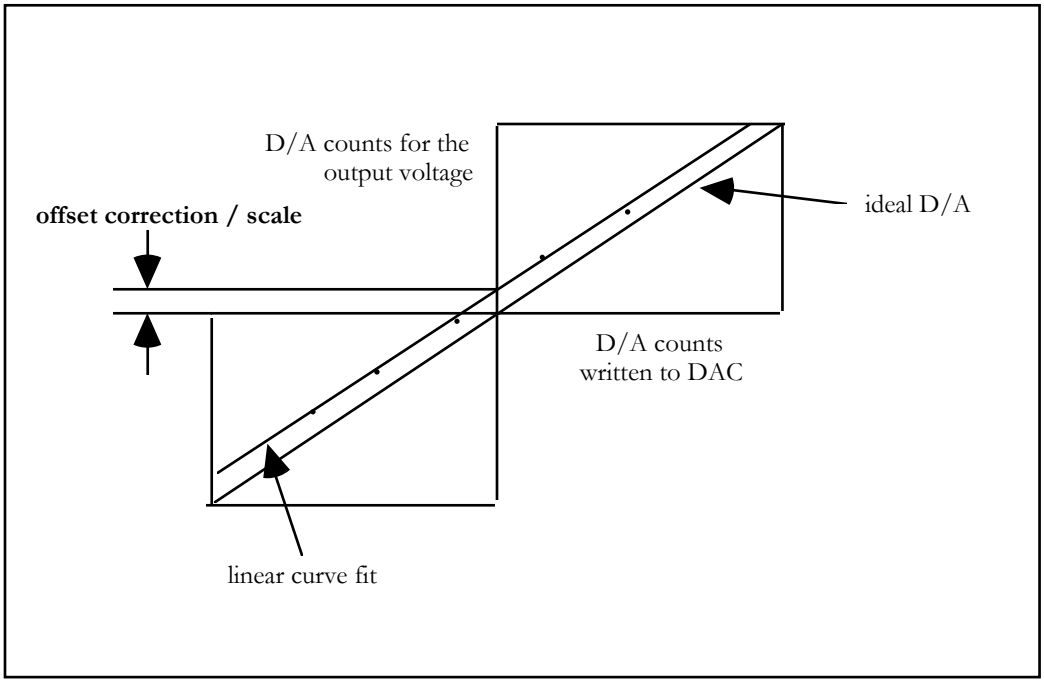


**Figure 14 Offset Correction**

## Gain Correction

Gain correction is defined in Figure 15. It is computed by the following equation and stored in the EEPROM

$$gain\_correction = scale * count\_span * (actual\_gain - 1)$$

where actual_gain is the slope of the linear curve fit of the calibration data and scale is a multiplication factor to allow gain_correction to have a precision of 1/scale counts. The actual gain is recovered from the gain correction factor by the following equation:

$$actual\_gain = \frac{scale * count\_span + gain\_correction}{scale * count\_span}$$

