IP-SOFTDAC

16-channel 16-bit
Digital/Analog Converter
Industry Pack Module

PROGRAMMING MANUAL

818-20-000-4000

Version 1.0

March 2006

ALPHI TECHNOLOGY CORPORATION
6202 S. Maple Avenue #120
Tempe, AZ 85283 USA
Tel: (480) 838-2428
Fax: (480) 838-4477

NOTICE

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

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

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

TABLE OF CONTENTS

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The **IP-SOFTDAC** is a high performance DIGITAL TO ANALOG module. The **IP-SOFTDAC** outputs 16 channels with a 16-bit resolution at a maximum settling time of 2 μS.

The primary features of the **IP-SOFTDAC** are as follows:

- 2 μSecond settling time (0 to 5 V)
- Six Programmable Output Ranges per channel
- Unipolar: 0V to 5V, 0V to 10V
- Bipolar Mode: ±5V, ±10V, ±2.5V,–2.5V to 7.5V
- 1LSB Max DNL and INL Over the Industrial Temperature Range
- Glitch Impulse < 2nV-s
- 16-Lead SSOP Package
- Power-On Reset to 0V
- Local 8kx8 Flash EPROM to store local user information
- Two stage buffers
- Global output buffer w/ internal or external triggering

1.2 FUNCTIONAL DESCRIPTION

The IP-SOFTDAC uses 16 Linear LTC 1592 D/A converters.

The Linear LTC1592 are serial input 16-bit multiplying current output DACs that operates from a single 5 Volt supply. These SoftSpan DACs can be software-programmed for either uni-polar or bi-polar mode through a 3-wire SPI interface. In either mode, the voltage output range can also be software-programmed. Two output ranges in uni-polar mode and four output ranges in bi-polar mode are available.

The DACs are accurate to 1LSB over the industrial temperature range in both uni-polar and bi-polar modes. True 16-bit 4-quadrant multiplication is achieved with on-chip four quadrant multiplication resistors.

These devices include an internal deglitcher circuit that reduces the glitch impulse to less than 2nV-s (typ).

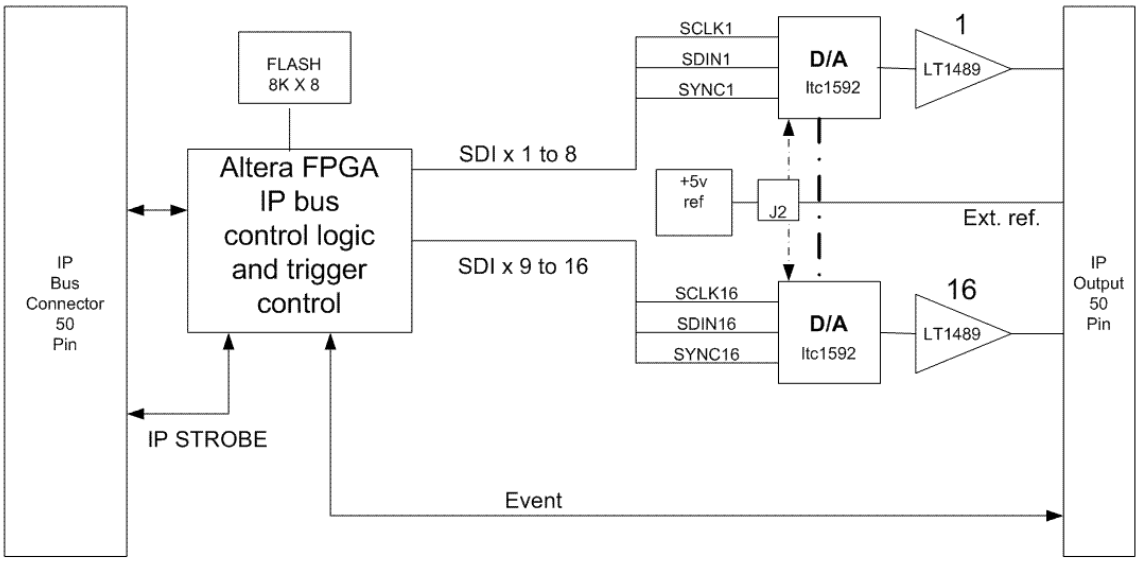

Figure 1.1: Block Diagram

2 INTERNAL ORGANIZATION

2.1 IP INTERFACE

2.1.1 IDSPACE

The on-board logic provides information about the module to the user. The lower address contains data related to the type of module, revision, etc...

