IP-48DAC-16

48-channel 16-bit
Digital/Analog Converter
With memory
Industry Pack Module

PROGRAMMING MANUAL

828-10-000-4000

Version 1.0

January 2007

ALPHI TECHNOLOGY CORPORATION
1898 E. Southern Ave
Tempe, AZ 85282 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-48DAC-16** is a high performance DIGITAL TO ANALOG module. The **IP-48DAC-16** outputs 16 channels with a 16-bit resolution at a maximum settling time of 2 µS.

The primary features of the **IP-48DAC-16** are as follows:

- 20 µSecond settling time (0 to 5 V)
- Bipolar Output Range: ±10V
- 1 LSB Max DNL and INL Over the Industrial Temperature Range
- Glitch Impulse < 20nV-s
- Power-On Reset to 0V
- Local 8kx8 Flash EPROM to store local user information
- Two stage buffers
- Global output buffer with timer based triggering
- 128Kbytes memory for waveform generation

1.2 FUNCTIONAL DESCRIPTION

The IP-48DAC-16 uses 3 Analog Devices AD5360 D/A converters.

The AD5360s are serial input 16-bit voltage output DACs.

The gain and offset of each group of 8 DACs can be independently adjusted by an offset DAC.

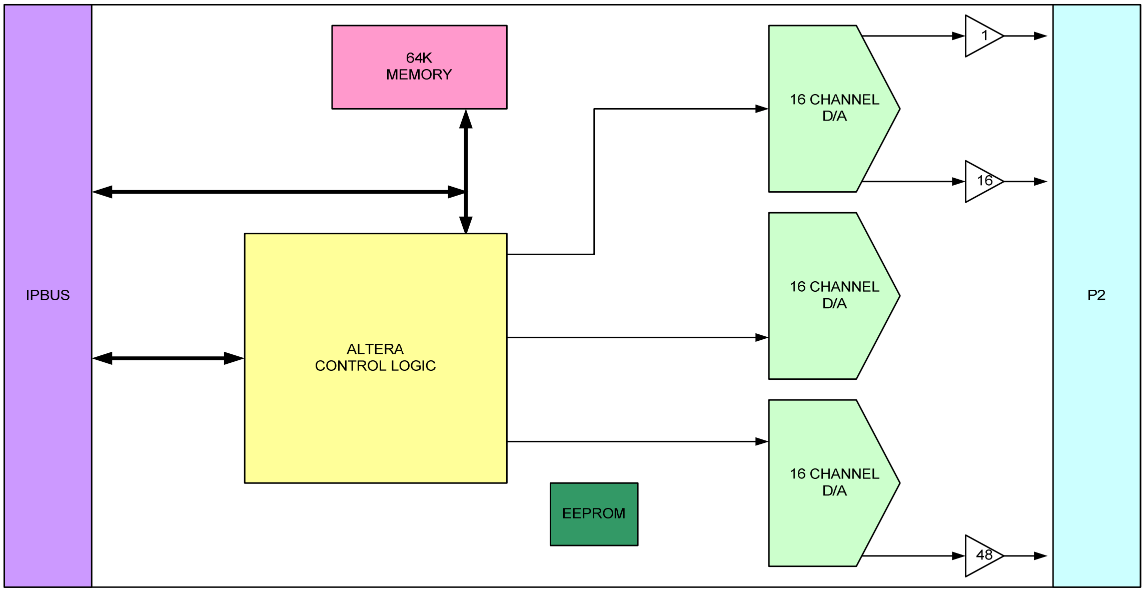

Figure 1.1: Block Diagram

2 IP INTERFACE

2.1 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	0x25
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	0x25
0x0D	Revision module	0x0A
0x0F	Reserved	

Table 2-2 IDSPACE content 8MHz IP

2.2 IOSPACE

The **IP-48DAC-16** module use 3 Analog Devices AD5360 D/A converter.