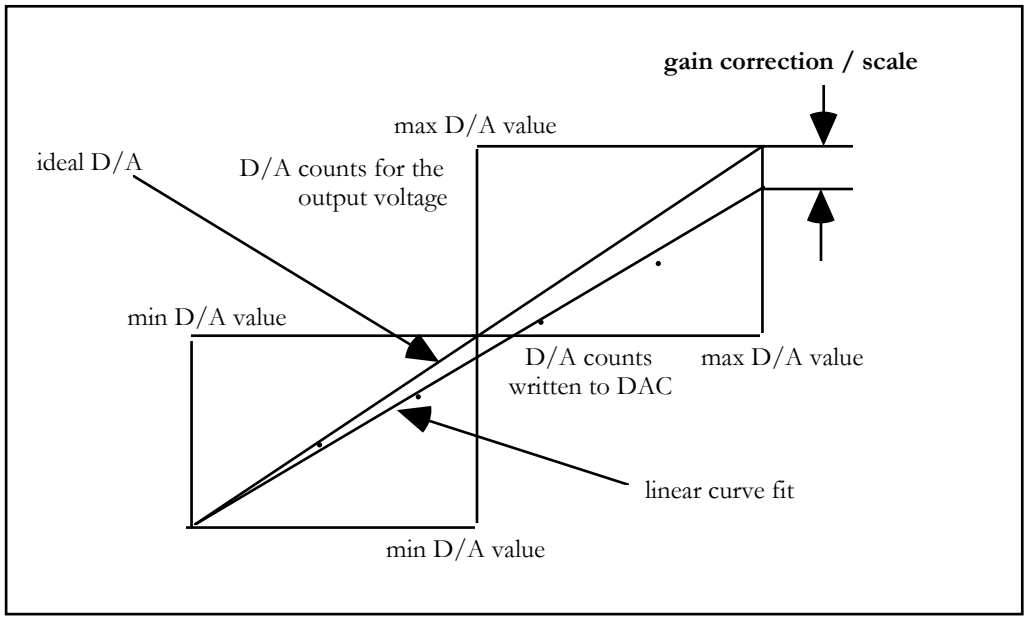


**Figure 15 Gain Correction**

## Using the Gain and Offset Correction Factors

### Convert the desired volts to ideal D/A counts

For IP–DAC SU, the value to write to the D/A to achieve the desired output voltage under ideal conditions is calculated by the following equation.

$$DAC = \frac{count\_span}{volt\_span} * (volts - volt\_offset)$$

where

| | |
|---|---|
| volts | Desired output voltage |
| volt_offset | Output voltage at a D/A input of zero counts. For a □5 volt configuration, volt_offset is 0. For a 0 to +10 volt configuration, volt_offset is 5 volts. |
| count_span | Difference between the maximum and minimum D/A values. The count_span for IP–DAC SU is 65536 (16 bit D/A). |
| volt_span | Difference between the maximum volt output and minimum volt output. For a □5 volt configuration, the volt_span is 10 volts. |
| DAC | Value to write to the D/A to achieve the desired output voltage under ideal conditions. |

For example, for a ±5 volt configuration, the DAC value would be computed as

$$DAC = \frac{65536}{10} * (volts - 0)$$

For a 0 to +10 volt configuration, the DAC value would be computed as

$$DAC = \frac{65536}{10} * (volts - 5)$$

For a –10 to 0 volt configuration, the DAC value would be computed as

$$DAC = \frac{65536}{10} * (volts + 5)$$

**Apply the correction formula**

$$correct\_DAC = \frac{\left[(scale * count\_span + gain\_correct) * DAC\right] + (count\_span * offset\_correct)}{scale * count\_span}$$

where

| | |
|---|---|
| correct_DAC | actual D/A input to achieve the desired output |
| gain_correct | gain correction factor stored in the ID PROM |
| offset_correct | offset correction factor stored in the ID PROM |
| scale | scaling factor applied to the gain and offset correction |

For IP–DAC SU, scale is 4096 and count_span is 65536. The correction formula is:

$$correct\_DAC = \frac{\left[(268,435,456 + gain\_correct) * DAC\right] + (65536 * offset\_correct)}{268,435,456}$$

## Gain and Offset Correction Data Format in EEPROM

A/D offset and gain correction data is stored in the serial EEPROM. The first 16 bit word is reserved to specify the format of the stored data. The second 16 bit word is reserved to specify the voltage range over which the calibration was performed. Data is stored high bit first. For example, when reading a 16 bit word from EEPROM, bit 15 is read first and bit 0 is read last. Correction data for IP–DAC SU are stored in format 12 as signed 32 bit values scaled by 4096. The data formats are specified as follows:

| Format code | Description |
|---|---|
| 0 | signed 8 bit value scaled by 4 |
| 1 | signed 16 bit value scaled by 256 |
| 2 | signed 16 bit value scaled by 8 |
| 3 | signed 16 bit value scaled by 16 |
| 4 | signed 16 bit value scaled by 32 |
| 5 | signed 16 bit value scaled by 64 |
| 6 | signed 16 bit value scaled by 128 |
| 7 | signed 16 bit value scaled by 256 |
| 8 | signed 32 bit value scaled by 256 |
| 9 | signed 32 bit value scaled by 512 |
| 10 | signed 32 bit value scaled by 1024 |
| 11 | signed 32 bit value scaled by 2048 |
| 12 | signed 32 bit value scaled by 4096 |
| 13 | signed 32 bit value scaled by 8192 |
| 14 | signed 32 bit value scaled by 16384 |
| 15 | signed 32 bit value scaled by 32768 |
| 16 | signed 32 bit value scaled by 65536 |

