ATC-IP-AD3216

32 (2x 16) Channel multiplexed A/D 16 bits Industry Pack Module

REFERENCE MANUAL

790-11-000-4000
Version 1.4
March 2003

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 **ATC-AD3216** is a 16 bit single width IP module designed for high speed burst A/D data acquisition in 16 bits. The primary features of the **ATC-AD3216** are:

- Two groups of 16 multiplexed channels using 2 A/D simultaneous with 16 bit A/D acquisitions, operating at a maximum rate of 100 kHz per A/D converter.
- Up to 16K of samples stored and divided among active A/D converters in onboard FIFO memory.
- Full support for pre-trigger and post trigger acquisition.
- Sampling clock selected from one of the following sources: Internal divider (IPCLK / N) and IPSTROBE.
- Trigger event selected from one of the following sources: Write to IP register, IPSTROBE, and external trigger.
- Continuous streaming acquisition is also possible at lower throughput.
- IP bus operates at 32 MHz.
- Input signal selectable from 0 – 10v, +/-10v, +/-5v, 0-+5v, +/-.3.3v, 0-+4v.

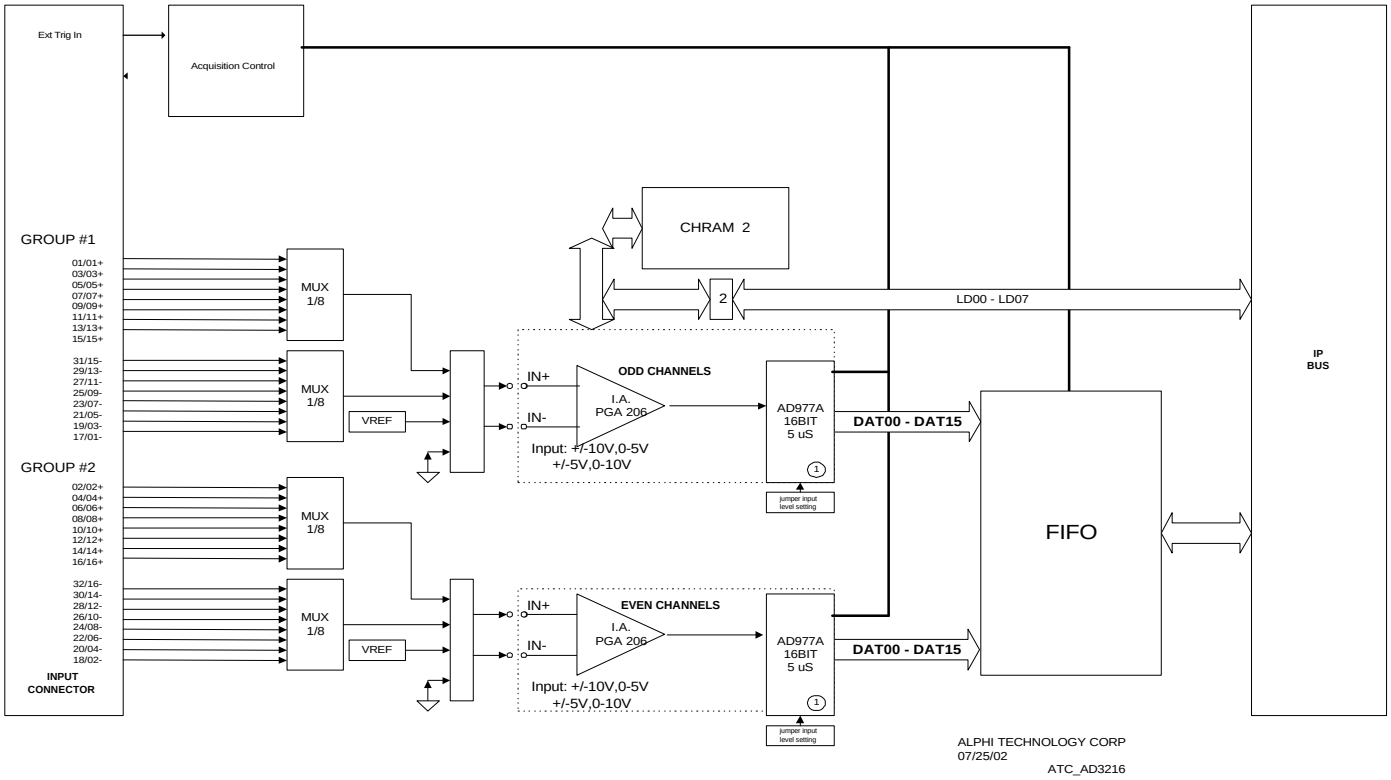


Figure 1.1 : Data Flow Block Diagram

1.2 FUNCTIONAL DESCRIPTION

A data flow block diagram of the **ATC-AD3216** is presented in Figure 1-1.