ID space address	Description	Value
0x01	ASCII “I”	0x49
0x03	ASCII “P”	0x50
0x05	ASCII “A”	0x41
0x07	ASCII “H” (32 MHz)	0x48
0x09	Manufacturer identification	0x11
0x0B	Module type	0x23
0x0D	Revision module	0x0A
0x0F	Reserved	

Table 2-1 IDSPACE content 32MHz IP

ID space address	Description	Value
0x01	ASCII “I”	0x49
0x03	ASCII “P”	0x50
0x05	ASCII “A”	0x41
0x07	ASCII “H” (8 MHz)	0x43
0x09	Manufacturer identification	0x11
0x0B	Module type	0x23
0x0D	Revision module	0x0A
0x0F	Reserved	

Table 2-2 IDSPACE content 8MHz IP

2.1.2 IOSPACE

The **IP-SOFTDAC** module use 16 Linear Technology LTC1592 D/A converter.

A double buffered interface is use to transfer incoming data to the output.

Offset	Register Name	
0x00	Ch # 0	D/A # 0 data register
0x02	Ch # 1	D/A # 1 data register
0x04	Ch # 2	D/A # 2 data register
0x06	Ch # 3	D/A # 3 data register
0x08	Ch # 4	D/A # 4 data register
0x0A	Ch # 5	D/A # 5 data register
0x0C	Ch # 6	D/A # 6 data register
0x0E	Ch # 7	D/A # 7 data register
0x10	Ch # 8	D/A # 8 data register
0x12	Ch # 9	D/A # 9 data register
0x14	Ch # 10	D/A # 10 data register
0x16	Ch # 11	D/A # 11 data register
0x18	Ch # 12	D/A # 12 data register
0x1A	Ch # 13	D/A # 13 data register
0x1C	Ch # 14	D/A # 14 data register
0x1E	Ch # 15	D/A # 15 data register
0x40	Trigger	Writing to this address generates an internal trigger.
0x44	Control Register	Trigger configuration. Specifies which trigger to use, and if needed, where to output an external trigger
0x48	Command Register	Command sent to the DAC in the next access.

Table 2.1: SoftDAC IOSPACE Address Map

2.1.3 MEM SPACE

The SOFTDAC board contains an AT28C64 8K by 8 Flash EPROM.

It is available to the user to stored information related to the module offset gain error for eventual software correction

2.2 ANALOG OUTPUT

The **IP-SOFTDAC** has sixteen analog outputs each with its own buffer.

The output ranges are programmable using the board control register.

2.2.1 LTC1592 Command Structure

The LTC1592 receive serially a 24-bit input word. The 4 first bits are a command. The next 4 bits are unused. The last 16-bits are the value to be digitized.

The input word is send to one of the D/A when a write access takes place to the corresponding data register. The serialization logic on the SOFTDAC module use the 4 lower bits of the Command Register (0x48) as the 4 command bit, and the 16 bits being written as the value to be digitized.

It is not necessary to write into the command register between 2 data access. Whatever value was already there is used. Conversely, writing into the command register does not generate any access to the D/As.