**Figure 16 Gain and Offset Correction Format**

| Offset | Channel | Description |
|---|---|---|
| 00 | – | data format, 16 bits, high bit first |
| 02 | – | voltage range, 16 bits, high bit first |
| 04 | - | Reference DAC value, 32 bits, high bit first |
| 08 | 1 | gain correction, 32 bits, high bit first |
| 0C | 1 | offset correction, 32 bit, high bit first |
| 10 | 2 | gain correction, 32 bits, high bit first |
| 14 | 2 | offset correction, 32 bit, high bit first |
| 18 | 3 | gain correction, 32 bits, high bit first |
| 1C | 3 | offset correction, 32 bit, high bit first |
| 20 | 4 | gain correction, 32 bits, high bit first |
| 24 | 4 | offset correction, 32 bit, high bit first |
| 28 | 5 | gain correction, 32 bits, high bit first |
| 2C | 5 | offset correction, 32 bit, high bit first |
| 30 | 6 | gain correction, 32 bits, high bit first |
| 34 | 6 | offset correction, 32 bit, high bit first |
| 38 | 7 | gain correction, 32 bits, high bit first |
| 3C | 7 | offset correction, 32 bit, high bit first |
| 40 | 8 | gain correction, 32 bits, high bit first |
| 44 | 8 | offset correction, 32 bit, high bit first |
| 48 | 9 | gain correction, 32 bits, high bit first |
| 4C | 9 | offset correction, 32 bit, high bit first |
| 50 | 10 | gain correction, 32 bits, high bit first |
| 54 | 10 | offset correction, 32 bit, high bit first |
| 58 | 11 | gain correction, 32 bits, high bit first |
| 5C | 11 | offset correction, 32 bit, high bit first |
| 60 | 12 | gain correction, 32 bits, high bit first |
| 64 | 12 | offset correction, 32 bit, high bit first |
| 68 | 13 | gain correction, 32 bits, high bit first |
| 6C | 13 | offset correction, 32 bit, high bit first |
| 70 | 14 | gain correction, 32 bits, high bit first |
| 74 | 14 | offset correction, 32 bit, high bit first |
| 78 | 15 | gain correction, 32 bits, high bit first |
| 7C | 15 | offset correction, 32 bit, high bit first |
| 80 | 16 | gain correction, 32 bits, high bit first |
| 84 | 16 | offset correction, 32 bit, high bit first |

**Figure 17 Gain and Offset Correction Data Format**

# I/O Pin Wiring

This section gives the pin assignments for IP–DAC-SU.

The pin numbers given in Figure 11 correspond to numbers on the 50–pin IndustryPack I/O connector, the wires on a 50–pin flat cable plugged into a standard IP carrier board and to the screw terminal numbers on the IP–Terminal block.

| I/O Pin | Signal | I/O Pin | Signal |
|---|---|---|---|
| 1 | AGND | 26 | AGND |
| 2 | CH 1 | 27 | SYNC A |
| 3 | AGND | 28 | AGND |
| 4 | CH 2 | 29 | SYNC B |
| 5 | AGND | 30 | AGND |
| 6 | CH 3 | 31 | Reserved |
| 7 | AGND | 32 | AGND |
| 8 | CH 4 | 33 | N/C |
| 9 | AGND | 34 | AGND |
| 10 | CH 5 | 35 | CH 9 |
| 11 | AGND | 36 | AGND |
| 12 | CH 6 | 37 | CH 10 |
| 13 | AGND | 38 | AGND |
| 14 | CH 7 | 39 | CH 11 |
| 15 | AGND | 40 | AGND |
| 16 | CH 8 | 41 | CH 12 |
| 17 | AGND | 42 | AGND |
| 18 | N/C | 43 | CH 13 |
| 19 | AGND | 44 | AGND |
| 20 | N/C | 45 | CH 14 |
| 21 | AGND | 46 | AGND |
| 22 | N/C | 47 | CH 15 |
| 23 | AGND | 48 | AGND |
| 24 | EXTPWR+ | 49 | CH 16 |
| 25 | EXTPWR– | 50 | AGND |