The ATC-AD3216 has four CMOS analog multiplexer that are fault protected. Each multiplexer has 8 inputs and one common output. These outputs are acquired by a four input differential multiplexer. The differential inputs can then be configured for single-ended, differential modes, calibration modes. These outputs then go to a PGA where the gain can be set for 1, 2, 4, and 8

When the acquisition is started the two A/D converters acquire the data simultaneously from the multiplexers. The data is stored in a 16-bit FIFO in an interleave matter reading group #1 (odd channel) first then group #2 (even channel). Further acquisitions before the trigger will result in the earliest data being discarded, thus maintaining the most recent data in the FIFO.

Once the trigger event is seen, no more data is discarded from the FIFO, and the acquisition proceeds until the FIFO is full. At this point, the acquisition stops, and the HOST can read the data from the IP. Interrupts can be sent to the HOST at Event and at acquisition finished.

If desired, such as for a continuous acquisition, data can be read from the FIFO while acquisition is in progress, for continuous streaming

1.2.1 SOFTWARE SUPPORT

The **ATC-AD3216** is supported under *Windows NT* / 2000 by two sample programs, which are supplied with the IP in the board support package. Both examples are designed to work with an IP-type carrier from ALPHI, such as the PCI-4IPM.

One sample program, called SnapShot, fully exercises the IP module in pre- and post-trigger modes, and displays the data to the screen. Data can be stored to a file and can be reloaded in the program at a later time.

The second program, called DrawIpAdc, operates the IP in continuous mode, and displays the data to the screen.

Full source to both the DSP code and the applications are provided.

2. THEORY OF OPERATION

2.1 ANALOG INPUTS

There are two groups of 16 channel analog inputs that are multiplexed using two eight channel multiplexers per A/D.

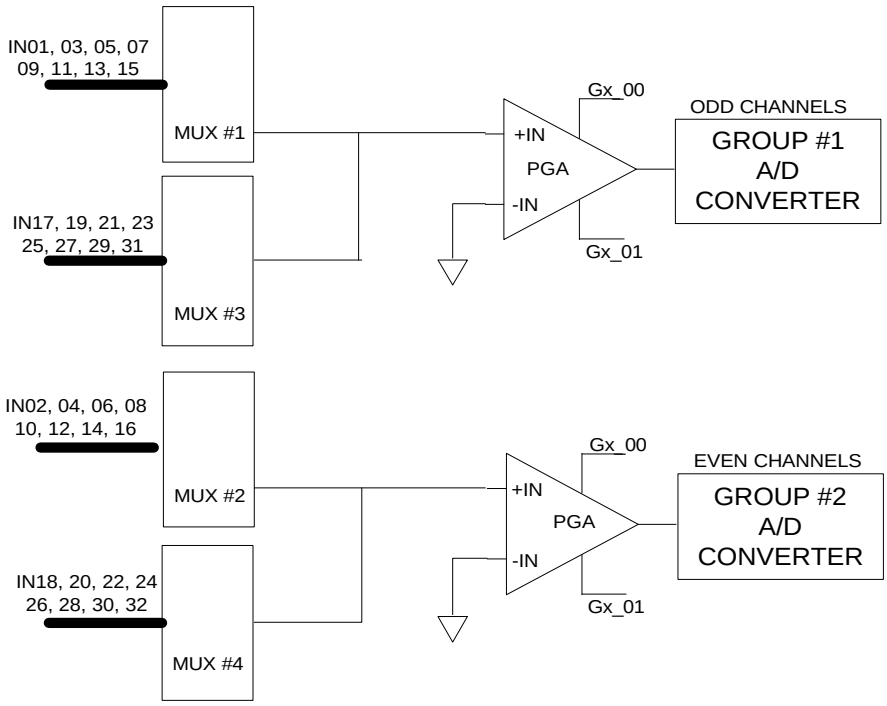
Each multiplexers output goes respectively to a switch that determines whether the data is in single-ended, differential or calibration mode. That data then goes to a instrument amplifier which gain is programmed on the fly. This configuration allows the user to have a mix of single ended or differential inputs selected on the fly. There is one 200KSPS A/D converter by group of 16 channels (if singled ended) or 8 channels (if programmed differential). A look-up table memory is used for this purpose.