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

NAME	ADDR	DATA	R/W	COMMENTS
SampleClock	0x00	32-bit	R/W	Divisor for Internal Sampling Clock
CurrentAddr	0x04	16-bit	RO	Current Address for the State Machine
LastAddr	0x06	16-bit	R/W	Last Address with Valid Data, Bank 0
IntAddr	0x08	16-bit	R/W	Buffer address that will generate an interrupt.
DAC01readback	0x0a	16-bit	R/W	DAC #1 read back
DAC02readback	0x0c	16-bit	R/W	DAC #2 read back
DAC03readback	0x0e	16-bit	R/W	DAC #3 read back
DAC01	0x10	24-bit	R/W	DAC #1 Register Access
DAC02	0x14	24-bit	R/W	DAC #2 Register Access
DAC03	0x18	24-bit	R/W	DAC #3 Register Access
CtrlRegister	0x1c	16-bit	R/W	General control and status
DACoutput	0x20-0x7f	16-bit x 48	W	Direct D/A write

2.2.1 SampleClock (Read / Write 32 bits)

This register sets the sampling rate of the internal sampling rate generator. The internal sampling rate generator is based on a 32 MHz oscillator on the card. The sampling rate is set by the following formula where N is the contents of this register.

$$SamplingRate = \frac{32000000}{2 + N}$$

Since the maximum sampling rate supported by the DACs on the **IP-48DAC-16** is 50 KHz, the smallest value for *N* should be 620.

This register can be accessed in WORD

2.2.2 CurrentAddress (Read Only 16 bits)

This register reports the current buffer address of the state machine. It can be cleared by writing to the **ResetAddress** register.

The current buffer address is the next offset into the active bank, which will be written to the DACs on the next sample clock. Although the address counter is 24 bits, only the lowest 11 bits are significant at this time.

2.2.3 LastAddr (Read / Write 16 bits)

These registers contain the last valid data point address when the card is operated in state machine mode.

Although these registers and the address counter are 16 bits, only the lowest 11 bits are significant at this time.

2.2.4 CtrlRegister (Read / Write 16 bits)

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
AUTO UPDATE DAC	ENABLE STATE MACHINE	ENABLE INT SAMP CLOCK	READ BUSY (Read only)	S2COMP	INT_WHEN _MATCH		SM MODE

AUTO UPDATE DAC   ***Update outputs on manual DAC writes***

When this bit is set to 1, a manual update to a DAC register will also immediately update the output voltage. When this bit is cleared to 0, a manual write will only store the value in a holding register, and the output voltage can be updated by manually writing to the update address **UPDATE DACS**, or by the internal sampling clocks if enabled.

This bit is cleared to a 0 by a board RESET.

ENABLE STATE MACHINE   ***Enables automatic output from the RAM buffers***

When this bit is set to 1, the card will automatically reload the DAC registers from the output buffers at each sampling clock based on the current address and the active bank. When this bit is cleared to 0, no automatic updates will occur.

This bit is cleared to a 0 by a board RESET.

ENABLE INT SAMP CLOCK

This bit determines the source of the sampling clock as demonstrated in the following table.

ENABLE INT SAMP CLOCK	SOURCE
1	Internal sampling clock generator
0	Writes to UPDATE DACS generate sampling clock

Table 2.5: Sampling Clock Options

These bits are cleared to a 0 by a board RESET.

INT WHEN MATCH   ***Interrupt HOST when address match***

When this bit is set to 1 and this state machine address counter matches the value in the IntAddr register, the HOST is interrupted. When this bit is cleared to 0, no interrupt is generated. This bit is cleared to 0 by a board RESET.

READ BUSY

When this bit is set to 1 a register read operation is taking place and the data for the read operation is not yet valid.

S2COMP

When this bit is set to 0, the D/A uses unsigned binary for the value (0x0000 is -10V, 0xffff is +10V), when it is set to 1, it uses 2's complement (0x8000 is -10 V, 0x7fff is +10V).

SM MODE ***What to do when bank is done***

When this bit is set to 1, when the state machine reaches the end of its buffer, it stops.

When the bit is set to 0, the address counter resets to 0 and the state machine keeps outputting values looping through the memory.

BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 09	BIT 08
RESET SAMPLE CLK	RESET ADDRESS	RESET DACS	UPDATE DACS		BUSY 2 (Read only)	BUSY 1 (Read only)	BUSY 0 (Read only)

RESET SAMPLE CLK

Writing to this location will force the internal sampling clock generator to reload the count from the **INT SAMP CLK** register.

RESET ADDRESS

Writing to this location will clear the state machine **CurrentAddress** counter.

NOTE: If this is written while the state machine is active, there will be a phase jump in the output, and additionally, if the switch occurs in the middle of an update, then some outputs may be updated from the old location, and some from the new location for one sample clock.

RESET DACS

Writing a 1 to this location will clear DAC holding registers, and force the DACs to output 0 volts.

UPDATE DACS

Writing a 1 to this location will generate a manual sample clock pulse. It can be used in a full manual mode to update all DAC outputs simultaneously.

2.2.5 DACs Register Access (Read/Write)

Writing to this location will send the written data to the corresponding DAC. Upon reading, a '0' in the upper bit indicates that the DAC is busy.

2.3 MEM Space

2.3.1 Waveform SRAM storage

The IP48DAC contains a 128Kbytes RAM used by the state machine to provide automatic DAC updates. The memory is accessible from 0x00000 to 0x1ffff in the IP MEM space.

The memory is logically split in 64 banks. 1024 16-bit locations available for each DAC channel in the corresponding bank. The on-board logic provides the arbitration to allow the host to write in the memory while it is used by the state machine.

2.3.2 Flash

The IP48DAC board contains an AT28C64 8K by 8 Flash EPROM available to the user to stored information related to the module offset gain error for eventual software correction. The memory is located at the address 0x20000 to 0x23ffe on the even bytes.

The AT28C64 is a low-power, high-performance 8,192 words by 8-bit nonvolatile electrically erasable and programmable read only memory with popular, easy-to-use features.

The AT28C64 is accessed like a Static RAM for the read or write cycles without the need for external components. During a byte write, the address and data are latched internally, freeing the microprocessor address and data bus for other operations. Following the initiation of a write cycle, the device will go to a busy state and automatically clear and write the latched data using an internal control timer.

The AT28C64E offers a byte write time of 200 µs maximum. This feature allows the entire device to be rewritten in 1.6 seconds.

The IP48DAC implementation uses DATA Polling of I/O_7 to detect the end of a write cycle. Once the end of a write cycle has been detected, a new access for a read or write can begin. During a write cycle, an attempted read of the data being written results in the complement of that data for I/O_7 (the other outputs are indeterminate). When the write cycle is finished, true data appears on all outputs.

2.4 MODES OF OPERATION

The card can be viewed as operating in one of the following modes.

- State Machine providing Automatic Update and Load on Sampling Clock
- Manual Load with Update on Sampling Clock
- Manual Load and Update

2.4.1 State Machine providing Automatic Update and Load on Sampling Clock

The card contains a state machine capable of automatically loading the DAC holding registers from the RAM buffers on each sample clock.

On each sampling clock, the DACs are updated from the holding registers, and then 48 values are read from the active buffer bank into the holding registers for the next sampling clock.

On the first sampling clock after the state machine is enabled, the DAC holding registers will contain zero if the DACs were RESET. The first data point will be output on the second sampling clock. When the state machine is disabled at the end of a bank, the actual last point is output one sampling clock later.

2.4.2 Manual Load with Update on Sampling Clock

In this mode, the card will transfer the values in the DAC holding registers to the output on each internal sampling clock. A HOST interrupt is used to have the HOST load the next set of data to the DAC holding registers.

Obviously, this mode will make much greater demands of the HOST as it will be interrupted at every sample clock. Sample rates above a few kHz might not be possible due to the needs of the interrupt routine.

2.4.3 Manual Load and Update

This is a purely manual mode of operation, without making use of any timing on the part of the card. The HOST can write the desired values to the DAC holding registers and then write to **UPDATE DACS** to update all 48 outputs at the same time.

