CHAPTER 1
INTRODUCTION

The PCI bus 14 bit data acquisition card is a 32 bits PCI bus adapter with Plug and Play (PnP) features, it is a programmable I/O interface for Pentium or compatible computers. The PnP features let hardware configuration for IRQ and I/O address is detected by BIOS automatically, you don't need set switch and jumper.

The PCI bus 14 bit data acquisition card is a high precision data conversion/acquisition system. It contains 16 analog to digital channels with unipolar or bipolar input, 2 digital to analog channels with unipolar or bipolar output and 1 digital I/O channel with 16 bit line. The on board 8254(71054) chip provides programmable interval timer/counter functions to trig A/D conversion. The PCI bus 14 bit data acquisition card also provides interrupt driven for convention A/D input.

The features of the PCI bus 14 bit data acquisition adapter are:

- 32 bits PCI bus with Plug and Play (PnP) features.
- Programmable I/O control functions.
- Provides 16 A/D channels and the resolution is 14 bits.
- Provides 2 D/A channels and the resolution is 14 bits.
- D/A voltage range from 0 to 10V or –10V to 10V selectable.
- Different on board A/D chips selectable. The input voltage range from 0 to 2.5V or 0 to 5V for unipolar, and from –2.5V to 2.5V or –5V to 5V or –10V to 10V for bipolar.
- Provides 1 digital input/digital output channels and the resolution is 16 bits.
- Provides three 16 bits counter to trig A/D conversion.

- Provides software, external hardware signal or internal counter to trig A/D conversion.
- By using sampling and hold to get A/D signals.
- Interrupt or polling driven selectable.
- Gain control factor selectable from 1 to 8.

CHAPTER 2

UNPACKING INFORMATION

Check that your PCI bus 14 bit data acquisition package includes the following items:

- PCI bus 14 bit data acquisition adapter.
- User manual.
- Software utilities.
- Warranty form.

CHAPTER 3

SYSTEM REQUIREMENTS

Before installing your PCI bus 14 bit data acquisition adapter, make sure that:

- The host computer is an Pentium compatibles.
- The seven jumpers' blocks are correctly configured to coincide with the operating system you are using.

CHAPTER 4

HARDWARE INSTALLATION

Your PCI bus 14 bit data acquisition adapter is designed to be inserted in any available slot in your Pentium or compatibles. In order to gain access to the expansion slots, follow the steps listed below:

1. Turn off all power to your computer and all peripheral devices before installing your 14 bit data acquisition adapter.
2. Remove the cover of the computer.
3. Insert the pre-configured 14 bit data acquisition adapter into any available slot. Make sure the adapter is firmly seated in the chosen slot.
4. Replace the cover of the computer.

Note:

1. You must adjust the A/D full scalar reference voltage by screwing the VR resistor. (see VR Full Scalar Adjustment).
2. You must setup everything including the connection of the signal input/output into the DB25 and J2 connectors before turning on the PC power, otherwise it may damage the card.

CHAPTER 5

HARDWARE CONFIGURATION

5.1 Introduction

The seven jumper blocks on the PCI bus 14 bit data acquisition adapter must be configured correctly in accordance with the operating system you are using.

JP1 (Jumper 1)
Determines AD526 (gain control factor) is used.

JP2 (Jumper 2)
Adjust A/D converter input offset.

JP3 (Jumper 3)
Determine A/D voltage range.

JP4 (Jumper 4)
Determines D/A voltage range for channel 1.

JP5 (Jumper 5)
Determines D/A voltage range for channel 2.

JP6 (Jumper 6)
Determines unipolar or bipolar for D/A channel 1.

JP7 (Jumper 7)
Determines unipolar or bipolar for D/A channel 2.

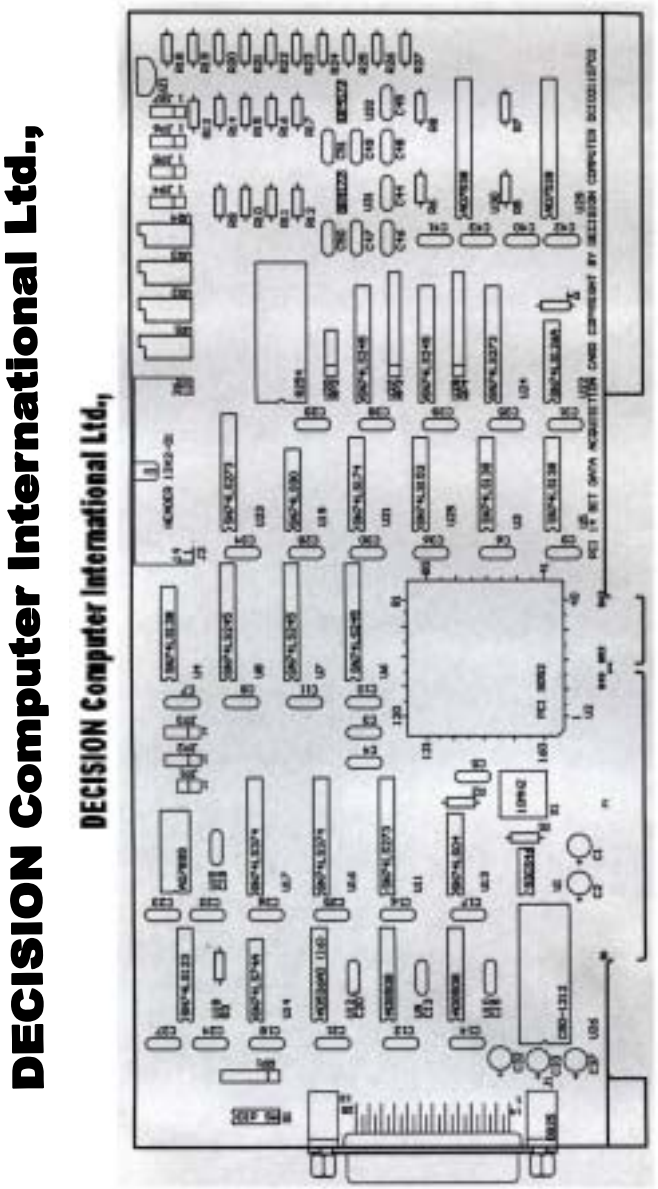
SW1 (Switch 1)
Identifies card number.

5.2 Configuration for Jumper

It is important to refer to the user manual to determine the correct configuration. Please contact your supplier if you have any difficulties with configuration.

☞ Please refer to the following settings for each jumper block.

Operations Manual PCI 14 bit data acquisition card



1. I/O Port Address

The plug and play features set I/O port address automatically, please refer the **device manager** of **control pane**l to get base port address of this adapter.

2. Voltage Range of D/A Channel

JP4 is used to select a range of output voltage for D/A channel 1, and JP5 is used to select a range of output voltage for D/A channel 2. JP6 is used to select unipolar or bipolar of D/A channel 1, and JP7 is used to select unipolar or bipolar of D/A channel 2.

JP4,JP5

JP6, JP7

JP4,5	JP6,7	Voltage Range
Short 2,3	Short 2,3	-10V to 10V
Short 1,2	Short 1,2	0V to 10V

3. Voltage Range of A/D Channel

JP3 is used to select input voltage range for A/D channel. There are several kinds of A/D chip can be plug in adapter. For different A/D chip, the voltage range and unipolar/bipolar setting are difference.

JP3

(1) For AD7899AR-1 chip
Voltage range : -5V to +5V or –10V to +10V +/-2LSB
Temperature : -40 to 85C

JP3	Voltage Range
Short 1,2	-10V to 10V
Short 2,3	-5V to 5V

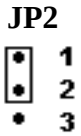
(2) For AD7899AR-2 chip
Voltage range : 0V to +2.5V or 0V to +5V +/-2LSB
Temperature : -40 to 85C

JP3	Voltage Range
Short 1,2	0V to 5V
Short 2,3	0V to 2.5V

(3) For AD7899AR-3 chip
Voltage range : -2.5V to +2.5V +/-2LSB
Temperature : -40 to 85C

JP3	Voltage Range
Short 1,2	-2.5V to 2.5V

4. Input Mode for AD7899



The JP2 is used to adjust input offset of AD7899. Setting the jumper to pin 2 and pin 3 will ground the input value (0V), so the data that will be converted should also be 0.

Shorting jumper to pin 1 and pin 2 will make use of the channel inputs to be read and converted. It is the normal input mode.

Jumper	Input Mode
short 2, 3	Ground
short 1, 2	Normal Input

5. Programmable Gain Control Factor (AD526)

The JP1 is used to select programmable gain control factor, when not short the jumper, it enable AD526, otherwise short the pin means no AD526 work (can not control gain control factor).

Jumper	AD526
short	no AD526
open	AD526

6. Card Identifier

The switch is used to identify card number, default setting is card 1, and there are two methods to set the card number:

a. PnP mode

Just plug in PCCOM PCI bus 14 bit data acquisition adapter into PCI slot, the PCI BIOS will allocate I/O address to each adapter automatically and assign card number start from 0 to each adapter. However, if you plug in more than one adapter, please set whole adapters to card number 1 (default setting), then use software tools to distinguish port id.

b. manual mode

Set card number by card identifier switch, the PCI BIOS will assign pre-allocated I/O address to each adapter. Please set different card number to each adapter (do not duplicate card number setting).

1	2	3	4	Card Number
ON	OFF	OFF	OFF	1
OFF	ON	OFF	OFF	2
ON	ON	OFF	OFF	3
OFF	OFF	ON	OFF	4

☞ The card number starts from 0 to 15.

5.3 I/O Address Specification

The I/O address specification are shown in the following:

For READ input
port + 0: input 14 bit A/D data.
port + 2: clear the interrupt signal that generated from the adapter.
port + 6: 16 bit digital signal input.
port + 8: counter 0 I/O buffer (8254 IC).
port + 9: counter 1 I/O buffer (8254 IC).
port + A: counter 2 I/O buffer (8254 IC).
port + B: counter control register (8254 IC).

For WRITE output
port + 0: select A/D channel number, enable/disable the selected channel, select IRQ and select control method.
port + 2: 14 bit D/A channel 1.
port + 4: 14 bit D/A channel 2.
port + 6: 16 bit digital output.
port + 8: counter 0 I/O buffer (8254 IC).
port + 9: counter 1 I/O buffer (8254 IC).
port + A: counter 2 I/O buffer (8254 IC).
port + B: counter control register (8254 IC).

5.4 VR Full Scalar Adjustment

VR Number	Function
VR1	Offset voltage for D/A channel 1
VR2	Offset voltage for D/A channel 2
VR3	D/A channel 1
VR4	D/A channel 2

Before adjust your D/A voltage offset, please run PCI14ADV00.EXE program under MS/DOS, then select calibration function as follows.

VR1 is used to adjust the D/A channel 1 offset voltage, and VR2 is used to adjust the D/A channel 2 offset voltage. In the following, we describe offset adjust.

To adjust D/A channel 1 offset voltage, please follow the step shown in the below:

Step 1: for min voltage, short pin 2 and pin 3 of JP4, and pin 2 and pin 3 of JP6, then adjust VR3 to let J1-15 output voltage to –10V. If the error rate more than 1.2mV, please enter step 2 to adjust VR1 for offset voltage.
Step 2: for middle voltage, short pin 2 and pin 3 of JP4, and pin 2 and pin 3 of JP6, then adjust VR1 to let J1-15 output voltage to 1.2mV.
Step 3: for max voltage, short pin 2 and pin 3 of JP4, and pin 2 and pin 3 of JP6, then adjust VR3 to let J1-15 output voltage to 10V. If the error rate more than 1.2mV, please enter step 2 to adjust VR1 for offset voltage.

To adjust D/A channel 2 offset voltage, please follow the step shown in the below:

Step 1: for min voltage, short pin 2 and pin 3 of JP5, and pin 2 and pin 3 of JP7, then adjust VR4 to let J1-3 output voltage to –10V. If the error rate more than 1.2mV, please enter step 2 to adjust VR2 for offset voltage.
Step 2: for middle voltage, short pin 2 and pin 3 of JP5, and pin 2 and pin 3 of JP7, then adjust VR2 to let J1-3 output voltage to 1.2mV.

Step 3: for max voltage, short pin 2 and pin 3 of JP5, and pin 2 and pin 3 of JP7, then adjust VR4 to let J1-15 output voltage to 10V. If the error rate more than 1.2mV, please enter step 2 to adjust VR2 for offset voltage.

5.5 Diagnostic Test

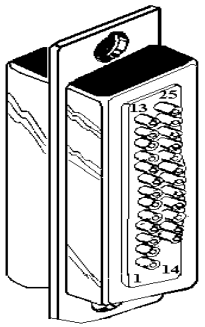
The PCI14ADV00.EXE also provides some diagnostic test functions:

1. A/D and D/A loopback test
2. D/A output test
3. A/D input test
4. DIO loopback test
5. DIO output test
6. DIO input test
7. D/A calibration

5.6 Pin Assignments

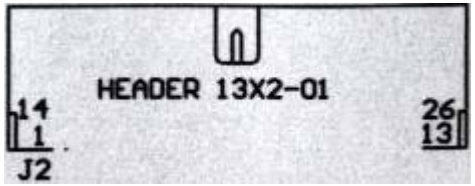
1. J1

Pin	Function	Pin	Function
1	+ 12 V	14	-12 V
2	/EXTRG	15	D/A channel 1
3	D/A channel 2	16	CH15
4	CH14	17	CH13
5	CH12	18	CH11
6	CH10	19	CH9
7	CH8	20	CH7
8	CH6	21	CH5
9	CH4	22	CH3
10	CH2	23	CH1
11	CH0	24	GND
12	GND	25	
13	+5 V		



2. J2

Pin	Function	Pin	Function
1		14	GND
2	+12V	15	GND
3	D8	16	D9
4	D10	17	D11
5	D12	18	D13
6	D14	19	D15
7	D0	20	D1
8	D2	21	D3
9	D4	22	D5
10	D6	23	D7
11		24	
12	+5V	25	GND
13	-12V	26	GND



CHAPTER 6

I/O PORT CONTROL

6.1 Input Port

1. port + 0

MSB															LSB
0	0	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Input A/D data, only D0 to D13 are useable, the highest two bits are 0.

2. port + 2

When read this port means clear the interrupt signal that generated from the adapter.

3. port + 6

MSB															LSB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Digital input channel. Before read this port, you must write FF value to this port.

4. port + 8

Counter 0 I/O buffer (8254 IC).

5. port + 9

Counter 1 I/O buffer (8254 IC).

6. port + A

Counter 2 I/O buffer (8254 IC).

7. port + B

Counter control register (8254 IC).

6.2 Output Port+0

The output port address+0 is used to control A/D conversion. The low 8 bit control format is :

MSB							LSB
x	R2	R1	R0	C3	C2	C1	C0

1. Select A/D channel number

The C0 to C3 are used to select input channel number.

C3	C2	C1	C0	Input Channel
0	0	0	0	CH0
0	0	0	1	CH1
0	0	1	0	CH2
0	0	1	1	CH3
0	1	0	0	CH4
0	1	0	1	CH5
0	1	1	0	CH6
0	1	1	1	CH7
1	0	0	0	CH8
1	0	0	1	CH9
1	0	1	0	CH10
1	0	1	1	CH11
1	1	0	0	CH12
1	1	0	1	CH13
1	1	1	0	CH14
1	1	1	1	CH15

2. Select gain control factor

The R0 to R2 are used to select gain control factor, if JP1 is short, the R0 to R2 are disable.

R2	R1	R0	Gain Control Factor
0	0	0	*1
0	0	1	*2
0	1	0	*3
0	1	1	*4
1	0	0	*5
1	0	1	*6
1	1	0	*7
1	1	1	*8

The gain control factor is used to scale your input voltage. For example, if you select unipolar and its voltage range from 0 to 20V, and the gain control factor is *8, then your input voltage range is from 0 to 2.5V, because whole the input voltage was scale 8 times.