Also, it is possible to inject a reference signal at the level of the switch upstream of the PGA amplifier and A/D converter to get a data reference.

Inputs signal scan can be selected by a range of jumpers from 0 – +10v, +/-10v, +/-5v, 0 - +5v, +/-.3.33v, 0-+4v.

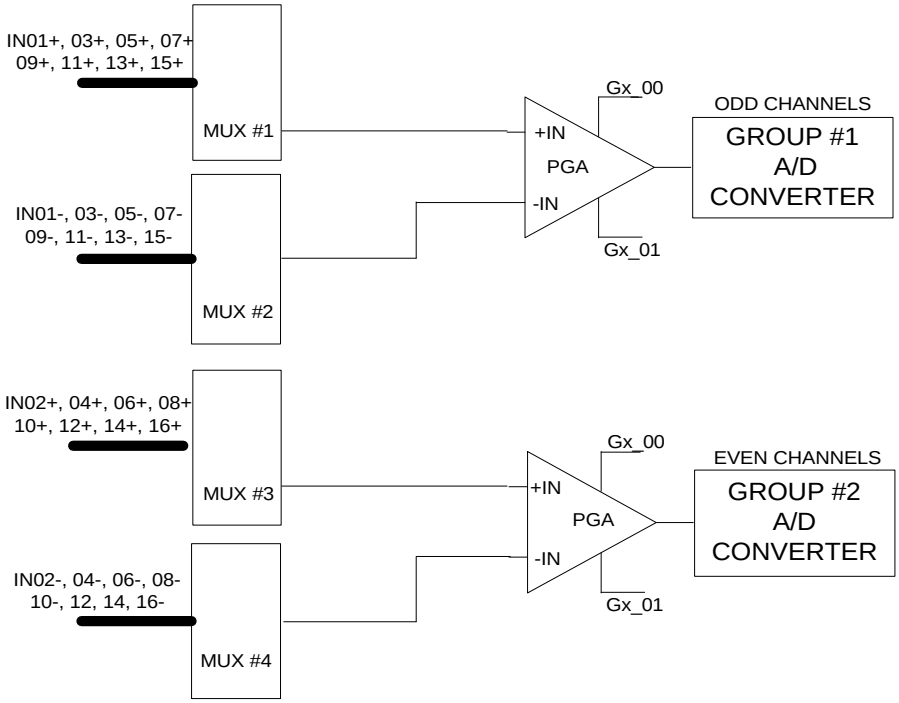
Below is a simple block diagram to show how the data is processed from the input to the A/D conversion in single-ended and differential modes. The FIFO takes the data in an interleaving matter starting with group #1 then group #2. When the inputs are acquired by the FIFO you will see IN01 (mux #1) then IN02 (mux #2) then back to IN03 (mux #1) then IN04 (mux # 2) and so on. Once the first 16 inputs are acquired by the A/D then the acquisition will switch to mux #3 & #4.

2.1.1 SINGLE-ENDED MODE



2.1.2 DIFFERENTIAL MODE

The FIFO takes the data in an interleaving matter starting with group #1 A/D converter then group #2 A/D converter. When the inputs are acquired by the FIFO you will see first IN01+/IN01- (mux #1 and #2) then IN02+ / IN02- (mux #3 and #4) then back to IN03+/IN03- (mux #1 and #2) then IN04+/IN04- (mux #3 and #4) and so on.



2.1.3 INPUT LEVEL SELECTION

The A/D converter can be selected for to different levels of input signal. Each set of jumpers are used for this purpose. Once a selection is made, it applies to all the inputs related to that A/D.

Input level Ch#1-16	JW5	JW6
+/-10v	3-4,6-8,9-10	1-2
+/-5v	2-4,7-8,10-12	1-2
+/-3.33v	1-3,7-8,9-10	1-2
0-+10v	2-4,7-8,13-14,10-12	
0-+5v	2-4,6-8,15-16,10-12	
0-+4v	3-4,6-8,15-16,9-10	
Input level Ch#17-32	**JW4**	**JW7**
+/-10v	3-4,6-8, 9-10	1-2
+/-5v	2-4,7-8,10-12	1-2
+/-3.33v	1-3,7-8,9-10	1-2
0-+10v	2-4,7-8,13-14,10-12	
0-+5v	2-4,6-8,15-16,10-12	
0-+4v	3-4,6-8,15-16,9-10	

2.1.4 CODE FORMAT

Jumper JW3 allows selection of code format for A/D data.