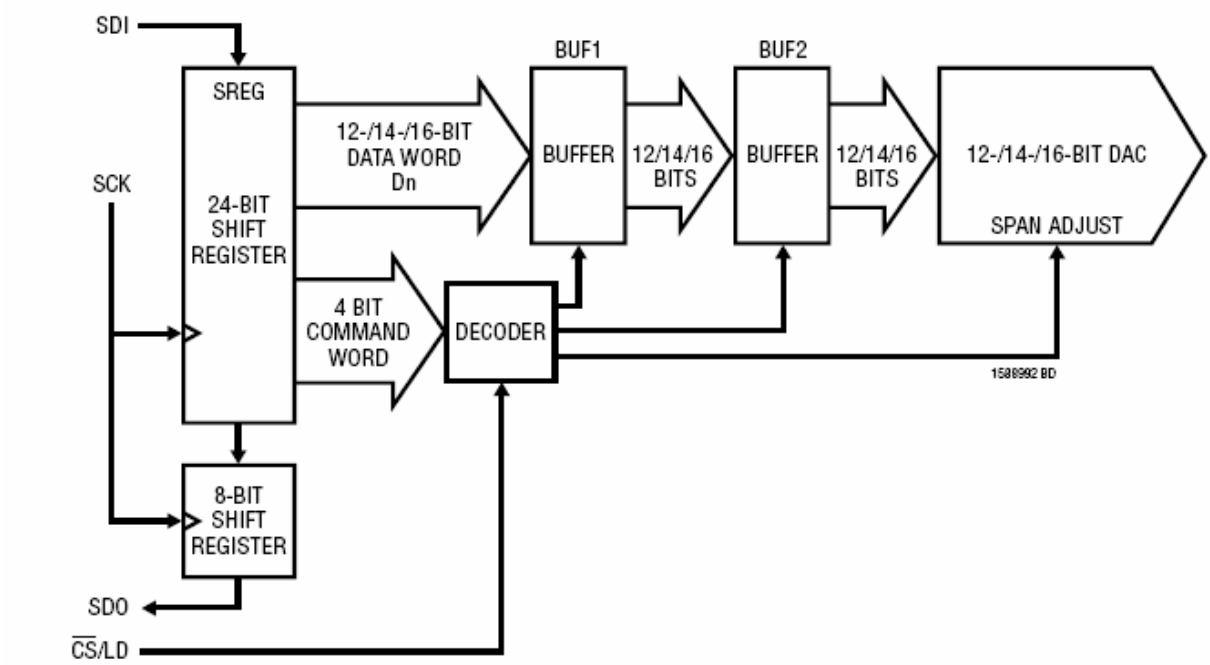

Figure 2.1: LTC1592 Block Diagram

COMMAND				OPERATION EACH COMMAND IS EXECUTED ON THE RISING EDGE OF CS/LD	Internal Register Status			
C3	C2	C1	C0		SREG DATA WORD Dn IN INPUT SHIFT REGISTER	BUF1 INPUT BUFFER	BUF2 DAC BUFFER (DAC OUTPUT)	DAC OUTPUT RANGE
0	0	0	0	Copy Dn in SReg to Buf1. Does not change range.	Dn	Dn	No Change	No Change
0	0	0	1	Copy the Data in Buf1 to Buf2	X	Dn	Dn	No Change
0	0	1	0	Copy Dn in SReg to Buf1 and Buf2 Does not change range.	Dn	Dn	Dn	No Change
0	0	1	1	Reserved (Do Not Use)				
0	1	0	0	Reserved (Do Not Use)				
0	1	0	1	Reserved (Do Not Use)				
0	1	1	0	Reserved (Do Not Use)				
0	1	1	1	Reserved (Do Not Use)				
1	0	0	0	Set Range to 0 to 5V. Copy Dn in SReg to Buf1 and Buf2	Dn	Dn	Dn	5V
1	0	0	1	Set Range to 0 to 10V. Copy Dn in SReg to Buf1 and Buf2	Dn	Dn	Dn	10V
1	0	1	0	Set Range to ±5V. Copy Dn in SReg to Buf1 and Buf2	Dn	Dn	Dn	±5V
1	0	1	1	Set Range to ±10V. Copy Dn in SReg to Buf1 and Buf2	Dn	Dn	Dn	±10V
1	1	0	0	Set Range to ±2.5V. Copy Dn in SReg to Buf1 and Buf2	Dn	Dn	Dn	±2.5V
1	1	0	1	Set Range to -2.5V to 7V. Copy Dn in SReg to Buf1 and Buf2	Dn	Dn	Dn	-2.5V to 7.5V
1	1	1	0	Reserved (Do Not Use)				
1	1	1	1	No Operation	X	No Change	No Change	No Change

Data Word Dn (n = 0 to 15) is the last 16 bits shifted into the input shift register SReg that corresponds to the DAC code.