3. Sampling and hold

The “x” bit is used to control sampling and hold, when write 0 to this bit, it latch the input voltage to let A/D converter get the input voltage, otherwise when write 1 to this bit, it enter sampling mode. Normally, this bit is 1.

x bit value	Action
0	Hold
1	Sampling

The high 8 bit control format of port address+0 is:

MSB							LSB
X	X	X	TRI	INT	ENX	SE1	SE0

4. Select trig method

The SE0 and SE1 are used to select trig method for A/D converter, user can select software trig, external hardware trig or trig by 8254.

SE1	SE0	Selection
0	0	Software trigger by TRI bit
0	1	Software trigger by TRI bit
1	0	External trig from J1-2
1	1	Trig by 8254

5. Enable/disable external hardware trig

ENX is used to enable/disable external hardware trig, when this bit is set to 0, it means disable external trig, otherwise this bit is set to 1 means enable external trig. When user select external hardware trig, he must connect external signal to J1 pin 2. When 8254 trig is selected, this bit must be set to 1.

ENX	Enable/Disable
0	Disable external trig
1	Enable external trig

6. Enable/disable interrupt

INT is used to enable/disable interrupt. If user enable interrupt, then after A/D conversion is finish, the hardware will generate interrupt. To set this bit to 0 means disable interrupt, otherwise set this bit to 1 to enable interrupt.

INT	Enable/Disable
0	Disable interrupt
1	Enable interrupt

7. Start software trig

TRI is used to start software trig. Normally, this bit is 1, when user start software trig, he must set this bit to 0, to let A/D converter start to convert. This software trig is enable while the signal from 1 to 0.

TRI	Enable/Disable
1	Normal
0	Start software trig

8. Start 8254 trig

When user select trig by 8254 (SE1=1 and SE0 =1), he must enable the gate by set ENX bit. The clock rate of 8254 is 1M, it is connected to counter 0, then the output of counter 0 is connected to counter 1, so that user need divide the clock by counter 0 then divide it by counter 1. The divided clock rate of counter 1 is used to trig A/D conversion.

9. Get results from A/D channel

Please note that, for different chip and voltage range set, the digital representation of A/D results are different, it defined as follows. Be careful that the digital representations are also different for D/A channel. We will describe it at next section.

(1) For AD7899AR-1 chip

JP3	Voltage Range
Short 1,2	-10V to 10V
Short 2,3	-5V to 5V

(a) –10V to 10V

-10V

MSB LSB

0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

0V

MSB LSB

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10V

MSB LSB

0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

(2) –5V to 5V

-5V

MSB LSB

0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

0V

MSB LSB

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5V

MSB LSB

0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

(2) For AD7899AR-2 chip

JP3	Voltage Range
Short 1,2	0V to 5V
Short 2,3	0V to 2.5V

(a) 0V to 5V

0V

MSB															LSB
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5V

MSB															LSB
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(2) 0V to 2.5V

0V

MSB															LSB
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2.5V

MSB															LSB
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(3) For AD7899AR-3 chip

JP3	Voltage Range
Short 1,2	-2.5V to 2.5V

-2.5V

MSB															LSB
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

0V

MSB															LSB
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2.5V

MSB															LSB
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

6.3 Data Output Port

1. port + 2

MSB															LSB
0	0	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The 14 bit D/A output channel 1.

2. port + 4

MSB															LSB
0	0	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The 14 bit D/A output channel 2.

For different voltage range, the digital representations of D/A output channel 1 and channel 2 are shown in the following.

JP4,5	JP6,7	Voltage Range
Short 2,3	Short 2,3	-10V to 10V
Short 1,2	Short 1,2	0V to 10V

(1) –10V to 10V
-10V

MSB															LSB
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10V

MSB															LSB
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(2) 0V to 10V
0V

MSB															LSB
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10V

MSB															LSB
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

3. port + 6

MSB															LSB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Digital output channel.

4. port + 8
counter 0 I/O buffer (8254 IC).

5. port + 9
counter 1 I/O buffer (8254 IC).

6. port + A
counter 2 I/O buffer (8254 IC).

7. port + B
counter control register (8254 IC).

CHAPTER 7

PROGRAMMING EXAMPLES

To use A/D converter, user must following the steps:

Step 1: Select A/D channel number.
Step 2: Hold the input signal.
Step 3: Start A/D conversion.
Step 4: Get results.

In the following we will describe each step:

7.1 Select A/D Channel Number

Set C3 to C0 of port address+0 (low 8 bits) to select A/D channel number.

MSB							LSB
				C3	C2	C1	C0

C3	C2	C1	C0	Input Channel
0	0	0	0	CH0
0	0	0	1	CH1
0	0	1	0	CH2
0	0	1	1	CH3
0	1	0	0	CH4
0	1	0	1	CH5
0	1	1	0	CH6
0	1	1	1	CH7
1	0	0	0	CH8
1	0	0	1	CH9

1	0	1	0	CH10
1	0	1	1	CH11
1	1	0	0	CH12
1	1	0	1	CH13
1	1	1	0	CH14
1	1	1	1	CH15

7.2 Hold the Input Signal

Set x bit of port address+0 (low 8 bits) to 0 to latch input signal.

MSB							LSB
x							

7.3 Start A/D Conversion

Set SE0 and SE1 of port address+0 (high 8 bits) to select trig method for A/D converter.

MSB							LSB
X	X	X	TRI	INT	ENX	SE1	SE0

SE1	SE0	Selection
0	0	Software trigger by TRI bit
0	1	Software trigger by TRI bit
1	0	External trig from J1-2
1	1	Trig by 8254

1. By using software trigger

1.1 Set or reset INT bit to enable/disable interrupt.

INT	Enable/Disable
0	Disable interrupt
1	Enable interrupt

1.2 Write 1 to TRI bit to start conversion.

2. By using 8254 trigger

2.1 Set or reset INT bit to enable/disable interrupt.
2.2 Set the ENX bit to 1.
2.3 Start 8254 counter.

The clock rate of 8254 is 1M, it is connected to counter 0, then the output of counter 0 is connected to counter 1, so that user need divide the clock by counter 0 then divide it by counter 1. The divided clock rate of counter 1 is used to trig A/D conversion

3. By using external trigger

3.1 Set or reset INT bit to enable/disable interrupt.
3.2 Set the ENX bit to 1.
3.3 Send "1" to "0" pulse from J1-02 pin.

7.4 Get Results

There are two methods to get the results of A/D conversion.

1. By using polling

After start conversion, wait at least 2.2us (dependent on CPU speed) to get the conversion result.

2. By using interrupt

2.1 Write an interrupt service routine.
2.2 After conversion, the adapter will generate an interrupt to start the interrupt service routine.
2.3 The interrupt service routine read 14 bits result from port address + 0.

MSB LSB

0	0	13	12	11	10	9	8	7	6	5	4	3	2	1	0

2.4 The interrupt service routine read port address + 2 to clear interrupt signal.

7.5 How to Use D/A Converter

1. Write 14 bits D/A to port address + 2 (D/A channel 1).

MSB LSB

0	0	13	12	11	10	9	8	7	6	5	4	3	2	1	0

2. Write 14 bits D/A to port address + 4 (D/A channel 2).

MSB LSB

0	0	13	12	11	10	9	8	7	6	5	4	3	2	1	0

3. After 20ns (dependent on CPU speed), the D/A converter will finish output procedure.

7.6 How to Use Digital Output Channel

Write data to port address + 6.