Alternatively, if the ***AUTO UPDATE DAC*** bit is set, the DACs will update the output voltage at the same time that the holding register is written.

2.5 ANALOG OUTPUT

The **IP-48DAC-16** has sixteen analog outputs each with its own buffer.

The output ranges are programmable using the board control register.

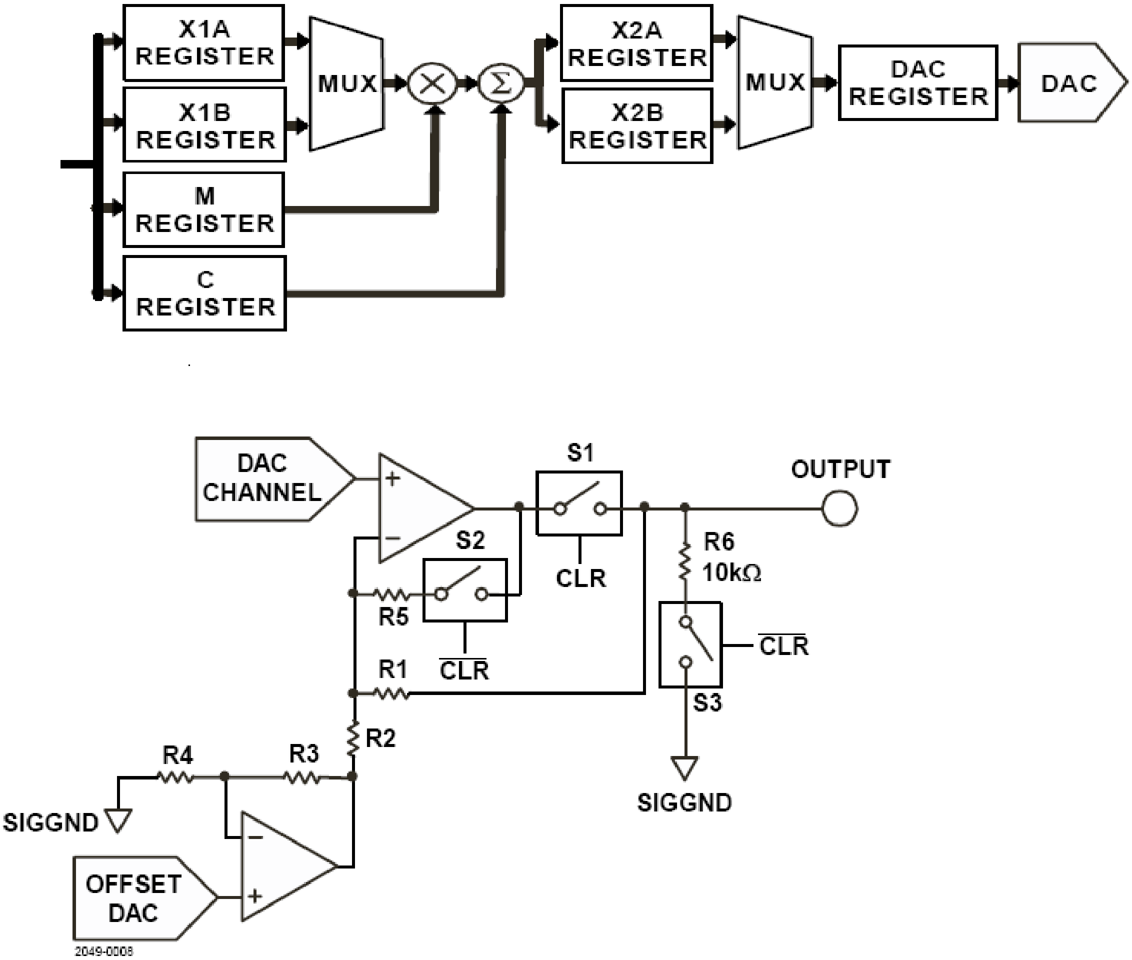

Figure 2.1: AD5360 Block Diagram

2.5.1 AD5360 Command Structure

The AD5360 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.