Table 2.2: LTC1592 Commands

2.2.2 Immediate Mode Operation

Writing in a D/A data register will update the corresponding analog output. Along the data, the content of the command register will be sent to the D/A.

Since the command register contains the output range information, to allow for different ranges in different channels, either the command register will need to be updated between the data writes, or, once all the channels have been programmed with the proper range, the value 0x02 should be used in the command register.

The Command Register value 0x02 allows sending data without range information. It should always be used, in the Command Register, after the initial range programming is completed, particularly if the range selection is not the same among the channels.


```
uint16 *commandReg = (uint16 *)(SOFTDAC_IOSPACE + 0x48);
uint16 *DA_data = (uint16 *)(SOFTDAC_IOSPACE + 0x48);

void initOutput(uint16 range[], uint16 initialValue[])
// range is an array of 16 command word to set the range of each D/A
{
	int i;

	for (i=0; i<16; i++) {
		*commandReg = range [i];
		DA_data[i] = initialValue[i];	// send the initial
			// value and the range data to the D/A
			// the best initial value has to be determined
			// depending on the application
	}

	*commandReg = 0x2;	// value that will allow to change
			// the data without changing the range

}

void output_noTrigger(uint16 out_data[])
// out_data is an array of 16 values to be sent to the D/A
{
	int i;

	for (i=0; i<16; i++) DA_data[i] = out_data[i];
}
```


2.2.3 Trigger-Synchronized Operation

The board has the ability to input or output an external trigger for synchronized updates.

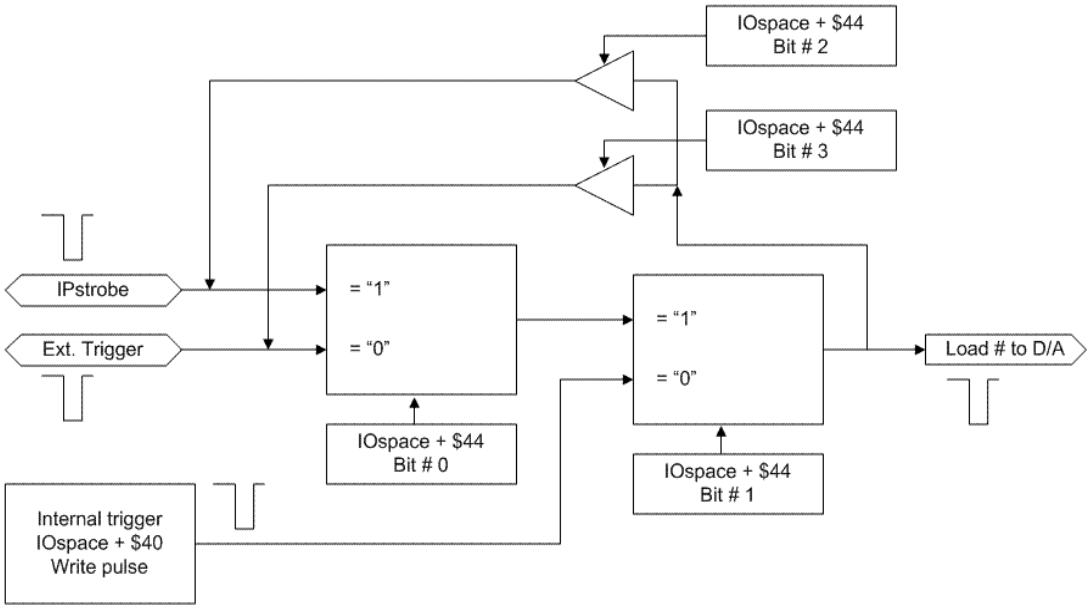

Figure 2.1: External Trigger Block Diagram

The Control Register is responsible for selecting the trigger source, and possible trigger output to synchronize other boards.