MSB															LSB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

7.7 How to Use Digital Input Channel

1. Write FF to port address + 6.
2. Read digital input from port address + 6.

CHAPTER 8

SOFTWARE DIAGNOSTIC

8.1 Programming Techniques Under MS/DOS

PCIADV.C

```
/*
   Test program for PCI-12/14bit A/D,D/A,D/I,DO
*/

#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

#include "pci.h"

#define ADC_CHANNEL     0x000F
#define ADC_SCALE    0x0070
#define ADC_SAMPLE 0x0080
#define ADC_TRIG_INT      0x0000
#define ADC_TRIG_EXT      0x0200
#define ADC_TRIG_8254     0x0300
#define ADC_EXT_GATE    0x0400
#define ADC_INTR       0x0800
#define ADC_TRIG       0x1000
#define ADC_HOLD     (~ADC_SAMPLE)

/* set default value for ioport, CNTR */
unsigned int ioport = 0;
unsigned int picbase = 0x20;
unsigned int picmask = 0;
unsigned long intcount = 0L;
```

```
unsigned int intdupcount = 0;
unsigned int irq = 0;
unsigned int adbit = 12;
unsigned int ssid = 0;

int old_adr = 0;


#define NUM_UNSIGNED     1
#define NUM_1_COMPLEMENT 2
#define NUM_2_COMPLEMENT 3
#define NUM_EXCEED       4

float num_volt = 10.0;
int num_system = NUM_2_COMPLEMENT;

static void interrupt (*oldisr) (void) = NULL;

static void interrupt isr (void)
{
//  disable ();
  intcount++;
  inport (ioport + 16);

  if (++intdupcount >= 50)
    outportb (picbase + 1, inportb (picbase + 1) | picmask);
  (*oldisr) ();
//  enable ();
  outportb (picbase, 0x20);
}
```



```
void Set_IRQ (void)
{
  int vect;

  if (0 == irq)
    return;
  picbase = (irq < 8)? 0x20 : 0xA0;
  picmask = 1 << (irq & 7);
  vect = (irq < 8)? 8 + irq : 0x68 + irq;
  oldisr = getvect (vect);
  setvect (vect, isr);
  outportb (picbase + 1, inportb (picbase + 1) & ~picmask);
}

void Reset_IRQ (void)
{
  int vect;

  if (0 == irq)
    return;
  outportb (picbase + 1, inportb (picbase + 1) | picmask);
  vect = (irq < 8)? 8 + irq : 0x68 + irq;
  setvect (vect, oldisr);
}

void Set_DA (int ch, int val)
{
  int OFS = 2;
  if (old_adr) OFS <<= 3;
  if (ch != 1 && ch != 2)
    return;
  outport (ioport + OFS * ch, val);
}
```

```
void Set_DO (unsigned x)
{
  int OFS = 6;

  if (old_adr) OFS <<= 3;
  outport (ioport + OFS, x);
}

unsigned int Get_DI (void)
{
  int OFS = 6;

  if (old_adr) OFS <<= 3;
  outport (ioport + OFS, 0xffff);
  return inport (ioport + OFS);
}

void Set_2VoltageRule (float range, int numsys)
{
  num_volt = range;
  num_system = numsys;
}

float Get_Voltage (int x)
{
  float voltage;
  int admax = (1 << adbit) - 1;

  voltage = ((float) num_volt / (admax + 1)) * x;

  return voltage;
}
```



```
void Test_AD_Loopback (void)
{
  unsigned int adc, nGain = 0, nChannel;
  int i, x, da, count, err, c;
  long int bias[16], mean[16], max[16];
  int admax = (1 << adbit) - 1;
  int OFS = 2;

  if (old_adr) OFS <<= 3;
  clrscr ();
  gotoxy (30, 2);
  cprintf ("<AD/DA loop-back test>");
  gotoxy (8, 4);
  cprintf ("Channel  Input  Error  MeanErr     Bias     MaxErr");
  for (i = 0; i < 16; i++) {
    bias[i] = mean[i] = max[i] = 0;
  }

  outport (ioport + OFS, 0);
  adc = ADC_TRIG_INT | ADC_INTR | ADC_SAMPLE;
  outport (ioport, adc);
  delay (1);
  adc &= ADC_HOLD;
  outport (ioport, adc);
  adc |= ADC_TRIG;
  outport (ioport, adc);
  adc &= ~ADC_TRIG;
  outport (ioport, adc);

  delay (100);
  count = -2;
```

```
 for (da = admax; da >= 0; da -= 5) {
    count++;
    outport (ioport + OFS, da);
    gotoxy (3, 22);
    cprintf ("D/A output: %6d", da);
    delay (10);
    for (nChannel = 0; nChannel < 16; nChannel++) {
      adc = nChannel | (nGain << 4) | ADC_TRIG_INT | ADC_INTR
| ADC_SAMPLE;
      outport (ioport, adc);
      delay (1);
      adc &= ADC_HOLD;
      outport (ioport, adc);
      adc |= ADC_TRIG;
      outport (ioport, adc);
      adc &= ~ADC_TRIG;
      outport (ioport, adc);
      x = inport (ioport);
      x &= admax;
      if (count <= 0)
        continue;
      bias[nChannel] += x - da;
      err = abs (x - da);
      if (max[nChannel] < err)
        max[nChannel] = err;
      mean[nChannel] += err;
      gotoxy (10, 5 + nChannel);
      cprintf ("%2d :%7d %5d     %7.4f    %7.4f  %7ld", nChannel + 1,
x, err,
      (float)mean[nChannel] / count,
      (float)bias[nChannel] / count,
      max[nChannel]);
    }
```



```
    if (kbhit ()) {
      c = getch ();
      if (c == 27)
        break;
      else if (c == ' ')
        getch ();
    }
  }
  getch ();
}

int Get_AD (int nChannel, int nGain)
{
  int admax = (1 << adbit) - 1;
  int signbit = 1 << (adbit - 1);
  unsigned int adc = 0;
  unsigned int x = 0;
  int OFS = 0;

  if (old_adr) OFS <<= 3;
  if (nChannel < 0 || nChannel > 15)
    return 0;
  if (nGain < 0 || nGain > 7)
    return 0;
  adc = nChannel | (nGain << 4) | ADC_TRIG_INT | ADC_INTR |
ADC_SAMPLE;
//  adc = nChannel | (nGain << 4) | ADC_TRIG_INT |
ADC_SAMPLE;
  outport (ioport + OFS, adc);
//  delay (5);
  delay (1);
  adc &= ADC_HOLD;
  outport (ioport + OFS, adc);
```

```
  adc |= ADC_TRIG;
  outport (ioport + OFS, adc);
  adc &= ~ADC_TRIG;
  outport (ioport + OFS, adc);
  x = inport (ioport + OFS);
  x &= admax;
  switch (num_system) {
    case NUM_UNSIGNED:
      break;
    case NUM_2_COMPLEMENT:
      if (x & signbit)
        x = x - (1 << adbit);
      break;
    case NUM_1_COMPLEMENT:
      if (x & signbit)
        x = signbit - x;
      break;
    case NUM_EXCEED:
        x = x - signbit;
      break;
  }
  return x;
}

void Show_AD_Value (int nChannels, int nGain)
{
  int x, n;
  float voltage;
//  int admax = (1 << adbit) - 1;
//  int c = 0;
//  int quit=0;

//  adc = ADC_TRIG_INT | ADC_INTR;
```



```
//  outport (ioport, adc);

  for (n = 0;;) {
    x = Get_AD (n, nGain);
    gotoxy (3, 2 + n);
    voltage = Get_Voltage (x);
    gotoxy (3, 2 + n);
    cprintf ("Channel#%2d :%7d      %8.4f V", n, x, voltage);
    if (n < nChannels - 1)
      n++;
    else
      n = 0;
    if (kbhit ())
      break;
  }
}

void AD_Calibration (void)
{
  int admax = (1 << adbit) - 1;
  int vrange = 10;
  int numsys = NUM_2_COMPLEMENT;

  if (adbit == 14) {
    if (ssid == PCI_SID_AD7899AR1) {
      numsys = NUM_2_COMPLEMENT;
      vrange = 20;
    } else if (ssid == PCI_SID_AD7899AR2) {
      numsys = NUM_UNSIGNED;
      vrange = 10;
    }
    Set_2VoltageRule (vrange, numsys);
  }
```

```
  clrscr ();
//  if (adbit == 12) {
    gotoxy (3, 19);
    cprintf ("JP2  JP3  JP4  JP5  (Open JP1 if AD526 is used)");
    gotoxy (3, 20);
    cprintf (" 23   23  12   12   Adjust VR3 and VR4 to %d\n", admax
/ 2);
    Show_AD_Value (16, 0);
    while (kbhit()) getch ();
    gotoxy (3, 21);
    cprintf (" 23   23  12    23   Adjust VR2 to 0\n");
    Show_AD_Value (16, 0);
    while (kbhit()) getch ();
    gotoxy (3, 22);
    cprintf (" 23  12   12   12    Adjust VR1 to %d\n", admax / 2);
    Show_AD_Value (16, 0);
    while (kbhit()) getch ();
    gotoxy (3, 23);
    cprintf ("12   12   12    23   Adjust VR1 to %d\n", admax / 2);
    Show_AD_Value (16, 0);
    while (kbhit()) getch ();
//  }
}

/*
void testadinput (int adbit)
{
  unsigned int adc, x, nGain = 0, nChannel;
  long loop;
  int c, i, err, duration = 8, step = 0;
  int admax = (1 << adbit) - 1;
  int da = admax / 2;
```



```
  int OFS = 64 + 12;

  if (old_adr) OFS <<= 3;

  clrscr ();
  gotoxy (3, 24);
  cprintf ("1 : scale   2, 3 : Trigger Source");

  outport (ioport + 64 + 12, 0x36); //Counter 0: Mode 3, LSB + MSB
  outport (ioport + 64, 100);
  outport (ioport + 64, 0);

  outportb (ioport + 64 + 12, 0x76); //Counter 1: Mode 3, LSB +
MSB
  outportb (ioport + 64 + 4, 100);
  outportb (ioport + 64 + 4, 0);

  inport (ioport + 16);// clear irq
  setirq ();

  for (loop = 0;;loop++) {
    outport (ioport + 16, da);
    adc = nChannel | (nGain << 4) | ADC_TRIG_8254 | ADC_INTR;
    //adc = nChannel | (nGain << 4) | ADC_TRIG_INT | ADC_INTR
| ADC_SAMPLE;
    outport (ioport, adc);
    for (nChannel = 0; nChannel < 16; nChannel ++) {
      intdupcount = 0;
      adc = nChannel | (nGain << 4) | ADC_TRIG_8254 | ADC_INTR;
      //adc = nChannel | (nGain << 4) | ADC_TRIG_INT |
ADC_INTR;
      outport (ioport, adc);
      //delay (duration);
```

```
    //adc &= ADC_HOLD;
    //outport (ioport, adc);
    //delay (1);
    //adc |= ADC_TRIG;
    //outport (ioport, adc);
    //adc &= ~ADC_TRIG;
    //outport (ioport, adc);
    for (i = 0; i < 5; i++) {
      if (intdupcount > 0)
        break;
      delay (1);
    }
    x = inport (ioport);
    err = abs (x - da);
    gotoxy (3, 5 + nChannel);
    cprintf ("Channel#%2d :%7d  %5d  %5d", nChannel, x, err,
intdupcount);
  }
  gotoxy (3, 22);
  cprintf ("Loop :%7ld   IRQ :%7ld   Scale = %2d   D/A : %4d",
loop, intcount, nGain + 1, da);
  while (kbhit () || step) {
    c = getch ();
    if (c == 27)
      break;
    else if (c == ' ')
      step = !step;
    else if (c == '+') {
      if (duration < 256)
        duration <<= 1;
    }
    else if (c == '-') {
      if (duration > 1)
```



```
        duration >>= 1;
    }
    else if (c == '1') {
      nGain++;
      if (nGain >= 8)
        nGain = 0;
    }
    else if (c == '3') {
      da += 100;
      if (da > admax)
        da = admax;
    }
    else if (c == '2') {
      da -= 100;
      if (da < 0)
        da = 0;
    }
    else if (!step)
      c = getch ();
  }
  if (c == 27)
    break;
 } // for (loop)
 if (kbhit ())
  getch ();
 resetirq ();
}
*/

void Test_AD (void)
{
 //long loop;
 int c, x;
```

```
 int numsys = NUM_2_COMPLEMENT;
 int vrange = 10;

 if (adbit == 14) {
  switch (ssid) {
   case PCI_SID_AD7899AR1:
    numsys = NUM_2_COMPLEMENT;
    vrange = 20;
    break;

   case PCI_SID_AD7899AR2:
    numsys = NUM_UNSIGNED;
    vrange = 10;
    break;
  }
  Set_2VoltageRule (vrange, numsys);
 }

 //loop = 0;
 x = 0;
 clrscr ();
 for (;;) {
  gotoxy (3, 23);
  cprintf ("Scale: %3d", 1 << x);
  Show_AD_Value (16, x);
  while (kbhit ()) {
   c = getch ();
   switch (c) {
    case '1':
     if (x > 0)
      x--;
     break;
    case '2':
```



```
     if (x < 4)
      x++;
     break;
   }
   if (c == 27)
    break;
  }
  if (c == 27)
   break;
 }
}

void Calibrate_DA (int nChannel)
{
 int x = 0;
 unsigned char c = 0;
 int admax = (1 << adbit) - 1;

 gotoxy (3, 24);
 cprintf ("1: Min   2: Middle    3: Max    ESC: Continue");

 for (;;) {
  Set_DA (nChannel, x);
  gotoxy (3, 18);
  cprintf ("D/A Channel#%d :%7d", nChannel, x);
  if (kbhit ()) {
   c = getch ();
   if (c == 27)
    break;
   else if (c == '1') {
    x = 0;
   }
   else if (c == '2') {
```

```
      x = admax / 2 + 1;
     }
     else if (c == '3') {
      x = admax;
     }
   }
  } // for
  if (kbhit ())
   getch ();
}

void DA_Calibration (void)
{
  clrscr ();

  if (adbit == 12) {
   gotoxy (3, 10);
   cprintf ("1: JP5->12, Adjust VR5, -10 to 10 V");
   Calibrate_DA (1);
   gotoxy (3, 11);
   cprintf ("2: JP5->34, Adjust VR5, 0 to 10 V");
   Calibrate_DA (1);
   gotoxy (3, 12);
   cprintf ("3: JP6->12, Adjust VR6, -10 to 10 V");
   Calibrate_DA (2);
   gotoxy (3, 13);
   cprintf ("4: JP6->34, Adjust VR6, 0 to 10 V");
   Calibrate_DA (2);
  }
  else {
   gotoxy (3, 10);
   cprintf ("Channel 1: JP4->12 for Unipolar or JP4-23 for Bipolar");
   gotoxy (3, 11);
```



```
   cprintf ("Adjust VR1 for offset and VR3 for range");
   gotoxy (3, 12);
   cprintf ("Output: 0 - 10V for Unipolar, -10V to 10V for Bipolar");
   Calibrate_DA (1);
   gotoxy (3, 14);
   cprintf ("Channel 2: JP5->12 for Unipolar or JP5-23 for Bipolar");
   gotoxy (3, 15);
   cprintf ("Adjust VR2 for offset and VR4 for range");
   gotoxy (3, 16);
   cprintf ("Output: 0 - 10V for Unipolar, -10V to 10V for Bipolar");
   Calibrate_DA (2);
  }
}

void Test_DA (void)
{
  int chwave[2] = {3, 3};
  char *desc[8] = {"min",
                   "max",
                   "square",
                   "triangle",
                   "rising triangle",
                   "falling triangle",
                   "bit increasing",
                   "DC"};

  int x[2] = {0, 0};
  int y[2] = {0, 0};
  int dflag[2] = {0, 0};
  unsigned long loop = 0L;
  int i, flag, quit;
  int step = 0;
  unsigned char c = 0;
```

```
  unsigned duration = 27;
  int admax = (1 << adbit) - 1;
  int actch = 0;
  int dc[2] = {0, 0};

  clrscr ();
  gotoxy (30, 2);
  cprintf ("< DA test >");
  gotoxy (3, 20);
  cprintf ("Press 1 or 2 to change wave type.");
  gotoxy (3, 21);
  cprintf ("Press + - to change frequency.");
  gotoxy (3, 22);
  cprintf ("Press Up Down PgUp PgDn Home End to change DC
value.");
  for (loop = 0, quit = 0;!quit;loop++) {
    Set_DA (1, x[0]);
    Set_DA (2, x[1]);
    gotoxy (10, 10);
    cprintf ("Active Channel: %d              DC Voltage   DC Voltage
DC Voltage", actch + 1);
    gotoxy (10, 11);
    cprintf ("Channel   Wave form            single polar  bi-polar(+) bi-
polar(-)");
    gotoxy (10, 8);
    cprintf ("Loop:%9ld     D/A #1 :%7d     D/A #2 :%7d", loop, x[0],
x[1]);
    for (i = 0; i < 2; i++) {
      gotoxy (10, 12 + i);
      cprintf ("   %d      %-20s %8.3fV      %8.3fV   %8.3fV",
        i + 1, desc[chwave[i]], (10.0 * dc[i]) / (admax + 1),
        (20.0 * dc[i]) / (admax + 1) - 10.0,
        10.0 - (20.0 * dc[i]) / (admax + 1));
```



```
    switch (chwave[i]) {
      case 0:
        x[i] = 0;
        break;
      case 1:
        x[i] = admax;
        break;
      case 2:
        y[i] += duration;
        if (y[i] > 1000) {
          y[i] = 0;
          x[i] = (x[i])? 0 : admax;
        }
        break;
      case 3:
        if (dflag[i]) {
          x[i] += duration;
          if (x[i] > admax) {
            x[i] = admax;
            dflag[i] = 0;
          }
        } else {
          x[i] -= duration;
          if (x[i] < 0) {
            x[i] = 0;
            dflag[i] = 1;
          }
        }
        break;
      case 4:
        x[i] += duration;
        if (x[i] > admax)
          x[i] = 0;
```

```
      break;
    case 5:
      x[i] -= duration;
      if (x[i] < 0)
        x[i] = admax;
      break;
    case 6:
      x[i] <<= 1;
      if (x[i] > admax)
        x[i] = 1;
      break;
    case 7:
      x[i] = dc[i];
      break;
  } //switch
} //for

if (kbhit () || step) {
  c = getch ();
  if (c == 0 && kbhit()) {
    c = getch ();
    switch (c) {
      case 73:
       dc[actch] += 1;
       if (dc[actch] > admax)
         dc[actch] = admax;
       break;
      case 72:
       dc[actch] += 32;
       if (dc[actch] > admax)
         dc[actch] = admax;
       break;
      case 71:
```



```
       dc[actch] += 1024;
       if (dc[actch] > admax)
         dc[actch] = admax;
       break;
      case 81:
       dc[actch] -= 1;
       if (dc[actch] < 0)
         dc[actch] = 0;
       break;
      case 80:
       dc[actch] -= 32;
       if (dc[actch] < 0)
         dc[actch] = 0;
       break;
      case 79:
       dc[actch] -= 1024;
       if (dc[actch] < 0)
         dc[actch] = 0;
       break;
      case 75:
       actch = 0;
       break;
      case 77:
       actch = 1;
       break;
    } //switch
  }
  else {
    switch (c) {
      case 27:
        quit = 1;
        break;
      case ' ':
```

```
      step = !step;
      break;
    case '+':
      if (duration < 200)
        duration *= 3;
      break;
    case '-':
      if (duration >= 3)
        duration /= 3;
      break;
    case '1':
      if (actch != 0) {
        actch = 0;
      } else {
        chwave[0]++;
        if (chwave[0] > 7)
          chwave[0] = 0;
      }
      break;
    case '2':
      if (actch != 1) {
        actch = 1;
      } else {
        chwave[1]++;
        if (chwave[1] > 7)
          chwave[1] = 0;
      }
      break;
    default:
      if (!step)
        getch ();
    } // switch
  } //else
```



```
    } //if kbhit
  } // for (loop)
}

void Test_DIO_Loopback (void)
{
  int loop, i, c;
  unsigned x, y;

  int OFS = 6;

  if (old_adr) OFS <<= 3;
  for (loop = 0; loop < 10; loop ++) {
    for (i = 0; i < 16; i++) {
      x = 1 << i;
      outport (ioport + OFS, x);
      delay (10);
      y = inport (ioport + OFS);
      gotoxy (3, 21);
      cprintf("Loop: %3d  Testing Pin:%2d  Output: %04X
Input:%04X",
              loop + 1, i, x, y);
      if (x != y) {
        cputs (" Error ");
        c = getch ();
        if (c == 27)
          break;
      }
      else {
        cputs ("  OK   ");
        //c = getch ();
      }
      if (kbhit()) {
```

```
      c = getch ();
      c = getch ();
    }
    if (c == 27)
      break;
  }
  if (c == 27)
    break;
 }
 gotoxy (3, 17);
 cputs("Press any key when ready");
 if (kbhit ())
  getch();
}

void dumpbits (unsigned long x, int n)
{
 int i;
 unsigned long b = 1L << (n - 1);

 for (i = 0; i < n; i++) {
  putch (((b >> i) & x)? '1' : '0');
 }
}

void Test_DO (void)
{
 unsigned x;
 unsigned long loop = 0L;
 int step = 0;
 unsigned char c = 0;
 unsigned duration = 64;
```



```
 for (loop = 0, x = 1;;loop++) {
  if (x == 0x8000)
   x = 1;
  else
   x <<= 1;
  Set_DO (x);
  gotoxy (3, 21);
  cprintf("Loop: %8ld : ", loop);
  dumpbits (x, 16);
  delay (duration);
  if (kbhit () || step) {
   c = getch ();
   if (c == 27)
    break;
   else if (c == ' ')
    step = !step;
   else if (c == '+') {
    if (duration > 1)
     duration >>= 1;
   }
   else if (c == '-') {
    if (duration < 512)
     duration <<= 1;
   }
   else if (!step)
    getch ();
  }
 } // for (i)
 if (kbhit ())
  getch ();
}

void Test_DI (void)
```

```
{
  unsigned x;
  unsigned long loop = 0L;
  unsigned duration = 64;
  int step = 0;
  unsigned char c;

  for (loop = 0;;loop++) {
    x = Get_DI ();
    gotoxy (3, 21);
    cprintf("Loop: %8ld : ", loop);
    dumpbits (x, 16);
    delay (duration);
    if (kbhit () || step) {
      c = getch ();
      if (c == 27)
        break;
      else if (c == ' ')
        step = !step;
      else if (c == '+') {
        if (duration > 1)
          duration >>= 1;
      }
      else if (c == '-') {
        if (duration < 512)
          duration <<= 1;
      }
      else if (!step)
        getch ();
    }
  } // for (loop)
  if (kbhit ())
    getch ();
```



```
}

char *ssid2str (int ssid)
{
  switch (ssid) {
    case PCI_SID_AD7899AR1: return "AD7899AR-1";
    case PCI_SID_AD7899BR1: return "AD7899BR-1";
    case PCI_SID_AD7899SR1: return "AD7899SR-1";
    case PCI_SID_AD7899AR2: return "AD7899AR-2";
    case PCI_SID_AD7899AR3: return "AD7899AR-3";
    case PCI_SID_AD7899BR3: return "AD7899BR-3";
    case PCI_SID_AD7899ARS1: return "AD7899ARS-1";
    case PCI_SID_AD7899ARS2: return "AD7899ARS-2";
    case PCI_SID_AD7899ARS3: return "AD7899ARS-3";
    default: return "unknown";
  }
}

void main()
{
  int r;
  int flag = 1;
  char ch;

  while (flag) {
    clrscr();
    adbit = 12;
    r =  GetPCICardInfo (PCI_DEVICE_ID_12ADDA, &ioport,
&irq);
    if (r)
      ioport = 0;
    if (ioport == 0) {
      adbit = 14;
```

```
    r =  GetPCICardInfo (PCI_DEVICE_ID_14ADDA, &ioport,
&irq);
    if (r)
      ioport = 0;
    else
      GetPCISubsystemID (PCI_DEVICE_ID_14ADDA, &ssid);
  }
  if (ioport == 0) {
    cputs ("PCI-12/14 bit AD/DA Test Program: Card not
found!\n\r");
    break;
  }
  cputs ("\n\n\n\r");
  cputs ("              PCI-12/14 bit AD/DA CARD Test Program
V1.1\n\r");
  cputs ("
=========================================\n\r");
  cprintf ("                  PCI %d bit AD/DA card found.\n\r",
adbit);
  cprintf ("                  I/O Address: %04X   IRQ: %d\n\r", ioport,
irq);
  if (adbit == 14)
    cprintf ("                  A/D: %s\n\r", ssid2str(ssid));
  cputs ("\n\r");
  cputs ("                (1): ADA Loopback test\n\r");
  cputs ("                (2): D/A Output test\n\r");
  cputs ("                (3): A/D Input test\n\r");
  cputs ("                (4): DIO Loopback test\n\r");
  cputs ("                (5): DIO Output test\n\r");
  cputs ("                (6): DIO Input test\n\r");
  cputs ("                (7): D/A Calibration\n\r");
  cputs ("                (8): A/D Calibration\n\r");
  cputs ("                ESC: EXIT\n\r");
```



```
  cputs ("                Choose:");
  do {
    ch = getch();
  } while (ch != '1' && ch != '2' && ch != '3'
      && ch != '4' && ch != '5' && ch != '6'
      && ch != '7' && ch != '8' && ch != 27);

  switch (ch) {
    case 27:
      flag = 0;
      break;
    case '1':
      Test_AD_Loopback ();
      break;
    case '2':
      Test_DA ();
      break;
    case '3':
      Test_AD ();
      break;
    case '4':
      Test_DIO_Loopback ();
      break;
    case '5':
      Test_DO ();
      break;
    case '6':
      Test_DI ();
      break;
    case '7':
      DA_Calibration ();
       break;
    case '8':
```

```
      AD_Calibration ();
       break;
   }
  }
} /* end of main */
```