2.5.2 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.

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

2.5.3 Thermal Monitoring Function

The AD5360/AD5361 can be programmed to power down the DACs if the temperature on the die exceeds 130°C. Setting Bit 1 in the control register (see Table 14) will enable this function. If the die temperature exceeds 130°C the AD5360/AD5361 will enter a temperature power-down mode, which is equivalent to setting the power-down bit in the control register. To indicate that the AD5360/AD5361 has entered temperature shutdown mode Bit 4 of the control register is set. The AD5360/AD5361 will remain in temperature shutdown mode, even if the die temperature falls, until Bit 1 in the control register is cleared.

2.5.4 Toggle Mode

The AD5360/AD5361 has two X2 registers per channel, X2A and X2B, which can be used to switch the DAC output between two levels with ease. This approach greatly reduces the overhead required by a micro-processor which would otherwise have to write to each channel individually. When the user writes to either the X1A, X2A, M or C registers the calculation engine will take a certain amount of time to calculate the appropriate X2A or X2B values. If the application only requires that the DAC output switch between two levels, such as a data generator, any method which reduces the amount of calculation time encountered is advantageous. For the data generator example the user need only set the high and low levels for each channel once, by writing to the X1A and X1B registers. The values of

X2A and X2B will be calculated and stored in their respective registers. The calculation delay therefore only happens during the setup phase, i.e. when programming the initial values. To toggle a DAC output between the two levels it is only required to write to the relevant A/B Select Register to set the MUX2 register bit. Furthermore, since there are 8 MUX2 control bits per register it is possible to update eight channels with a single write. Table 16 shows the bits that correspond to each DAC output.

Table 9. AD5360 Serial Word Bit Assignation

I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
M1	M0	A5	A4	A3	A2	A1	A0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

M1 and M0 are mode bits.

A5 is an unused address bit and must always be written as 0.

A4 to A0 are address bits.
D15 to D0 are data bits.
*In the AD5361, bits I1 and I0 only used in Special Function Mode

2.5.5 SPI Readback MODE

The AD5360/AD5361 allows data read-back from every register directly accessible to the serial interface, which is all registers except the X2A, X2B and DAC registers. In order to read back a register, it is first necessary to tell the AD5360/AD5361 which register is to be read. This is achieved by writing to the device a word whose first two bits are the special function code 00. The remaining bits then determine if the operation is a read-back, and the register which is to be read back, or if it is a write to of the special function registers such as the control register.

After the special function write has been performed, if it is a readback command then data from the selected register will be clocked out of the SDO pin during the next SPI operation.

2.5.6 Channel Addressing and Special Modes

If the mode bits are not 00, then the data word D15 to D0 is written to the device. Address bits A4 to A0 determine which channel or channels is/are written to, while the mode bits determine to which register (X1A, X1B, C or M) the data is written, as shown in Table 9 or Table 10. If data is to be written to the X1A or X1B register, the setting of the A/B bit in the Control Register determines which register is used (0 -> X1A, 1 -> X1B).

Table 11. Mode Bits

M1	M0	Action
1	1	Write DAC input data (X1A or X1B) register, depending on Control Register A/B bit.
1	0	Write DAC offset (C) register
0	1	Write DAC gain (M) register
0	0	Special function, used in combination with other bits of word

The AD5360/AD5361 has very flexible addressing that allows writing of data to a single channel, all channels in a group, the same channel in groups 0 and 1, or all channels in the device. Table 12 shows all these address modes.

Table 12. Group and Channel Addressing

This table shows which group(s) and which channel(s) is/are addressed for every combination of address bits A4 to A0.