**Figure 18 I/O Pin Assignment**

# IndustryPack Logic Interface Pin Assignment

Figure below gives the pin assignments for the IndustryPack Logic Interface connector. Some pins that are irrelevant to IP–DAC-SU are not used.

| | | | | |
|---|---|---|---|---|
| GND | GND | 1 | 26 | |
| CLK | +5V | | 2 | 27 |
| Reset* | R/W* | 3 | 28 | |
| D0 | IDSel* | | 4 | 29 |
| D1 | DMAReq0* | 5 | 30 | |
| D2 | MemSel* | | 6 | 31 |
| D3 | DMAReq1* | 7 | 32 | |
| D4 | IntSel* | | 8 | 33 |
| D5 | DMAck0* | 9 | 34 | |
| D6 | IOSel* | | 10 | 35 |
| D7 | *reserved* | 11 | 36 | |
| D8 | A1 | | 12 | 37 |
| D9 | DMAEnd* | 13 | 38 | |
| D10 | A2 | | 14 | 39 |
| D11 | Error* | 15 | 40 | |
| D12 | A3 | | 16 | 41 |
| D13 | IntReq0* | 17 | 42 | |
| D14 | A4 | | 18 | 43 |
| D15 | IntReq1* | 19 | 44 | |
| BS0* | A5 | | 20 | 45 |
| BS1* | Strobe* | 21 | 46 | |
| –12 V | A6 | | 22 | 47 |
| +12 V | Ack* | 23 | 48 | |
| +5 V | *reserved* | | 24 | 49 |
| GND | GND | 25 | 50 | |

**Figure 19 IP Logic Interface Pin Assignment**

# ID PROM

Every IP contains an ID PROM,. The ID PROM aids in software auto configuration and configuration management. The user's software, or a supplied driver, may verify that the device it expects is actually installed at the location it expects, and is nominally functional. The ID PROM contains the manufacturing revision level of the IP. If a driver requires that a particular revision be present, it may check for it directly.

Standard data in the ID PROM on the IP–DAC-SU is shown in Figure 13 below. The Driver ID field represents the number of channels available in Hex. i.e. 16CH = 0x10 and 8CH = 0x08. For more information on IP ID PROMs refer to the IndustryPack Logic Interface Specification, available from SBS.

The location of the ID PROM in the host's address space is dependent on which carrier board used. Normally for VMEbus carriers the ID PROM space is directly above the IP's I/O space, or at IP–base + $80. Macintosh drivers use the ID PROM automatically. RM1260 address may be derived from Figure 13 below by multiplying the addresses given by two, then subtracting one. RM1270 addresses may be derived by multiplying the addresses given by two, then adding one.

The ID PROM is implemented in the Xilinx FPGA.

| | | |
|---|---|---|
| 17 | CRC | (63) |
| 15 | No of bytes used | (0C) |
| 13 | Driver ID, high byte | (00) |
| 11 | Driver ID, low byte | (10) |
| 0F | Reserved | (00) |
| 0D | Revision | (C1) |
| 0B | Model No IP–Manual | (4F) |
| 09 | Manufacturer ID SBS | (F0) |
| 07 | ASCII “C” | (43) |
| 05 | ASCII “A” | (41) |
| 03 | ASCII “P” | (50) |
| 01 | ASCII “I” | (49) |

**Figure 20 ID PROM Data (hex)**

# Construction and Reliability

IndustryPacks were conceived and engineered for rugged industrial environments. The IP–DAC-SU are constructed out of 0.062 inch thick FR4 material. The six copper layers consist of a ground plane, a power plane and four digital signal planes.