PCI.H

```
#ifndef _PCI_H
#define _PCI_H

#define PCI_DEVICE_ID_2_PORT  0x0004
#define PCI_DEVICE_ID_4_PORT  0x0001
#define PCI_DEVICE_ID_8_PORT  0x0002
#define PCI_DEVICE_ID_16_PORT 0x0003

#define PCI_DEVICE_ID_2_8255            0x0101
#define PCI_DEVICE_ID_INDUSTRIAL    0x1011
#define PCI_DEVICE_ID_8_RELAY_8_PHOTO      0x1021
#define PCI_DEVICE_ID_4_RELAY_4_PHOTO      0x1022
#define PCI_DEVICE_ID_16_PHOTO      0x1023
#define PCI_DEVICE_ID_16_RELAY      0x1024

#define PCI_DEVICE_ID_12ADDA        0x0200
#define PCI_DEVICE_ID_14ADDA        0x0201


#define PCI_DEVICE_ID_INDUSTRIAL_OLD  0x0003

#define PCI_SID_AD7899AR1             0x0100
#define PCI_SID_AD7899AR2          0x0101
#define PCI_SID_AD7899AR3          0x0102
#define PCI_SID_AD7899BR1          0x0103
#define PCI_SID_AD7899BR3          0x0104
#define PCI_SID_AD7899SR1          0x0105
```



```
#define PCI_SID_AD7899ARS1          0x0106
#define PCI_SID_AD7899ARS2          0x0107
#define PCI_SID_AD7899ARS3          0x0108


#define PCI_VENDER_ID  0x6666


int IsPCIPresent(void);
unsigned int FindPCIDevice(unsigned int);
unsigned int FindPCIClassCode(unsigned int);
unsigned char ReadPCIConfigByte (unsigned int, int);
unsigned int ReadPCIConfigWord (unsigned int, int);
unsigned long ReadPCIConfigDWord (unsigned int, int);
int GetPCICardInfo (unsigned int deviceID, unsigned int *addr,
unsigned int *irq);
int GetPCISubsystemID (unsigned int deviceID, unsigned int *pid);

#endif /*_TYPEDEF_H*/
```

PCI.C

```
#include <stdlib.h>
#include <stdio.h>
#include <dos.h>

#include "pci.h"

int IsPCIPresent(void)
{
  union REGS r;

  r.x.ax = 0xb101;
  int86 (0x1a, &r, &r);
```

```
  return (r.h.ah == 0);
}

unsigned int FindPCIDevice(unsigned int deviceID)
{
  union REGS r;

  r.x.ax = 0xb102;
  r.x.cx = deviceID;
  r.x.dx = PCI_VENDER_ID;
  r.x.si = 0;
  int86 (0x1a, &r, &r);
  if (r.h.ah == 0)
    return r.x.bx;
  else
    return 0;
}

unsigned int FindPCIClassCode(unsigned int classCode)
{
  union REGS r;

  r.x.ax = 0xb103;
  r.x.cx = classCode;
  r.x.dx = PCI_VENDER_ID;
  r.x.si = 0;
  int86 (0x1a, &r, &r);
  if (r.h.ah == 0)
    return r.x.bx;
  else
    return 0;
}
```



```
unsigned char ReadPCIConfigByte (unsigned int pciaddr, int index)
{
  union REGS r;

  r.x.ax = 0xb108;
  r.x.bx = pciaddr;
  r.x.di = index;
  int86 (0x1a, &r, &r);
  if (r.h.ah == 0)
    return r.h.cl;
  else
    return 0;
}

unsigned int ReadPCIConfigWord (unsigned int pciaddr, int index)
{
  union REGS r;

  r.x.ax = 0xb109;
  r.x.bx = pciaddr;
  r.x.di = index;
  int86 (0x1a, &r, &r);
  if (r.h.ah == 0)
    return r.x.cx;
  else
    return 0;
}

unsigned long ReadPCIConfigDWord (unsigned int pciaddr, int
index)
{
  union REGS r;
```

```
  r.x.ax = 0xb10a;
  r.x.bx = pciaddr;
  r.x.di = index;
  int86 (0x1a, &r, &r);
  if (r.h.ah == 0)
    return r.x.cx;
  else
    return 0;
}

int GetPCICardInfo (unsigned int deviceID, unsigned int *addr,
unsigned int *irq)
{
  unsigned int pciaddr, index;

  if (!addr)
    return 1;

  if (!IsPCIPresent ())
    return 2;

  pciaddr = FindPCIDevice (deviceID);

  if (pciaddr == 0)
    return 3;

  for (index = 0x18; index <= 0x24; index += 4) {
    *addr = ReadPCIConfigDWord (pciaddr, index) & 0xfffe;
    if (*addr != 0)
      break;
  }
  *irq = ReadPCIConfigByte (pciaddr, 0x3c);
  return 0;
```



```
}

int GetPCISubsystemID (unsigned int deviceID, unsigned int *pid)
{
  unsigned int pciaddr, index;

  if (!IsPCIPresent ())
    return 2;

  pciaddr = FindPCIDevice (deviceID);

  if (pciaddr == 0)
    return 3;

  *pid = ReadPCIConfigWord (pciaddr, 0x2e);
  return 0;
}
```

APPENDIX A

WARRANTY INFORMATION

A.1 Copyright

Copyright DECISION COMPUTER INTERNATIONAL CO., LTD. All rights reserved. No part of PCI BUS 14 BIT DATA ACQUISITION CARD software and manual may be reproduced, transmitted, transcribed, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of DECISION COMPUTER INTERNATIONAL CO., LTD.

Each piece of PCI BUS 14 BIT DATA ACQUISITION CARD package permits user to use PCI BUS 14 BIT DATA ACQUISITION CARD only on a single computer, a registered user may use the program on a different computer, but may not use the program on more than one computer at the same time.

Corporate licensing agreements allow duplication and distribution of specific number of copies within the licensed institution. Duplication of multiple copies is not allowed except through execution of a licensing agreement. Welcome call for details.

A.2 Warranty Information

DECISION warrants that for a period of one year from the date of purchase (unless otherwise specified in the warranty card) that the goods supplied will perform according to the specifications defined in the user manual. Furthermore that the PCI BUS 14 BIT DATA ACQUISITION CARD product will be supplied free from defects in

materials and workmanship and be fully functional under normal usage.

In the event of the failure of a PCI BUS 14 BIT DATA ACQUISITION CARD product within the specified warranty period, DECISION will, at its option, replace or repair the item at no additional charge. This limited warranty does not cover damage resulting from incorrect use, electrical interference, accident, or modification of the product.

All goods returned for warranty repair must have the serial number intact. Goods without serial numbers attached will not be covered by the warranty.

Transportation costs for goods returned must be paid by the purchaser. Repaired goods will be dispatched at the expense of PCI BUS 14 BIT DATA ACQUISITION CARD.

To ensure that your PCI BUS 14 BIT DATA ACQUISITION CARD product is covered by the warranty provisions, it is necessary that you return the Warranty card.

Under this Limited Warranty, DECISION's obligations will be limited to repair or replacement only, of goods found to be defective as specified above during the warranty period. DECISION is not liable to the purchaser for any damages or losses of any kind, through the use of, or inability to use, the PCI BUS 14 BIT DATA ACQUISITION CARD product.

DECISION reserves the right to determine what constitutes warranty repair or replacement.

Return Authorization: It is necessary that any returned goods are clearly marked with an RA number that has been issued by DECISION. Goods returned without this authorization will not be attended to.

APPENDIX B

DATA SHEET

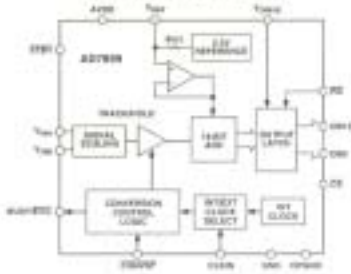
ANALOG DEVICES

5 V Single Supply 14-Bit 400 kSPS ADC

AD7899

FEATURES
Fast (2.2 µs) 14-Bit ADC
400 kSPS Throughput Rate
0.3 µs Track/Hold Acquisition Time
Single Supply Operation
Selection of Input Ranges: ±10 V, ±5 V and ±2.5 V
  0 V to 2.5 V and 0 V to 5 V
High-Speed Parallel Interface which Also Allows
  Interfacing to 3 V Processors
Low Power, 80 mW Typ
Power-Saving Mode, 28 µW Typ
Overvoltage Protection on Analog Inputs
Power-Down Mode via STBY Pin

FUNCTIONAL BLOCK DIAGRAM



GENERAL DESCRIPTION

The AD7899 is a fast, low-power, 14-bit A/D converter that operates from a single 5 V supply. The part contains a 2.2 µs successive-approximation ADC, a track/hold amplifier, 2.5 V reference, on-chip clock oscillator, signal conditioning circuitry, and a high-speed parallel interface. The part accepts analog input ranges of ±10 V, ±5 V, ±2.5 V, 0 V to 2.5 V, and 0 V to 5 V. Overvoltage protection on the analog input for this part allows the input voltage to be exceeded without damaging the parts.

Speed of conversion can be controlled either by an externally trimmed clock oscillator or by an external clock.

A conversion start signal ($\overline{CONVST}$) places the track/hold into hold mode and initiates conversion. The BUSY/$\overline{EOC}$ signal indicates the end of the conversion.

Data is read from the part via a 14-bit parallel data bus using the standard $\overline{CS}$ and $\overline{RD}$ signals. Maximum throughput for the AD7899 is 400 kSPS.

The AD7899 is available in a 28-lead SOIC and SSOP packages.

PRODUCT HIGHLIGHTS

1. The AD7899 features a fast (2.2 µs) ADC allowing throughput rates of up to 400 kSPS.
2. The AD7899 operates from a single 5 V supply and consumes only 80 mW typ making it ideal for low power and portable applications.
3. The part offers a high-speed parallel interface. The interface can operate in 3 V and 5 V mode allowing for easy connection to 3 V or 5 V microprocessors, microcontrollers, and digital signal processors.
4. The part is offered in three versions with different analog input ranges. The AD7899-1 offers the standard industrial ranges of ±10 V and ±5 V, the AD7899-2 offers a unipolar range of 0 V to 2.5 V or 0 V to 5 V, and the AD7899-3 has an input range of ±2.5 V.

REV. A

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 www.analog.com
Fax: 781/326-8703
© Analog Devices, Inc., 2001

AD7899–SPECIFICATIONS

(V_{DD} = 5 V ± 5%, AGND = DGND = 0 V, f_{CLK} = Internal. Clock = Internal, all specifications T_{min} to T_{max} and valid for V_{DRIVE} = 3 V ± 5% and 5 V ± 5% unless otherwise noted.)

Parameter	A Version[1]	B Version[1]	S Version[1]	Unit	Test Conditions/Comments
SAMPLE AND HOLD					
-0.1 dB Full Power Bandwidth	500	500	500	kHz typ	
-3 dB Full Power Bandwidth	4.5	4.5	4.5	MHz typ	
Aperture Delay	20	20	20	ns max	
Aperture Jitter	25	25	25	ps typ	
DYNAMIC PERFORMANCE[2]					f_{IN} = 100 kHz, f_S = 400 kSPS
AD7899-1					
Signal to (Noise + Distortion) Ratio[3] @ 25°C	78	78	78	dB min	
T_{MIN} to T_{MAX}	78	78	77	dB min	
Total Harmonic Distortion[3]	-84	-84	-82	dB max	
Peak Harmonic or Spurious Noise[3]	-86	-86	-85	dB max	
AD7899-2					
Signal to (Noise + Distortion) Ratio[3] @ 25°C	78			dB min	
T_{MIN} to T_{MAX}	77			dB min	
Total Harmonic Distortion[3]	-82			dB max	
Peak Harmonic or Spurious Noise[3]	-82			dB max	
AD7899-3					
Signal to (Noise + Distortion) Ratio[3] @ 25°C	78	78		dB min	
T_{MIN} to T_{MAX}	77	77		dB min	
Total Harmonic Distortion[3]	-84	-84		dB max	
Peak Harmonic or Spurious Noise[3]	-86	-86		dB max	
Intermodulation Distortion[3]					fa = 49 kHz, fb = 50 kHz
2nd Order Terms	-88	-88	-88	dB typ	
3rd Order Terms	-89	-89	-89	dB typ	
DC ACCURACY					
Resolution	14	14	14	Bits	
Relative Accuracy (INL)[4]	±2	±1.5	±2	LSB max	
Differential Nonlinearity (DNL)[5]	±1	±1	±1	LSB max	No Missing Codes Guaranteed
AD7899-1					
Input Voltage Range	±5, ±10	±5, ±10		Volts	
Input Current	0.8, 0.8	0.8, 0.8		mA max	V_{IN} = -5 V and -10 V Respectively
Positive Gain Error[6]	±10	±9	±12	LSB max	
Negative Gain Error[6]	±10	±9	±12	LSB max	
Bipolar Zero Error	±12	±9	±12	LSB max	
AD7899-2					
Input Voltage Range	0 to 2.5 0 to 5			Volts	
Input Current	0.4, 800			µA max	V_{IN} = 2.5 V, V_{IN} = 5 V
Positive Gain Error[6]	±14			LSB max	
Offset Error[7]	±10			LSB max	
AD7899-3					
Input Voltage Range	±2.5	±2.5		Volts	
Input Current	0.8	0.8		mA max	V_{IN} = -2.5 V
Positive Gain Error[6]	±14	±12		LSB max	
Negative Gain Error[6]	±14	±12		LSB max	
Bipolar Zero Error	±14	±12		LSB max	
REFERENCE INPUT/OUTPUT					
V_{REF} IN Input Voltage Range	2.375/2.625	2.375/2.625	2.375/2.625	V_{min}/V_{max}	2.5 V ± 5%
V_{REF} IN Input Capacitance[8]	10	10	10	pF max	
V_{REF} OUT Output Voltage	2.5	2.5	2.5	V (nom)	
V_{REF} OUT Error @ 25°C	±10	±10	±10	mV max	
V_{REF} OUT Error T_{MIN} to T_{MAX}	±25	±25	±25	mV max	
V_{REF} OUT Temperature Coefficient	25	25	25	ppm/°C typ	
V_{REF} OUT Output Impedance	6	6	6	Ω typ	See Reference Section

Parameter	A Version[1]	B Version[1]	S Version[1]	Unit	Test Conditions/Comments
LOGIC INPUTS					
Input High Voltage, V_{INH}	$V_{DRIVE}/2 + 0.4$	$V_{DRIVE}/2 + 0.4$	$V_{DRIVE}/2 + 0.4$	V min	V_{DD} = 5 V ± 5%
Input Low Voltage, V_{INL}	$V_{DRIVE}/2 - 0.6$	$V_{DRIVE}/2 - 0.6$	$V_{DRIVE}/2 - 0.6$	V max	V_{DD} = 5 V ± 5%
Input Current, I_{IN}	±10	±10	±10	μA max	
Input Capacitance, C_{IN}[2]	10	10	10	pF max	
LOGIC OUTPUTS					
Output High Voltage, V_{OH}	$V_{DRIVE} - 0.4$	$V_{DRIVE} - 0.4$	$V_{DRIVE} - 0.4$	V min	I_{SOURCE} = 400 μA
Output Low Voltage, V_{OL}	0.4	0.4	0.4	V max	I_{SINK} = 1.6 mA
DB13–DB0					
High Impedance					
Leakage Current	±10	±10	±10	μA max	
Capacitance[2]	10	10	10	pF max	
Output Coding					
AD7899-1, AD7899-3		Two's Complement			
AD7899-2		Straight (Natural) Binary			
CONVERSION RATE					
Conversion Time	2.2	2.2	2.2	μs max	
Track/Hold Acquisition Time[2, 3]	0.3	0.3	0.3	μs max	
Throughput Time	400	400	400	kSPS max	
POWER REQUIREMENTS					
V_{DD}	5	5	5	V nom	
I_{DD}					
Normal Mode	25	25	25	mA max	Typically 16 mA
Standby Mode	80	20	20	μA max	(5 μA typ) Logic Inputs = 0 V or V_{DD}
Power Dissipation					
Normal Mode	125	125	125	mW max	Typically 80 mW, V_{DD} = 5 V
Standby Mode	400	100	125	μW max	

NOTES
[1] Temperature Ranges are as follows: A, B Versions: –40°C to +85°C; S Version: –55°C to +125°C.
[2] Performance measured through full Control (MSB and ADC).
[3] See Terminology.
[4] Sample tested @ 25°C to ensure compliance.
Specifications subject to change without notice.