		ADDRESS BITS A4 TO A3			
		00	01	10	11
ADDRESS BITS A2 TO A0	000	All groups, all channels	Group 0, channel 0	Group 1, channel 0	Unused
	001	Group 0, all channels	Group 0, channel 1	Group 1, channel 1	Unused
	010	Group 1, all channels	Group 0, channel 2	Group 1, channel 2	Unused
	011	Unused	Group 0, channel 3	Group 1, channel 3	Unused
	100	Unused	Group 0, channel 4	Group 1, channel 4	Unused
	101	Unused	Group 0, channel 5	Group 1, channel 5	Unused
	110	Unused	Group 0, channel 6	Group 1, channel 6	Unused
	111	Unused	Group 0, channel 7	Group 1, channel 7	Unused

SPECIAL FUNCTION MODE

If the mode bits are 00, then the special function mode is selected, as shown in Table 13. Bits I21 to I16 of the serial data word select the special function, while the remaining bits are data required for execution of the special function, for example the channel address for data read-back. The codes for the special functions are shown in Table 14. Table 15 shows the addresses for data read-back.

Table 13. Special Function Mode

I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
0	**0**	S5	S4	S3	S2	S1	S0	F15	F14	F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0

Table 14. Special Function Codes

SPECIAL FUNCTION CODE						DATA	ACTION
S5	S4	S3	S2	S1	S0	F15-F0	
0	0	0	0	0	0	0000 0000 0000 0000	NOP
0	0	0	0	0	1	XXXX XXXX XXX[F4:F0]	Write control register F4 = 1 -> Over-temperature; F4 = 0 -> Temp OK (Read-only bit) F3 = 1 -> PEC error; F3 = 0 -> PEC OK (Read-only bit) F2 = 1 -> Select B register for input; F2 = 0 -> Select A register for input F1 = 1 -> Enable temperature shutdown; F1 = 0 -> Disable temperature shutdown F0 = 1 -> Soft power down; F0 = 0 -> Soft power up
0	0	0	0	1	0	[F13:F0]	Write data in F13:F0 to OFS0 register
0	0	0	0	1	1	[F13:F0]	Write data in F13:F0 to OFS1 register
0	0	0	1	0	1	See Table 15	Select register for readback
0	0	0	1	1	0	XXXX XXXX[F7:F0]	Write data in F7:F0 to A/B Select Register 0
0	0	0	1	1	1	XXXX XXXX[F7:F0]	Write data in F7:F0 to A/B Select Register 1
0	0	1	1	0	0	XXXX XXXX XX[F5:F0]	Monitor configure and write F5 = 1 -> Monitor enable; F5 = 0 -> Monitor disable F4 = 1 -> Monitor input pin selected by I0 (0 = MON_IN0, 1 = MON_IN1) F4 = 0 -> Monitor DAC channel selected by F3:F0 (0000 = channel 0 -> 1111 = channel 15)
0	0	1	1	0	1	XXXX XXXX XXXX XX[F1:F0]	GPIO configure and write F1 = 1 -> GPIO is output. Data to output is written to F0 F1 = 0 -> GPIO is input. Data can be read from F0 on read-back

Note. When writing to the offset registers, the 14-bit data is right justified (bits F15 and F14 are don't care). When writing to the X, M or C registers of the AD5361, the 14-bit data is left-justified (bits 1 and 0 of the data word are don't care).

Table 15. Address Codes for Data Readback

F15	F14	F13	F12	F11	F10	F9	F8	F7	REGISTER READ[1]
0	0	0	Bits F12 to F7 select channel to be read back, from Channel 0 = 001000 to Channel 15 = 010111						X1A Register
0	0	1							X1B Register
0	1	0							C Register
0	1	1							M Register
1	0	0	0	0	0	0	0	1	Control Register
1	0	0	0	0	0	0	1	0	OFS0 Data Register
1	0	0	0	0	0	0	1	1	OFS1 Data Register
1	0	0	0	0	0	1	1	0	A/B Select Register 0
1	0	0	0	0	0	1	1	1	A/B Select Register 1
1	0	0	0	0	1	0	1	1	GPIO read (data in F0)[2]

[1] F6 to F0 are don't care for data readback functions except for GPIO read.

[2] F6 to F0 should be 0 for GPIO read

Table 16. DACs Select by A/B Select Registers

A/B Select Register	Bits		
	F7	F6	F5