Through hole component mounting is used. IC sockets use gold plated screw–machine pins. High insertion and removal forces are required, which assists in keeping components in place. If the application requires unusually high reliability or is in an environment subject to high vibration, the user may solder the four corner pins of each socketed IC into the socket, using a grounded soldering iron. Shunts may be replaced with wire–wrap® wires if desired.

The IndustryPack connectors are keyed, shrouded and gold plated on both contacts and receptacles. They are rated at 1 Amp per pin, 200 insertion cycles minimum. These connectors make consistent, correct insertion easy and reliable.

The IP is secured to the carrier with four metric M2 stainless steel screws. The heads of these screws are countersunk into the IP. The four screws provide significant protection against shock, vibration, and incomplete insertion. For most applications they are not required.

The IndustryPack provides a low temperature coefficient of 0.89 W/°C for uniform heat. This is based on the temperature coefficient of the base FR4 material of .31 W/m–°C, and taking into account the thickness and area of the IP. This coefficient means that if 0.89 Watts is applied uniformly on the component side, that the temperature difference between the component and the solder side is one degree Celsius.

# Repair

Before returning a product for repair, verify as well as possible that the suspected unit is at fault. Then call the Customer Service Department for a RETURN MATERIAL AUTHORIZATION (RMA) number. Carefully package the unit, in the original shipping carton if this is available, and ship prepaid and insured with the RMA number clearly written on the outside of the package. Include a return address and the telephone number of a technical contact. For out–of–warranty repairs, a purchase order for repair charges must accompany the return. SBS will not be responsible for damages due to improper packaging of returned items. For service on SBS products not purchased directly from SBS contact your reseller. Products returned to SBS for repair by other than the original customer will be treated as out–of–warranty.

# Specifications

## IP-DAC-SU Specifications

| | |
|---|---|
| IP Type | 16 Channel, Type II single high, double sided<br>8 Channel, Type I, single high, single sided |
| Logic Interface | IndustryPack logic Interface, 0.7 compatible |
| Number of Channels | 16 |
| Resolution | 16–bit |
| Data Format | 16–bit 2's Complement |
| Ranges | 0 to +10V, ±5 Volts and 0 to –10 V standard,<br>Other ranges available as Customer Specials |
| Update Rate | 10 kHz x 16 Channels maximum throughput |
| Refresh Rate | 32.125 kHz |
| Accuracy | Absolute DC or "repeatability"<br>16 Counts typical @ 25 C°<br>80 Counts tested max.<br>Relative AC or "linearity"<br>1 Count (LSB) typical<br>2.5 Counts tested max. |
| "Slew Rate" | All conversions take 32 uS.<br>10 Volts/32 uS = 312 mV/S |
| Settling Time | 600 uS digital latency<br>200 uS analog swing with DSP Damping<br>800 uS single conversion<br>100 uS /pipelined conversion |
| Power Consumption | +5 Volts, 285 mA, typical<br>+12 Volts, 20 mA, typical<br>–12 Volts, 20 mA, typical |
| Temperature Coefficient | 0.89 W/°C for uniform heat,<br>component side to solder side |
| Dimensions | 1.800 by 3.900 by 0.340 inches maximum |
| Environmental | Operating temperature: 0 to 70°C |
| Humidity: | 5 to 95% non–condensing |
| Storage: | –10 to +85°C |

## DAC Specifications

| | |
|---|---|
| Device | Crystal Semiconductor CS4333 |
| Resolution | 16–bit |
| Type | delta sigma, 1–bit D/A conversion |
| Dynamic Range | 94 dB |
| THD | 0.003% |
| Drift | 250 PPM/°C |
| Power Consumption | 10 mA/Channel |

## Amplifier Specifications

| | |
|---|---|
| Device | Linear Technology LT1114 |
| Offset | 75 µV max |
| Drift | 0.5 µV/°C |
| Power Consumption | 900 µA/Channel |