The source can come from the on-board logic. In this case, writing to the trigger register will generate the trigger signal. It can also come from the IP Strobe line or from the External Trigger pin on the IP I/O connector.

For system integrity reasons, the trigger is disabled if the Command Register does not contain 0x01

The trigger can be output on either the IP Strobe line or from the External Trigger pin on the IP I/O connector, depending on the control register programming

To use the trigger synchronized operation, follow this sequence:

- Set the Command Register to 0: this will allow programming the D/As while preventing the outputs from being updated.
- Write the updated data in each D/A data registers.
- Set the Command Register to 1: this will tell the D/A to update the outputs, however, this command will not be sent until the trigger is received.
- Generate the internal trigger by writing any value to the Trigger Register, **or** wait for the outside trigger.

Control Register Data	Trigger Input Source	Trigger Output Destination
0x00	Internal Trigger	No output
0x02	External Trigger from the Event line	No output
0x03	External Trigger from the Strobe line	No output
0x04	Internal Trigger	Strobe line
0x06	External Trigger from the Event line	Strobe line
0x08	Internal Trigger	Event line
0x0B	External Trigger from the Strobe line	Event line
0x0C	Internal Trigger	Event line, Strobe line
others	Reserved (Do Not Use)	Reserved (Do Not Use)

Upon trigger reception, the logic sends simultaneously a write to all the D/As. As in a normal write, the 4 command bits are coming from the command register. The 16 data bits are the same value as the latest write to the D/A register.

This trigger operation is intended to be used with 0x01 in the command register. Using that command, the data field of the 24-bit input word is ignored.

It should be noted that writing in the command register does not write to the D/A. The command register value is sent to the D/As in only 2 cases:

- One particular D/A when writing the D/A data register.
- All the D/As simultaneously when receiving a trigger, provided that the command register contains 0x01.


```
uint16 *commandReg = (uint16 *)(SOFTDAC_IOSPACE + 0x48);
uint16 *DA_data = (uint16 *)(SOFTDAC_IOSPACE + 0x48);
uint16 *DA_trigger = (uint16 *)(SOFTDAC_IOSPACE + 0x40);

void output_noTrigger(uint16 out_data[])
// out_data is an array of 16 values to be sent to the D/A
{
        int i;

        *commandReg = 0;    // prevents the D/A from
                            // updating the output

        for (i=0; i<16; i++) DA_data[i] = out_data[i];  // the data is set
                            // in the data latches but the DA will not
                            // update the outputs yet

        *commandReg = 1;    // set the command register to update
                            // the output with the data stored
                            // already stored in the D/A

        *DA_trigger = 0;        // send simultaneously a command word
                                // to all the D/A, this updates the output
                            // synchronously.  The data is the same
                            // as earlier but, anyway, not used since the
                            // command is '1'.

}
```

2.2.4 External 5 Volt Reference

The jumper area J2 allows selecting an external reference for the D/As.

When pins 1 and 2 are connected, the channels 0-7 use the internal 5-Volt reference, when pins 3 and 4 are connected; the channels 0-7 use the external reference.

When pins 5 and 6 are connected, the channels 8-15 use the internal 5-Volt reference, when pins 7 and 8 are connected; the channels 8-15 use the external reference.

3 APPENDIX A: OUTPUT CONNECTOR

Pin	Signal	Pin	Signal
1	D/A # 0	26	AGND
2	D/A # 1	27	AGND
3	D/A # 2	28	AGND
4	D/A # 3	29	AGND
5	D/A # 4	30	AGND
6	D/A # 5	31	AGND
7	D/A # 6	32	AGND
8	D/A # 7	33	AGND
9	D/A # 8	34	AGND
10	D/A # 9	35	AGND
11	D/A # 10	36	AGND
12	D/A # 11	37	AGND
13	D/A # 12	38	AGND
14	D/A # 13	39	AGND
15	D/A # 14	40	AGND
16	D/A # 15	41	AGND
17		42	
18		43	
19		44	
20		45	
21		46	
22		47	
23		48	
24	GND	49	AGND
25	Ext. Trigger	50	Ext. 5V Reference