ABSOLUTE MAXIMUM RATINGS*
(T_A = 25°C unless otherwise noted)

V_{DD} to AGND ..... –0.3 V to +7 V
V_{DD} to DGND ..... –0.3 V to +7 V
V_{DRIVE} to DGND ..... V_{DD} + 0.3 V
Analog Input Voltage to AGND
AD7899-1 (±10 V Range) ..... ±18 V
AD7899-1 (±5 V Range) ..... –9 V to +18 V
AD7899-2 ..... –1 V to +18 V
AD7899-3 ..... –4 V to +18 V
Reference Input Voltage to AGND ..... –0.3 V to V_{DD} + 0.3 V
Digital Input Voltage to DGND ..... –0.3 V to V_{DD} + 0.3 V
Digital Output Voltage to DGND ..... –0.3 V to V_{DD} + 0.3 V
Operating Temperature Range
Commercial (A, B Version) ..... –40°C to +85°C
Military (S Version) ..... –55°C to +125°C
Storage Temperature Range ..... –65°C to +150°C
Junction Temperature ..... 150°C

SOIC Package, Power Dissipation ..... 450 mW
θ_{JA} Thermal Impedance ..... 95°C/W
Lead Temperature, Soldering
Vapor Phase (60 sec) ..... 215°C
Infrared (15 sec) ..... 220°C
SSOP Package, Power Dissipation ..... 450 mW
θ_{JA} Thermal Impedance ..... 95°C/W
Lead Temperature, Soldering
Vapor Phase (60 sec) ..... 215°C
Infrared (15 sec) ..... 220°C

*Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

CAUTION
ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the AD7899 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high-energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

ORDERING GUIDE

Model	Input Ranges	Relative Accuracy	Temperature Range	Package Description	Package Option
AD7899AR-1	±5 V, ±10 V	±2 LSB	–40°C to +85°C	Small Outline	R-28
AD7899BR-1	±5 V, ±10 V	±1.5 LSB	–40°C to +85°C	Small Outline	R-28
AD7899SR-1	±5 V, ±10 V	±2 LSB	–55°C to +125°C	Small Outline	R-28
AD7899AR-2	0 V to 5 V, 0 V to 2.5 V	±2 LSB	–40°C to +85°C	Small Outline	R-28
AD7899AR-3	±2.5 V	±2 LSB	–40°C to +85°C	Small Outline	R-28
AD7899BR-3	±2.5 V	±1.5 LSB	–40°C to +85°C	Small Outline	R-28
AD7899ARS-1	±5 V, ±10 V	±2 LSB	–40°C to +85°C	Shrink Small Outline	RS-28
AD7899ARS-2	0 V to 5 V, 0 V to 2.5 V	±2 LSB	–40°C to +85°C	Shrink Small Outline	RS-28
AD7899ARS-3	±2.5 V	±2 LSB	–40°C to +85°C	Shrink Small Outline	RS-28

AD7899

PIN FUNCTION DESCRIPTIONS

Pin No.	Mnemonic	Description
1	V_{REF}	Reference Input/Output. This pin is provides access to the internal reference (2.5 V ± 20 mV) and also allows the internal reference to be overdriven by an external reference source (2.5 V ± 5%). A 0.1 μF decoupling capacitor should be connected between this pin and GND.
2, 6	GND	Ground Pin. This pin should be connected to the system's analog ground plane.
3, 4	V_{IN0} V_{IN1}	Analog Inputs. See Analog Input Section.
5	V_{DD}	Positive Supply Voltage, 5.0 V ± 5%.
7–13	DB15–DB7	Data Bit 15 is the MSB, followed by Data Bit 12 to Data Bit 7. Three-state outputs.
14	OPGND	Output Driver Ground. This is the ground pin of the output drivers for D13 or D0 and BUSY/$\overline{EOC}$. It should be connected to the system's analog ground plane.
15	V_{DRIVE}	This pin provides the positive supply voltage for the digital inputs and outputs. It is normally tied to V_{DD}, but may also be powered by a 3 V ± 10% supply which allows the inputs and outputs to be interfaced to 3 V processors and DSPs. V_{DRIVE} should be decoupled with a 0.1 μF capacitor to GND.
16–22	DB6–DB0	Data Bit 6 to Data Bit 0. Three-state Outputs.
23	BUSY/$\overline{EOC}$	BUSY/$\overline{EOC}$ Output. Digital output pin used to signify that a conversion is in progress or that a conversion has finished. The function of the BUSY/$\overline{EOC}$ is determined by the state of $\overline{CONVST}$ at the end of conversion. See the Timing and Control Section.
24	$\overline{RD}$	Read Input. Active low logic input which is used in conjunction with $\overline{CS}$ low to enable the data outputs.
25	$\overline{CS}$	Chip Select Input. Active low logic input. The device is selected when this input is active.
26	$\overline{CONVST}$	Convert Start Input. Logic Input. A low to high transition on this input puts the track/hold into hold mode and starts conversion.
27	CLKIN	Conversion Clock Input. CLKIN is an externally applied clock which allows the user to control the conversion rate of the AD7899. If the CLKIN input is high on the rising edge of $\overline{CONVST}$ an externally applied clock will be used as the conversion clock. If the CLKIN is low on the rising edge of $\overline{CONVST}$ the internal laser-trimmed oscillator is used as the conversion clock. Each conversion needs sixteen clock cycles in order for the conversion to be completed. The externally applied clock should have a duty cycle no greater than 60/40. The CLKIN pin can be tied to GND if an external clock is not required.
28	$\overline{STBY}$	Standby Mode Input. Logic input which is used to put the device into the power save or standby mode. The $\overline{STBY}$ input is high for normal operation and low for standby operation.

PIN CONFIGURATION
SOIC/SSOP

TERMINOLOGY

Signal to (Noise + Distortion) Ratio
This is the measured ratio of signal to (noise + distortion) at the output of the A/D converter. The signal is the rms amplitude of the fundamental. Noise is the rms sum of all nonfundamental signals up to half the sampling frequency ($f_S/2$), excluding dc. The ratio is dependent upon the number of quantization levels in the digitization process; the more levels, the smaller the quantization noise. The theoretical signal to (noise + distortion) ratio for an ideal N-bit converter with a sine wave input is given by:

$$Signal\ to\ (Noise + Distortion) = (6.02N + 1.76)\ dB$$

Thus for a 14-bit converter, this is 98.04 dB.

Total Harmonic Distortion
Total harmonic distortion (THD) is the ratio of the rms sum of harmonics to the fundamental. For the AD7899 it is defined as:

$$THD\ (dB) = 20 \log \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + V_5^2 + V_6^2}}{V_1}$$

where V_1 is the rms amplitude of the fundamental and V_2, V_3, V_4, and V_5 are the rms amplitudes of the second through the fifth harmonics.

Peak Harmonic or Spurious Noise
Peak harmonic or spurious noise is defined as the ratio of the rms value of the next largest component in the ADC output spectrum (up to $f_S/2$ and excluding dc) to the rms value of the fundamental. Normally, the value of this specification is determined by the largest harmonic in the spectrum, but for parts where the harmonics are buried in the noise floor, it will be a noise peak.

Intermodulation Distortion
With inputs consisting of sine waves at two frequencies, fa and fb, any active device with nonlinearities will create distortion products at sum and difference frequencies of mfa ± nfb where m, n = 0, 1, 2, 3, etc. Intermodulation terms are those for which neither m or n are equal to zero. For example, the second order terms include (fa + fb) and (fa – fb), while the third order terms include (2fa + fb), (2fa – fb), (fa + 2fb) and (fa – 2fb).

The AD7899 is tested using two input frequencies. In this case, the second and third order terms are of different significance. The second order terms are usually distanced at frequencies from the original sine waves while the third order terms are usually at a frequency close to the input frequencies. As a result, the second and third order terms are specified separately. The calculation of the intermodulation distortion is as per the THD specification where it is the ratio of the rms sum of the individual distortion products to the rms amplitude of the fundamental expressed in dBs.

Differential Nonlinearity
This is the difference between the measured and the ideal 1 LSB change between any two adjacent codes in the ADC.

Positive Gain Error (AD7899-1, AD7899-3)
This is the deviation of the last code transition (01 . . . 110 to 01 . . . 111) from the ideal $4 \times V_{REF} - 3/2$ LSB (AD7899 at ±10 V_A, $2 \times V_{REF} - 3/2$ LSB (AD7899 at ±5 V range) or $V_{REF} - 3/2$ LSB (AD7899 at ±2.5 V range) after the Bipolar Offset Error has been adjusted out.

Positive Gain Error (AD7899-2)
This is the deviation of the last code transition (11 . . . 110 to 11 . . . 111) from the ideal $2 \times V_{REF} - 3/2$ LSB (AD7899 at ±10 V_A, $2 \times V_{REF} - 3/2$ LSB (AD7899 at 0 V or 5 V range) or $V_{REF} - 3/2$ LSB (AD7899 at 0 V to 2.5 V range) after the Unipolar Offset Error has been adjusted out.

Unipolar Offset Error (AD7899-2)
This is the deviation of the first code transition (00 . . . 00 to 00 . . . 01) from the ideal AGND +1/2 LSB.

Bipolar Zero Error (AD7899-1, AD7899-3)
This is the deviation of the midscale transition (all 0s or all 1s) from the ideal AGND – 1/2 LSB.

Negative Gain Error (AD7899-1, AD7899-3)
This is the deviation of the first code transition (10 . . . 000 to 10 . . . 001) from the ideal $-4 \times V_{REF} + 1/2$ LSB (AD7899 at ±10 V_A, $-2 \times V_{REF} + 1/2$ LSB (AD7899 at ±5 V range) or $-V_{REF} + 1/2$ LSB (AD7899 at ±2.5 V range) after Bipolar Zero Error has been adjusted out.

Track/Hold Acquisition Time
Track/Hold acquisition time is the time required for the output of the track/hold amplifier to reach its final value, within ±1/2 LSB, after the end of conversion (the point at which the track/hold returns to track mode). It also applies to situations where there is a step input change on the input voltage applied to the selected V_{IN0}/V_{IN1} input of the AD7899. It means that the user must wait for the duration of the track/hold acquisition time after the end of conversion or after a step input change to V_{IN0}/V_{IN1} before starting another conversion, to ensure that the part operates to specification.

CONVERTER DETAILS

The AD7899 is a high-speed, low-power, 14-bit A/D converter that operates from a single 5 V supply. The part contains a 2.2 μs successive-approximation ADC, track/hold amplifier, an internal 2.5 V reference and a high-speed parallel interface. The part accepts an analog input range of ±10 V or ±5 V (AD7899-1), 0 V to 2.5 V or 0 V to 5 V (AD7899-2) and ±2.5 V (AD7899-3). Overvoltage protection on the analog inputs for the part allows the input voltage to go to ±18 V (AD7899-2 with ±10 V input range), −9 V to +18 V (AD7899-1 with ±5 V input range), −1 V to +10 V (AD7899-2) and −4 V to +10 V (AD7899-3) without causing damage.

A conversion is initiated on the AD7899 by pulsing the $\overline{CONVST}$ input. On the rising edge of $\overline{CONVST}$, the on-chip track/hold is placed into hold and the conversion is started. The BUSY/$\overline{EOC}$ output signal is triggered high on the rising edge of $\overline{CONVST}$ and will remain high for the duration of the conversion sequence. The conversion clock for the part is generated internally using a laser-trimmed clock oscillator circuit. There is also the option of using an external clock. An external monotonous clock is applied to the CLKIN pin. If, on the rising edge of $\overline{CONVST}$, this input is high, the external clock will be used. The external clock should not start until 100 ns after the rising edge of $\overline{CONVST}$. The optimum throughput is obtained by using the internally generated clock—see Using an External Clock. The BUSY/$\overline{EOC}$ signal indicates the end of the conversion, and at this time the Track and Hold returns to tracking mode. The conversion results can be read at the end of the conversion (indicated by BUSY/$\overline{EOC}$ going low) via a 14-bit parallel data bus with standard $\overline{CS}$ and $\overline{RD}$ signals—see Timing and Control.

Conversion time for the AD7899 is 2.2 μs and the track/hold acquisition time is 0.5 μs. To obtain optimum performance from the part, the read operation should not occur during a conversion or during the 150 ns prior to the next $\overline{CONVST}$ rising edge. This allows the part to operate at throughput rates up to 400 kHz and achieve data sheet specifications.

CIRCUIT DESCRIPTION

Track/Hold Section

The track/hold amplifier on the AD7899 allows the ADCs to accurately convert an input sine wave of full-scale amplitude to 14-bit accuracy. The input bandwidth of the track/hold is greater than the Nyquist rate of the ADC even when the ADC is operated at its maximum throughput rate of 400 kSPS (i.e., the track/hold can handle input frequencies in excess of 200 kHz).

The track/hold amplifier's acquire input signals to 14-bit accuracy in less than 500 ns. The operation of the track/hold is essentially transparent to the user. The track/hold amplifier samples the input channel on the rising edge of $\overline{CONVST}$. The aperture time for the track/hold (i.e., the delay time between the external $\overline{CONVST}$ signal and the track/hold actually going into hold) is typically 15 ns and, more importantly, is well matched from device to device. It allows multiple AD7899s to sample more than one channel simultaneously. At the end of a conversion, the part returns to its tracking mode. The acquisition time of the track/hold amplifier begins at this point.

Reference Section

The AD7899 contains a single reference pin, labelled V_{REF}, which either provides access to the part's own 2.5 V reference or allows an external 2.5 V reference to be connected to provide the reference source for the part. The part is specified with a 2.5 V reference voltage.

To use the internal reference as the reference source for the AD7899, simply connect a 0.1 μF capacitor from the V_{REF} pin to AGND. The voltage that appears at this pin is internally buffered before being applied to the ADC. If this reference is required for use external to the AD7899, it should be buffered, as the part has a FET switch in series with the reference output resulting in a source impedance for this output of 6 kΩ nominal. The tolerance on the internal reference is ±10 mV at 25°C with a typical temperature coefficient of 25 ppm/°C and a maximum error over temperature of ±20 mV.

If the application requires a reference with a tighter tolerance or the AD1868 needs to be used with a system reference, the user has the option of connecting an external reference to this V_{REF} pin. The external reference will effectively overdrive the internal reference and thus provide the reference source for the ADC. The reference input is buffered before being applied to the ADC with the maximum input current of ± 100 μA. Suitable reference sources for the AD7899 include the AD680, AD780, REF192, and REF43 precision 2.5 V references.

Analog Input Section

The AD7899 is offered in three part types, the AD7899-1 where the input can be configured for ±10 V or a ±5 V input voltage range, the AD7899-2 where the input can be configured for 0 V to 5 V or a 0 V to 2.5 V input voltage range and the AD7899-3 which handles input voltage range ±2.5 V. The amount of current flowing into the analog input will depend on the analog input range and the analog input voltage. The maximum current flows when negative full-scale is applied.

AD7899-1

Figure 2 shows the analog input section of the AD7899-1. The input can be configured for ±5 V or ±10 V operation on the AD7899-1. For ±5 V operation, the V_{IN1} and V_{IN2} inputs are tied together and the input voltage is applied to both. For ±10 V operation, the V_{IN2} input is tied to AGND and the input voltage is applied to the V_{IN1} input. The V_{IN1} and V_{IN2} inputs are symmetrical and fully interchangeable.

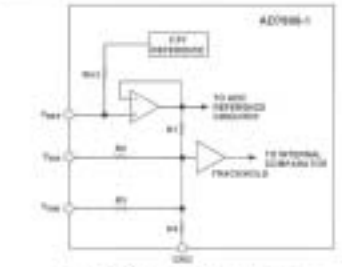


Figure 2. AD7899-1 Analog Input Structure

For the AD7899-1, R1 = 4 kΩ, R2 = 16 kΩ, R3 = 16 kΩ and R4 = 8 kΩ. The resistor input stage is followed by the high input impedance stage of the track/hold amplifier.

The designed code transitions take place midway between successive integer LSB values (i.e., 1/2 LSB, 3/2 LSBs, 5/2 LSBs etc.) LSB size is given by the formula, 1 LSB = FSR/16384. For the ±5 V range, 1 LSB = 10 V/16384 = 610.3 μV. For the ±10 V range, 1 LSB = 20 V/16384 = 1.22 mV. Output coding is two's complement binary with 1 LSB = FSR/16384. The ideal input/output transfer function for the AD7899-1 is shown in Table I.

Table I. Ideal Input/Output Code Table for the AD7899-1

Analog Input[1]	Digital Output Code Transition
+FSR/2 − 3/2 LSB[2]	011 . . . 110 to 011 . . . 111
+FSR/2 − 5/2 LSB	011 . . . 101 to 011 . . . 110
+FSR/2 − 7/2 LSB	011 . . . 100 to 011 . . . 101
GND + 3/2 LSB	000 . . . 001 to 000 . . . 010
GND + 1/2 LSB	000 . . . 000 to 000 . . . 001
GND − 1/2 LSB	111 . . . 111 to 000 . . . 000
GND − 3/2 LSB	111 . . . 110 to 111 . . . 111
−FSR/2 + 5/2 LSB	100 . . . 010 to 100 . . . 011
−FSR/2 + 3/2 LSB	100 . . . 001 to 100 . . . 010
−FSR/2 + 1/2 LSB	100 . . . 000 to 100 . . . 001

NOTES
[1]FSR is full-scale range and is 20 V for the ±10 V range and 10 V for the ±5 V range. nom V_{REF} = 2.5 V.
[2]1 LSB = FSR/16384 = 1.22 mV/ 610.3 μV — AD7899-2 and 610.3 μV for the AD7899-1 with V_{REF} = 2.5 V.

AD7899-2

Figure 3 shows the analog input section of the AD7899-2. Each input can be configured for 0 V to 5 V operation or 0 V to 2.5 V operation. For 0 V to 5 V operation, the V_{IN2} input is tied to GND and the input voltage is applied to the V_{IN1} input. For 0 V to 2.5 V operation, the V_{IN1} and V_{IN2} inputs are tied together and the input voltage is applied to both. The V_{IN1} and V_{IN2} inputs are symmetrical and fully interchangeable.

For the AD7899-2, R1 = 4 kΩ and R2 = 4 kΩ. Once again, the designed code transitions occur on successive integer LSB values. Output coding is straight (natural) binary with 1 LSB = FSR/16384 = 2.5 V/16384 = 0.153 mV, and 5 V/16384 = 0.305 mV, for the 0 to 2.5 V and the 0 to 5 V options respectively. Table II shows the ideal input and output transfer function for the AD7899-2.

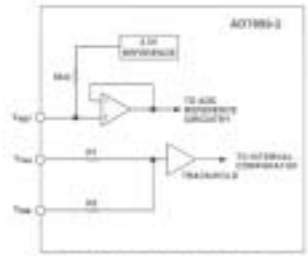


Figure 3. AD7899-2 Analog Input Structure

Table II. Ideal Input/Output Code Table for the AD7899-2

Analog Input[1]	Digital Output Code Transition
+FSR − 3/2 LSB[2]	111 . . . 110 to 111 . . . 111
+FSR − 5/2 LSB	111 . . . 101 to 111 . . . 110
+FSR − 7/2 LSB	111 . . . 100 to 111 . . . 101
GND + 3/2 LSB	000 . . . 010 to 000 . . . 011
GND + 1/2 LSB	000 . . . 001 to 000 . . . 010
GND − 1/2 LSB	000 . . . 000 to 000 . . . 001

NOTES
[1]FSR is Full-Scale Range and is 5 V or 2.5 V and 0 V or 0 V for AD7899-2 with V_{REF} = 2.5 V.
[2]1 LSB = FSR/16384 and is 0.153 mV (0 to 2.5 V) and 0.305 mV (0 to 5 V) for AD7899-2 with V_{REF} = 2.5 V.

AD7899-3

Figure 4 shows the analog input section of the AD7899-3. The analog input range is ±2.5 V on the V_{IN3} input. The V_{IN3} input can be left unconnected but if it is connected to a potential then that potential must be GND.

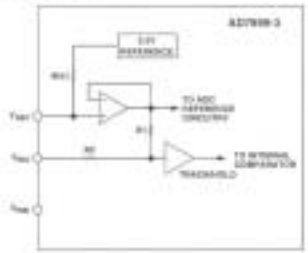


Figure 4. AD7899-3 Analog Input Structure

For the AD7899-3, R1 = 4 kΩ and R2 = 4 kΩ. The resistor input stage is followed by the high input impedance stage of the track/hold amplifier.

The designed code transitions take place midway between successive integer LSB values (i.e., 1/2 LSB, 3/2 LSBs, 5/2 LSBs etc.) LSB size is given by the formula, 1 LSB = FSR/16384. Output coding is two's complement binary with 1 LSB = FSR/16384 = 5 V/16384 = 610.4 μV. The ideal input/output transfer function for the AD7899-3 is shown in Table III.

Table III. Ideal Input/Output Code Table for the AD7899-3

Analog Input[1]	Digital Output Code Transition
+FSR/2 − 3/2 LSB[2]	011 . . . 110 to 011 . . . 111
+FSR/2 − 5/2 LSB	011 . . . 101 to 011 . . . 110
+FSR/2 − 7/2 LSB	011 . . . 100 to 011 . . . 101
GND + 3/2 LSB	000 . . . 001 to 000 . . . 010
GND + 1/2 LSB	000 . . . 000 to 000 . . . 001
GND − 1/2 LSB	111 . . . 111 to 000 . . . 000
GND − 3/2 LSB	111 . . . 110 to 111 . . . 111
−FSR/2 + 5/2 LSB	100 . . . 010 to 100 . . . 011
−FSR/2 + 3/2 LSB	100 . . . 001 to 100 . . . 010
−FSR/2 + 1/2 LSB	100 . . . 000 to 100 . . . 001

NOTES
[1] FSR is full-scale range is 5 V, with V_{REF} = 2.5 V.
[2] 1 LSB = FSR/16384 = 610.4 μV) (2.5 V – AD7899-3) with V_{REF} = 2.5 V.

TIMING AND CONTROL

Starting a Conversion

The conversion is initiated by applying a rising edge to the CONVST signal. This places the track/hold into hold mode and starts the conversion. The status of the conversion is indicated by the dual function signal BUSY/EOC. The AD7899 can operate in two conversion modes, EOC (End Of Conversion) mode and BUSY mode. The operating mode is determined by the state of CONVST at the end of the conversion.

Selecting a Conversion Clock

The AD7899 has an internal laser trimmed oscillator which can be used to control the conversion process. Alternatively an external clock source can be used to control the conversion process. The highest external clock frequency allowed is 6.5 MHz. This means a conversion time of 2.46 μs compared to 3.2 μs using the internal clock. However in some instances it may be useful to use an external clock when high throughput rates are not required. For example two or more AD7899s may be synchronized by using the same external clock for all devices. In this way there is no latency between output logic signals due to differences in the frequency of the internal clock oscillators.

On the rising edge of CONVST the AD7899 will examine the status of the CLKIN pin. If this pin is low it will use the internal laser trimmed oscillator as the conversion clock. If the CLKIN pin is high the AD7899 will wait for an external clock to be supplied to this pin which will then be used as the conversion clock. The first falling edge of the external clock should not happen for at least 100 ns after the rising edge of CONVST to ensure correct operation. Figure 5 shows how the BUSY/EOC output is synchronized to the CLKIN signal. Each conversion requires 16 clocks. The result of the conversion is transferred to the output data register on the falling edge of the 15th clock cycle. When the internal clock is selected the status of the CLKIN pin is free to change during conversion but the CLKIN setup and hold times must be observed in order to ensure that the correct conversion clock is used. The CLKIN pin can also be tied low permanently if the internal conversion clock is to be used.

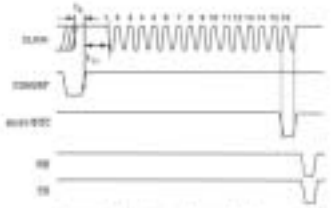


Figure 5. Using an External Clock

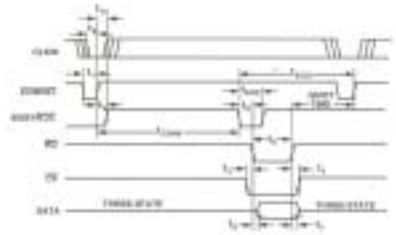


Figure 6. Conversion Sequence Timing Diagram (EOC Mode)

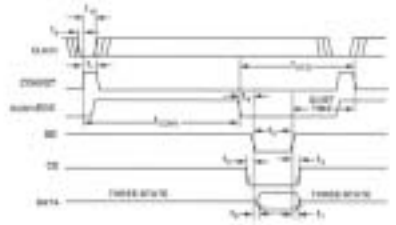


Figure 7. Conversion Sequence Timing Diagram (BUSY Mode)

EOC Mode

The CONVST signal is normally high. Pulsing the CONVST low will initiate a conversion on its rising edge. The state of the CONVST signal is checked at the end of conversion. Since the CONVST will be high when this happens the AD7899 BUSY/EOC pin will take on its EOC function and bring the BUSY/EOC line low for one clock period before returning high again. In this mode the EOC can be tied to the RD and CS signals to allow automatic reading of the conversion result if required. The timing diagram for operation in EOC mode is shown in Figure 6.

BUSY Mode

The CONVST signal is normally low. Pulsing the CONVST high will initiate a conversion on its rising edge. The state of the CONVST signal is checked at the end of conversion. Since the CONVST will be low when this happens the AD7899 BUSY/EOC pin will take on its BUSY function will bring BUSY/EOC low, indicating that the conversion is complete. BUSY/EOC will remain low until the next rising edge of CONVST where BUSY/EOC returns high. The timing diagram for operation in BUSY mode is shown in Figure 7.

Continuous Conversion Mode

When the AD7899 is used with an external clock, connecting the CLKIN and CONVST signals together will cause the AD7899 to continuously perform conversions. As each conversion completes the BUSY/EOC pin will pulse low for one clock period (EOC function) indicating that the conversion result is available. Figure 8 shows the timing and control sequence of the AD7899 in Continuous Conversion Mode.

Reading Data from the AD7899

Data is read from the part via a 14-bit parallel data bus with standard CS and RD signals. The CS and RD inputs are internally gated to enable the conversion result onto the data bus. The data lines DB0 to DB13 leave their high impedance state when both CS and RD are logic low. Therefore CS may be permanently tied logic low and the RD signal used to access the conversion result if required. Figures 6 and 7 show a timing specification called "Quiet Time." This is the amount of time which should be left after a read operation and before the next conversion is initiated. The quiet time depends heavily on data bus capacitance but a figure of 50 ns to 100 ns is typical, with a worst case figure of 150 ns.

Figure 8. Continuous Conversion Mode

Standby Mode Operation

The AD7899 has a Standby Mode whereby the device can be placed in a low current consumption mode (5 μA typ). The AD7899 is placed in Standby by bringing the logic input $\overline{STBY}$ low. The AD7899 can be powered again up for normal operation by bringing $\overline{STBY}$ logic high. The output data buffers are still operational while the AD7899 is in Standby. This means the user can still continue to access the conversion results while the AD7899 is in standby. This feature can be used to reduce the average power consumption in a system using low throughput rates. To reduce the average power consumption, the AD7899 can be placed in standby at the end of each conversion sequence and taken out of standby again prior to the start of the next conversion sequence. The time it takes the AD7899 to come out of standby is called the "wake up" time. This wake-up time will limit the maximum throughput rate at which the AD7899 can be operated when powering down between conversions. When the AD7899 is used with the internal reference, the reference capacitor will begin to discharge during standby. The voltage remaining on the capacitor at wake-up time will depend upon the standby time and hence affect the wake-up time. The minimum wake-up time is typically 2 μs. The maximum wake-up time will be when the AD7899 has been in standby long enough for the reference capacitor to fully discharge. The wake-up time in this case will typically be 15 ms. The AD7899 will wake up to approximately 1 μs when using an external reference, regardless of sleep time.

When operating the AD7899 in a Standby mode between conversions, the power savings can be significant. For example, with a throughput rate of 10 kSPS and an external reference, the AD7899 will be powered up for 4.2 μs out of every 100 μs (2 μs for wake-up time and 2.2 μs for conversion time). Therefore, the average power consumption drops to 88 mW + 4.1% or approximately 3.36 mW.

AD7899 DYNAMIC SPECIFICATIONS

The AD7899 is specified and 100% tested for dynamic performance specifications as well as traditional dc specifications such as Integral and Differential Nonlinearity. These ac specifications are required for the signal processing applications such as phased array sonar, adaptive filters, and spectrum analysis. These applications require information on the ADC's effect on the spectral content of the input signal. Hence, the parameters for which the AD7899 is specified include SNR, harmonic distortion, intermodulation distortion, and peak harmonics. These terms are discussed in more detail in the following sections.

Signal-to-Noise Ratio (SNR)

SNR is the measured signal-to-noise ratio at the output of the ADC. The signal is the rms magnitude of the fundamental. Noise is the rms sum of all the nonfundamental signals up to half the sampling frequency ($f_S/2$) excluding dc. SNR is dependent upon the number of quantization levels used in the digitization process; the more levels, the smaller the quantization noise. The theoretical signal to noise ratio for a sine wave input is given by

$$SNR = (6.02N + 1.76)\ dB \qquad (1)$$

where N is the number of bits.

Thus for an ideal 14-bit converter, SNR = 86.04 dB.

Figure 9 shows a histogram plot for 8192 conversions of a dc input using the AD7899 with 5 V supply. The analog input was set at the center of a code transition. It can be seen that most of the codes appear in one output bin, indicating very good noise performance from the ADC.

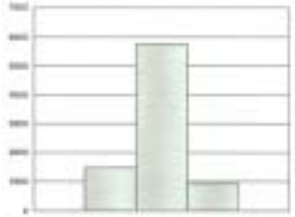


Figure 9. Histogram of 8192 Conversions of a DC Input

The output spectrum from the ADC is evaluated by applying a sine wave signal of very low distortion to the analog input. A Fast Fourier Transform (FFT) plot is generated from which the SNR data can be obtained. Figure 10 shows a typical 4096 point FFT plot of the AD7899 with an input signal of 100 kHz and a sampling frequency of 400 kHz. The SNR obtained from this graph is 86.5 dB. It should be noted that the harmonics are taken into account when calculating the SNR.

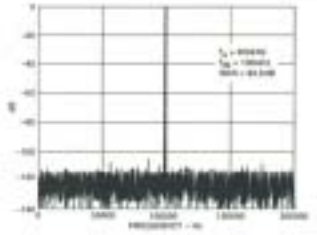


Figure 10. FFT Plot

Effective Number of Bits

The formula given in Equation 1 relates the SNR to the number of bits. Rewriting the formula, as in Equation 2, it is possible to obtain a measure of performance expressed in effective number of bits (N).

$$N = \frac{SNR - 1.76}{6.02} \qquad (2)$$

The effective number of bits for a device can be calculated directly from its measured SNR. Figure 11 shows a typical plot of effective number of bits versus frequency for an AD7899.

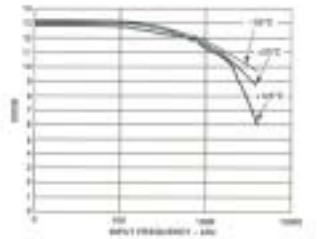


Figure 11. Effective Number of Bits vs. Frequency

Intermodulation Distortion

With inputs consisting of sine waves at two frequencies, fa and fb, any active device with nonlinearities will create distortion products at sum and difference frequencies of mfa ± nfb where m, n = 0, 1, 2, 3 . . . , etc. Intermodulation terms are those for which neither m nor n are equal to zero. For example, the second order terms include (fa + fb) and (fa − fb) while the third order terms include (2fa + fb), (2fa − fb), (fa + 2fb) and (fa − 2fb).

The AD7899 is tested using two input frequencies. In this case the second and third order terms are of different significance. The second order terms are usually distanced in frequency from the original sine waves while the third order terms are usually at a frequency close to the input frequencies. As a result, the second and third order terms are specified separately. The calculation of the intermodulation distortion is as per the THD specification where it is the ratio of the rms sum of the individual distortion products to the rms amplitude of the fundamental expressed in dBs. In this case, the input consists of two, equal amplitude, low distortion sine waves. Figure 12 shows a typical IMD plot for the AD7899.

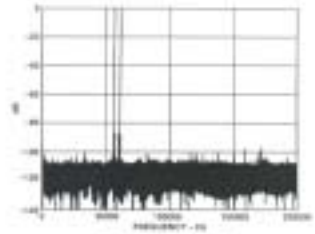


Figure 12. IMD Plot

AC Linearity Plots

The plots in Figure 13 show typical DNL and INL for the AD7899.

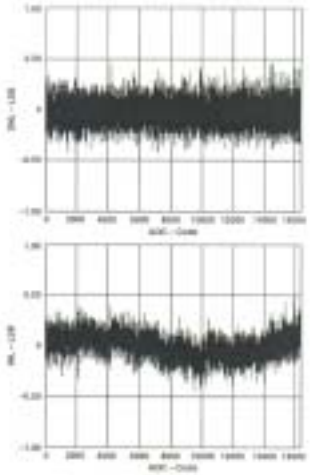


Figure 13. Typical DNL and INL Plots

MICROPROCESSOR INTERFACING

The high-speed parallel interface of the AD7899 allows easy interfacing to most DSPs and microprocessors. The AD7899 interface of the AD7899 consists of the data lines (DB0 to DB13), $\overline{CS}$, $\overline{RD}$, and BUSY/$\overline{EOC}$.

AD7899–ADSP-21xx Interface

Figure 14 shows an interface between the AD7899 and the ADSP-21xx. The $\overline{CONVST}$ signal can be generated by the ADSP-21xx or from some other external source. Figure 14 shows the $\overline{CS}$ being generated by a combination of the DMS signal and the address bus of the ADSP-21xx. In this way the AD7899 is mapped into the data memory space of the ADSP-21xx.

The AD7899 BUSY/$\overline{EOC}$ line provides an interrupt to the ADSP-21xx when the conversion is complete. The conversion result can then be read from the AD7899 using a read operation. The AD7899 is read using the following instruction:

$$AR0 = DM(ADC)$$

where *AR0* is the ADSP-21xx MR0 register and *ADC* is the AD7899 address.

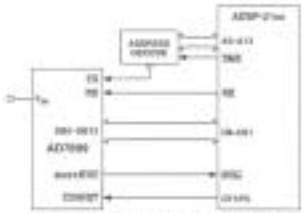


Figure 14. AD7899–ADSP-21xx Interface

AD7899–TMS320Cx Interface

Figure 15 shows an interface between the AD7899 and the TMS320Cx. As with the previous interfaces, conversion can be initiated from the TMS320Cx or from an external source and the processor is interrupted when the conversion sequence is completed. The $\overline{CS}$ signal to the AD7899 derived from the DS signal and a decode of the address bus. This maps the AD7899 into external data memory. The $\overline{RD}$ signal from the TMS320 is used to enable the ADC data onto the data bus. The AD7899 has a free parallel bus so there are no wait state requirements. The following instruction is used to read the conversion results from the AD7899:

$$DY\ DADC$$

where *D* is Data Memory address and *ADC* is the AD7899 address.

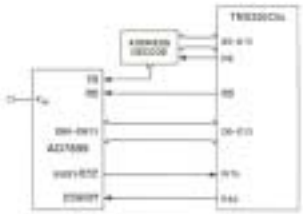
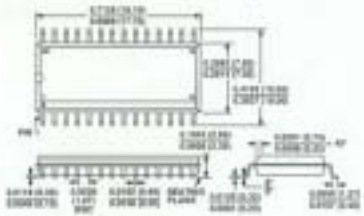


Figure 15. AD7899–TMS320Cx Interface

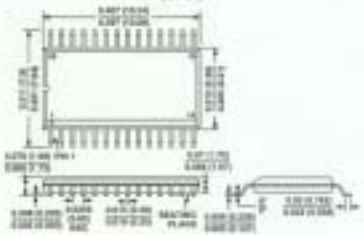
OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

28-Lead Small Outline
(R-28)



28-Lead Shrink Small Outline
(RS-28)



AD7899–Revision History

Location	Page
Data Sheet changed from REV. 0 to REV. A.	
Edit to Timing page heading	9
Edit to Converter Details section	8
Edit to Figure 14	14

ANALOG DEVICES

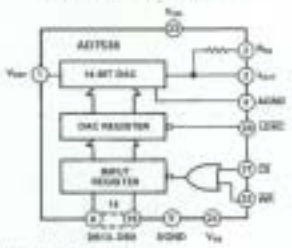
LC²MOS µP-Compatible 14-Bit DAC

AD7538

FEATURES
All Grades 14-Bit Monotonic Over the Full Temperature Range
Low Cost 14-Bit Upgrade for 12-Bit Systems
16-Bit Parallel Load with Double Buffered Inputs
Small 24-Pin, 0.3" DIP and SOIC
Low Output Leakage (<20 nA) Over the Full Temperature Range

APPLICATIONS
Microprocessor Based Control Systems
Digital Audio
Precision Servo Control
Control and Measurement in High Temperature Environments

FUNCTIONAL BLOCK DIAGRAM



GENERAL DESCRIPTION

The AD7538 is a 14-bit monolithic CMOS DAC converter which uses laser trimmed thin-film resistors to achieve excellent linearity.

The DAC is loaded by a single 14-bit wide word using standard Chip Select and Memory Write Logic. Double buffering, which is optional using LDAC, allows simultaneous update in a system containing multiple AD7538s.

A novel low leakage configuration (U.S. Patent No. 4,590,458) enables the AD7538 to exhibit excellent output leakage current characteristics over the specified temperature range.

The AD7538 is manufactured using the Linear Compatible CMOS (LC²MOS) process. It is speed compatible with most microprocessors and accepts TTL or CMOS logic level inputs.

PRODUCT HIGHLIGHTS

1. Guaranteed Monotonicity
   The AD7538 is guaranteed monotonic to 14 bits over the full temperature range for all grades.
2. Low Cost
   The AD7538, with its 14-bit dynamic range, offers a low cost solution for 12-bit system upgrades.
3. Small Package Size
   The AD7538 is packaged in a small 24-pin, 0.3" DIP and a 24-pin SOIC.
4. Low Output Leakage
   By tying V_{SS} (Pin 24) to a negative voltage, it is possible to achieve a low output leakage current at high temperatures.
5. Wide Power Supply Tolerance
   The device operates on a +12 V to +15 V V_{DD}, with a ±5% tolerance on this nominal figure. All specifications are guaranteed over this range.

REV. A

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 617/329-4700 Fax: 617/326-8703

AD7538–SPECIFICATIONS[1] (V_{DD} = +11.4 V to +15.75 V, V_{REF} = +10 V, V_{SS} = V_{DD} = 0 V, V_{IL} = –300 mV. All specifications T_{min} to T_{max} unless otherwise noted.)

Parameter	J, K Versions	A, B Versions	S Version	T Version	Units	Test Conditions/Comments
ACCURACY						
Resolution	14	14	14	14	Bits	
Relative Accuracy	±2	±1	±1	±1	LSB max	All Grades Guaranteed Monotonic
Differential Nonlinearity	±1	±1	±1	±1	LSB max	Over Temperature.
Full-Scale Error						Measured Using Internal R_{FB} DAC.
+25°C	±4	±4	±4	±4	LSB max	Register Loaded with All 1s.
T_{MIN} to T_{MAX}	±8	±5	±10	±8	LSB max	
Gain Temperature Coefficient[2]						
ΔGain/ΔTemperature	±3	±3	±3	±3	ppm/°C typ	
Output Leakage Current I_{OUT} (Pin 3)						
+25°C	±5	±5	±5	±5	nA max	All Digital Inputs 0 V
T_{MIN} to T_{MAX}	±50	±10	±20	±20	nA max	V_{IL} = –300 mV
T_{MIN} to T_{MAX}	±50	±10	±100	±100	nA max	V_{IH} = 0 V
REFERENCE INPUT						
Input Resistance, Pin 1	3.5	3.5	3.5	3.5	kΩ min	Typical Input Resistance = 5 kΩ.
	15	10	10	10	kΩ max	
DIGITAL INPUTS						
V_{IH} (Input High Voltage)	2.4	2.4	2.4	2.4	V min	
V_{IL} (Input Low Voltage)	0.8	0.8	0.8	0.8	V max	
I_{IN} (Input Current)						
+25°C	±1	±1	±1	±1	µA max	V_{IN} = 0 V or V_{DD}
T_{MIN} to T_{MAX}	±10	±10	±10	±10	µA max	
C_{IN} (Input Capacitance)[2]	7	7	7	7	pF max	
POWER SUPPLY						
V_{DD} Range	11.4/15.75	11.4/15.75	11.4/15.75	11.4/15.75	V min/V max	Specifications Guaranteed Over
V_{SS} Range	–(DD–10)0	–(DD–10)0	–(DD–30)0	–(DD–30)0	mV min/mV max	This Range
I_{DD}	4	4	4	4	mA max	All Digital Inputs V_{IL} or V_{IH}
	600	600	600	600	µA max	All Digital Inputs 0 V or V_{DD}

AC PERFORMANCE CHARACTERISTICS

These characteristics are included for Design Guidance only and are not subject to test. (V_{DD} = +11.4 V to +15.75 V, V_{REF} = +10 V, V_{RFB} = V_{SS} = 0 V, V_{IL} = 0 V or –300 mV, Output Amplifier is AD711 except where noted.)

Parameter	T_A = +25°C	T_A = T_{MIN}, T_{MAX}	Units	Test Conditions/Comments
Output Current Settling Time	1.5		µs max	To 0.003% of Full-Scale Range. I_{OUT} Load= 100 Ω, C_{OUT} = 13 pF. DAC Register Alternately Loaded with All 1s and All 0s. Typical Value of Settling Time Is 0.8 µs.
Digital to Analog Glitch Impulse	20		nV sec typ	Measured with V_{REF} = 0 V, I_{OUT} Load = 100 Ω, C_{OUT} = 13 pF. DAC Register Alternately Loaded with All 1s and All 0s.
Multiplying Feedthrough Error	1	1	mV p-p typ	V_{REF} = ±10 V, 10 kHz Sine Wave. DAC Register Loaded with All 0s.
Power Supply Rejection				
ΔGain/ΔV$_{DD}$	±0.01	±0.02	% per % max	ΔV$_{DD}$ = ±5%
Output Capacitance				
C_{OUT} (Pin 3)	260	280	pF max	DAC Register Loaded with All 1s
C_{OUT} (Pin 3)	130	130	pF max	DAC Register Loaded with All 0s
Output Noise Voltage Density (10 Hz–100 kHz)	13		nV/√Hz typ	Measured Between R_{FB} and I_{OUT1}

NOTES
Temperature range as follows: J, K Versions: 0°C to +70°C.
A, B Versions: –25°C to +85°C.
S, T Versions: –55°C to +125°C.
[1]Specifications are guaranteed for a V_{DD} of +11.4 V to +15.75 V. At V_{DD} = 5 V, the device is fully functional with degraded specifications.
[2]Sample tested to ensure compliance.
Specifications subject to change without notice.

–2–

REV. A

TIMING CHARACTERISTICS[1]

(V_{DD} = +11.4 V to +15.75 V, V_{REF} = +10 V, V_{OUT1} = V_{OUT2} = 0 V, R_{FB} = 0 Ω or −100 mV. All specifications T_{min} to T_{max} unless otherwise noted. See Figure 1 for Timing Diagram.)

Parameter	Limit at T_A = +25°C	Limit at T_A = 0°C to +70°C T_A = −25°C to +85°C	Limit at T_A = −55°C to +125°C	Units	Test Conditions/Comments
t_1	0	0	0	ns min	$\overline{CS}$ to $\overline{WR}$ Setup Time
t_2	0	0	0	ns min	$\overline{CS}$ to $\overline{WR}$ Hold Time
t_3	170	200	240	ns min	$\overline{LDAC}$ Pulse Width
t_4	170	200	240	ns min	Write Pulse Width
t_5	140	160	180	ns min	Data Setup Time
t_6	20	20	30	ns min	Data Hold Time

NOTES:
[1]Temperature range as follows: J, K Versions: 0°C to +70°C.
A, B Versions: −25°C to +85°C.
S, T Versions: −55°C to +125°C.
Specifications subject to change without notice.

ABSOLUTE MAXIMUM RATINGS*

(T_A = +25°C unless otherwise noted)

V_{DD} (Pin 23) to DGND ..... −0.3 V, +17 V
V_{SS} (Pin 24) to AGND ..... −15 V, +0.3 V
V_{OUT1} (Pin 1) to AGND ..... ±25 V
V_{OUT2} (Pin 2) to AGND ..... ±25 V
Digital Input Voltage (Pins 6–22)
to DGND ..... −0.3 V, V_{DD} +0.3 V
V_{REF} to DGND ..... −0.3 V, V_{DD} +0.3 V
AGND to DGND ..... −0.3 V, V_{DD} +0.3 V
Power Dissipation (Any Package)
To +75°C ..... 1000 mW
Derates Above +75°C ..... 10 mW/°C

Operating Temperature Range
Commercial (J, K Versions) ..... 0°C to +70°C
Industrial (A, B Versions) ..... −25°C to +85°C
Extended (S, T Versions) ..... −55°C to +125°C
Storage Temperature ..... −65°C to +150°C
Lead Temperature (Soldering, 10 sec) ..... +300°C

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the AD7538 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

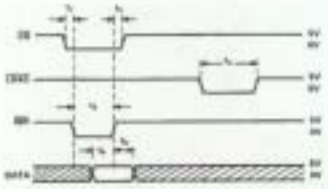


Figure 1. Timing Diagram

PIN CONFIGURATION
DIP, SOIC

TERMINOLOGY

RELATIVE ACCURACY
Relative accuracy or endpoint nonlinearity is a measure of the maximum deviation from a straight line passing through the endpoints of the DAC transfer function. It is measured after adjusting for zero error and full-scale error and is normally expressed in Least Significant Bits or as a percentage of full-scale reading.

DIFFERENTIAL NONLINEARITY
Differential nonlinearity is the difference between the measured change and the ideal 1 LSB change between any two adjacent codes. A specified differential nonlinearity of ±1 LSB max over the operating temperature range ensures monotonicity.

GAIN ERROR
Gain error is a measure of the output error between an ideal DAC and the actual device output. It is measured with all 1s in the DAC after offset error has been adjusted out and is expressed in Least Significant Bits. Gain error is adjustable to zero with an external potentiometer.

DIGITAL-TO-ANALOG GLITCH IMPULSE
The amount of charge injected from the digital inputs to the analog output when the inputs change state is called Digital-to-Analog Glitch Impulse. This is normally specified as the area of the glitch in either pA-secs or nV-secs depending upon whether the glitch is measured as a current or voltage. It is measured with V_{REF} = AGND.

OUTPUT CAPACITANCE
This is the capacitance from I_{OUT1} to AGND.

OUTPUT LEAKAGE CURRENT
Output Leakage Current is current which appears at I_{OUT1} with the DAC register loaded to all 0s.

MULTIPLYING FEEDTHROUGH ERROR
This is the ac error due to capacitive feedthrough from V_{REF} terminal to I_{OUT1} with DAC register loaded to all zeros.

ORDERING GUIDE

Model	Temperature Range	Relative Accuracy	Full-Scale Error	Package Option*
AD7538JN	0°C to +70°C	±2 LSB	±8 LSB	N-24
AD7538KN	0°C to +70°C	±1 LSB	±4 LSB	N-24
AD7538JR	0°C to +70°C	±2 LSB	±8 LSB	R-24
AD7538KR	0°C to +70°C	±1 LSB	±4 LSB	R-24
AD7538AQ	−25°C to +85°C	±2 LSB	±8 LSB	Q-24
AD7538BQ	−25°C to +85°C	±1 LSB	±4 LSB	Q-24
AD7538SQ	−55°C to +125°C	±2 LSB	±8 LSB	Q-24
AD7538TQ	−55°C to +125°C	±1 LSB	±4 LSB	Q-24

*N = Plastic DIP; Q = Cerdip; R = SOIC.

PIN FUNCTION DESCRIPTION

Pin	Mnemonic	Description
1	V_{OUT1}	Voltage Reference.
2	R_{FB}	Feedback Resistor. Used to close the loop around an external op amp.
3	I_{OUT1}	Current Output Terminal.
4	AGND	Analog Ground.
5	DGND	Digital Ground.
6–19	DB13–DB0	Data Inputs. Bit 13 (MSB) to Bit 0 (LSB).
20	$\overline{LDAC}$	Chip Select Input. Active LOW.
21	$\overline{CS}$	Asynchronous Load DAC Input. Active LOW.
22	$\overline{WR}$	Write Input. Active LOW.

$\overline{CS}$	$\overline{LDAC}$	$\overline{WR}$	OPERATION
0	1	0	Load Input Register.
1	0	X	Load DAC Register from Input Register.
0	0	0	Input and DAC Registers are Transparent.
1	1	X	No Operation.
X	1	1	No Operation.

NOTE: X (Don't Care)

23	V_{DD}	+12 V to +15 V supply input.
24	V_{SS}	Bias pin for High Temperature Low Leakage configuration. To implement low leakage system, this pin should be at a negative voltage. See Figures 4 and 5 for recommended circuitry.

DA SECTION

Figure 2 shows a simplified circuit diagram for the AD7538 DA section. The three MSBs of the 14-bit Data Word are decoded to drive the seven switches A-G. The 11 LSBs of the Data Word consist of an R-2R ladder operated in a current steering configuration.

The R-2R ladder current is 1/8 of the total reference input current. 7/8 I flows to the parallel ladder structure. Switches A-G steer equally weighted currents between I_{OUT} and AGND.

Since the input resistance at V_{REF} is constant, it may be driven by a voltage source or a current source of positive or negative polarity.

CIRCUIT INFORMATION

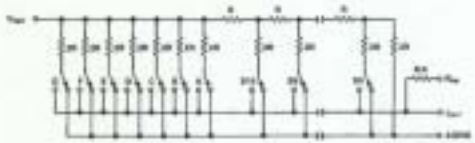


Figure 2. Simplified Circuit Diagram for the AD7538 DA Section

EQUIVALENT CIRCUIT ANALYSIS

Figure 3 shows an equivalent circuit for the analog section of the AD7538 DA converter. The current source $I_{LEAKAGE}$ is composed of surface and junction leakages. The resistor R_O denotes the equivalent output resistance of the DAC which varies with input code. C_{OUT} is the capacitance due to the current steering switches and varies from about 80 pF to 180 pF (typical values) depending upon the digital input. gV_{REF}. N is the Thevenin equivalent voltage generator due to the reference input voltage, V_{REF}, and the transfer function of the DAC ladder, N.

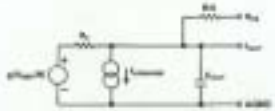


Figure 3. AD7538 Equivalent Analog Output Circuit

DIGITAL SECTION

The digital inputs are designed to be both TTL and 5 V CMOS compatible. All logic inputs are static protected MOS gates with typical input currents of less than 1 nA. To minimize power supply currents, it is recommended that the digital input voltages be driven as close as possible to 0 V and 5 V logic levels.

UNIPOLAR BINARY OPERATION (2-QUADRANT MULTIPLICATION)

Figure 4 shows the circuit diagram for unipolar binary operation. With an ac input, the circuit performs 2-quadrant multiplication. The code table for Figure 4 is given in Table I.

Capacitor C1 provides phase compensation and helps prevent overshoot and ringing when high speed op amps are used.

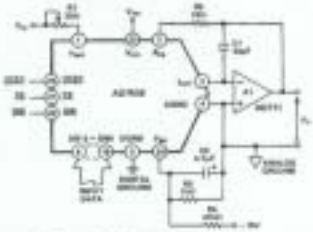


Figure 4. Unipolar Binary Operation

Table I. Unipolar Binary Code Table for AD7538

Binary Number In DAC Register MSB LSB	Analog Output, V_{OUT}
11 1111 1111 1111	$-V_{IN}\left(\frac{16383}{16384}\right)$
10 0000 0000 0000	$-V_{IN}\left(\frac{8192}{16384}\right) = -1/2\,V_{IN}$
00 0000 0000 0001	$-V_{IN}\left(\frac{1}{16384}\right)$
00 0000 0000 0000	0 V

For zero offset adjustment, the DAC register is loaded with all 0s and amplifier offset (V_{OS}) adjusted so that V_{OUT} is 0 V. Adjusting V_{OUT} to 0 V is not necessary in many applications, but it is recommended that V_{OS} be no greater than 125×10^{-6} (V_{REF}) to maintain specified DAC accuracy (see Applications Hints).

Full-scale trimming is accomplished by loading the DAC register with all 1s and adjusting R1 so that $V_{OUT} = -V_{IN}$ (16383/16384). For high temperature operation, resistors and potentiometers should have a low Temperature Coefficient. In many applications, because of the excellent Gain T.C. and Gain Error specifications of the AD7538, Gain Error trimming is not necessary. In fixed reference applications, full scale can also be adjusted by omitting R1 and R2 and trimming the reference voltage magnitude.

BIPOLAR OPERATION (4-QUADRANT MULTIPLICATION)

The recommended circuit diagram for bipolar operation is shown in Figure 5. Offset binary coding is used. The code table for Figure 5 is given in Table II.

With the DAC loaded to 10 0000 0000 0000, adjust R1 for $V_O = 0$ V. Alternatively, one can omit R1 and R2 and adjust the ratio of R5 and R8 for $V_O = 0$ V. Full-scale trimming can be accomplished by adjusting the amplitude of V_{IN} or by varying the value of R7.

The values given for R1, R2 are the minimum necessary to calibrate the system for resistor, R5, R6, R7 ratio matched to 0.1%. System linearity error is independent of resistor ratio matching and is affected by DAC linearity error only.

When operating over a wide temperature range, it is important that the resistors be of the same type so that their temperature coefficients match.

For further information see "CMOS DAC Application Guide", 3rd Edition. Publication Number G872b-8-1/88 available from Analog Devices.

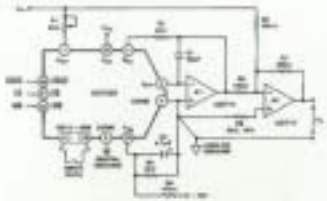


Figure 5. Bipolar Operation

LOW LEAKAGE CONFIGURATION

For CMOS Multiplying DA converters, as the device is operated at higher temperatures, the output leakage current increases. For a 14-bit resolution system, this can be a significant source of error. The AD7538 features a leakage reduction configuration (U.S. Patent No. 4,590,4581 to keep the leakage current low over an extended temperature range. One may operate the device with or without this configuration. If V_{OS} (Pin 24) is tied to AGND then the DAC will exhibit normal output leakage current at high temperatures. To use the low leakage facility,

Table II. Bipolar Code Table for Offset Binary Circuit of Figure 5.

Binary Number In DAC Register MSB LSB	Analog Output V_{OUT}
11 1111 1111 1111	$+V_{IN}\left(\frac{8191}{8192}\right)$
10 0000 0000 0001	$+V_{IN}\left(\frac{1}{8192}\right)$
10 0000 0000 0000	0 V
01 1111 1111 1111	$-V_{IN}\left(\frac{1}{8192}\right)$
00 0000 0000 0000	$-V_{IN}\left(\frac{8191}{8192}\right)$

V_{OS} should be tied to a voltage of approximately -0.3 V as in Figures 4 and 5. A simple resistor divider (R3, R4) produces approximately -300 mV from -15 V. The capacitor C2 in parallel with R3 is an integral part of the low leakage configuration and must be 4.7 µF or greater. Figure 6 is a plot of leakage current versus temperature for both conditions. It clearly shows the improvement gained by using the low leakage configuration.

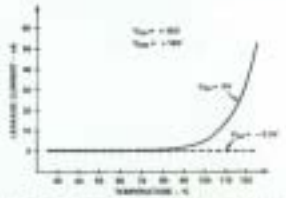


Figure 6. Graph of Typical Leakage Current vs. Temperature for AD7538

PROGRAMMABLE GAIN AMPLIFIER

The circuit shown in Figure 7 provides a programmable gain amplifier (PGA). In it the DAC behaves as a programmable resistance and thus allows the circuit gain to be digitally controlled.

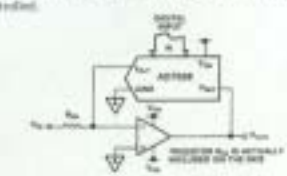


Figure 7. Programmable Gain Amplifier (PGA)

The transfer function of Figure 7 is:

$$Gain = \frac{V_{OUT}}{V_{IN}} = -\frac{R_{FB}}{R_{AB}} \quad (1)$$

R_{AB} is the equivalent transfer impedance of the DAC from the V_{REF} pin to the I_{OUT} pin and can be expressed as:

$$R_{AB} = \frac{2^n \cdot R_{IN}}{N} \quad (2)$$

Where: n is the resolution of the DAC

N is the DAC input code in decimal

R_{IN} is the constant input impedance of the DAC ($R_{IN} = R_{AB}$)

Substituting this expression into Equation 1 and assuming zero gain error for the DAC ($R_{FB} = R_{AB}$) the transfer function simplifies to:

$$\frac{V_{OUT}}{V_{IN}} = \frac{2^n}{N} \quad (3)$$

The ratio $N/2^n$ is commonly represented by the term D and, as such, is the fractional representation of the digital input word.

$$\frac{V_{OUT}}{V_{IN}} = -\frac{2^n}{N} = -\frac{1}{D} \quad (4)$$

Equation 4 indicates that the gain of the circuit can be varied from 16,384 down to unity (actually 16,384/16,383) in 16,383 steps. The all 0s code is never applied. This avoids an open-loop condition thereby saturating the amplifier. With the all 0s code excluded there remains $2^n - 1$ possible input codes allowing a choice of $2^n - 1$ output levels. In dB terms the dynamic range is:

$$20 \log_{10} \frac{V_{OUT}}{V_{IN}} = 20 \log_{10} (2^n - 1) = 84 \text{ dB}$$

APPLICATION HINTS

Output Offset: CMOS D/A converters in circuits such as Figures 4 and 5 exhibit a code dependent output resistance which in turn can cause a code dependent error voltage at the output of the amplifier. The maximum amplitude of this error, which adds to the D/A converter nonlinearity, depends on V_{OS}, where V_{OS} is the amplifier input offset voltage. To maintain specified accuracy with V_{REF} at 10 V, it is recommended that V_{OS} be no greater than 0.25 mV, or (25×10^{-6}) (V_{REF}), over the temperature range of operation. The AD711 is a suitable op amp. The op amp has a wide bandwidth and high slew rate and is recommended for ac and other applications requiring fast settling.

General Ground Management: Since the AD7538 is specified for high accuracy, it is important to use a proper grounding technique. AC or transient voltages between AGND and DGND can cause noise injection into the analog output. The simplest method of ensuring that voltages at AGND and DGND are equal is to tie AGND and DGND together at the AD7538. In more complex systems where the AGND and DGND interconnection is on the backplane, it is recommended that two diodes be connected in inverse parallel between the AD7538 AGND and DGND pins (1N914 or equivalent).

MICROPROCESSOR INTERFACING

The AD7538 is designed for easy interfacing to 16-bit microprocessors and can be treated as a memory mapped peripheral. This reduces the amount of external logic needed for interfacing to a minimal.

AD7538-8080 INTERFACE

Figure 8 shows the 8080 processor interface to a single device. In this setup the double buffering feature (using $\overline{LDAC}$) of the DAC is not used. The 14-bit word is written to the DAC in one MOV instruction and the analog output responds immediately.

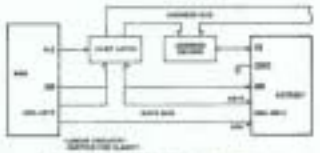


Figure 8. AD7538-8080 Interface Circuit

In a multiple DAC system the double buffering of the AD7538 allows the user to simultaneously update all DACs. In Figure 9, a 14-bit word is loaded to the Input Registers of each of the DACs in sequence. Then, with one instruction to the appropriate address, CS6 (i.e., LDAC) is brought low, updating all the DACs simultaneously.

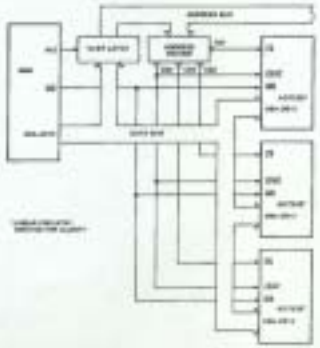


Figure 9. AD7538-8080 Interface: Multiple DAC System

AD7538-MC68000 INTERFACE

Figure 10 shows the MC68000 processor interface to a single device. In this setup the double buffering feature of the DAC is not used and the appropriate data is written into the DAC in one MOVE instruction.

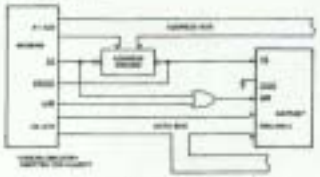


Figure 10. AD7538-MC68000 Interface

DIGITAL FEEDTHROUGH

The digital inputs to the AD7538 are directly connected to the microprocessor bus in the preceding interface configurations. These inputs will be constantly changing even when the device is not selected. The high frequency logic activity on the bus can feed through the DAC package capacitance to show up as noise on the analog output. To minimize this Digital Feedthrough isolate the DAC from the noise source. Figure 11 shows an interface circuit which uses this technique. All data inputs are latched from the bus by the $\overline{CS}$ signal. One may also use other means, such as peripheral interface devices, to reduce the Digital Feedthrough.

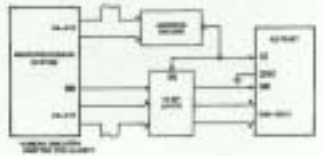
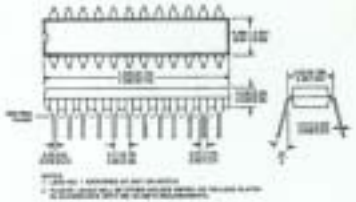
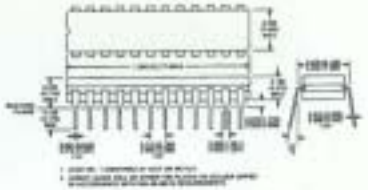


Figure 11. AD7538 Interface Circuit Using Latches to Minimize Digital Feedthrough

OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

24-Pin Plastic Suffix (N)



24-Pin Cerdip (Suffix Q)