JW3	Description
No jumper	Straight binary format
1-2	Two's complement format

Jumper JW8 connects external ground from input to the A/D converters.

JW8	Description
No jumper	On board ground
1-2	External Input ground

2.2 CHANNEL RAM (READ / WRITE) BASE ADDRESS OF MEMORY

This 8 bit register accessed through memory base address $0 – $F (for 16 channels) allows for the setup of calibration, gain and differential / single ended mode selection on a per channel basis. The address $0 – $F reflect all 32 channels, 1 –16 for both A/D. See the table below for the mapping of each bit per channel.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
CAL1	DIFF2	G2_01	G2_00	CAL0	DIFF1	G1_01	G1_00

Group #2 A/D Even Channel ***Group #1 A/D Odd Channel***

Gx_00, Gx_01 Gain Selection

Input gains are selected on a per channel basis according to the following table.

Gx_01	Gx_00	Gain Selected
0	0	1
0	1	2
1	0	4
1	1	8

Table 2.1.3.3: Gain Selection

DIFF2, DIFF1 Single-Ended/Differential mode selection

These bits select whether the channels are in single ended or differential mode. If the bit is set to 0, then the channels are in single-ended mode and when the bits are set to 1 the channels are in differential mode. Note that a channel will be scanned twice in differential mode when all 16 channels are selected, example when in differential mode input to channel #1 (pin 1) will scan Ch #1(pin + input) and Ch#9. See the following table for channel assignment.

CH-Ram Address	Group #1 DIFF1 INPUT	DIFF1 = 0 SingleEnd	CH-Ram Address	Group #2 DIFF2 INPUT	DIFF2 = 0 SingleEnd
$0	IN01	CH 01 SE	$0	IN02	CH 02 SE
$1	IN03	CH 03 SE	$1	IN04	CH 04 SE
$2	IN05	CH 05 SE	$2	IN06	CH 06 SE
$3	IN07	CH 07 SE	$3	IN08	CH 08 SE
$4	IN09	CH 09 SE	$4	IN10	CH 10 SE
$5	IN10	CH 11 SE	$5	IN12	CH 12 SE
$6	IN13	CH 13 SE	$6	IN14	CH 14 SE
$7	IN15	CH 15 SE	$7	IN16	CH 16 SE
$8	IN17	CH 17 SE	$8	IN18	CH 18 SE
$9	IN19	CH 19 SE	$9	IN20	CH 20 SE
$A	IN21	CH 21 SE	$A	IN22	CH 22 SE
$B	IN23	CH 23 SE	$B	IN24	CH 24 SE
$C	IN25	CH 25 SE	$C	IN26	CH 26 SE
$D	IN27	CH 27 SE	$D	IN28	CH 28 SE
$E	IN29	CH 29 SE	$E	IN30	CH 30 SE
$F	IN31	CH 31 SE	$F	IN32	CH 32 SE

Table 2.1.3.2: Single-Ended Input channel selections.

Group #1 DIFF1 INPUT	DIFF2 = 1 Differential	Group #2 DIFF2 INPUT	DIFF1 = 1 Differential
IN01/IN01+ IN17/IN01-	CH IN01+/IN01-DE	IN02/IN02+ IN18/IN02-	CH IN02+/IN02-DE
IN03/IN03+ IN19/IN03-	CH IN03+/IN03-DE	IN04/IN04+ IN20/IN04-	CH IN04+/IN04-DE
IN05/IN05+ IN21/IN05-	CH IN05+/IN05-DE	IN06/IN06+ IN22/IN06-	CH IN06+/IN06-DE
IN07/IN07+ IN23/IN07-	CH IN07+/IN07-DE	IN08/IN08+ IN24/IN08-	CH IN08+/IN08-DE
IN09/IN09+ IN25/IN09-	CH IN09+/IN09-DE	IN10/IN10+ IN26/IN10-	CH IN10+/IN10-DE
IN11/IN11+ IN27/IN11-	CH IN11+/IN11-DE	IN12/IN12+ IN28/IN12-	CH IN12+/IN12-DE
IN13/IN13+ IN29/IN13-	CH IN13+/IN13-DE	IN14/IN14+ IN30/IN14-	CH IN14+/IN14-DE
IN15/IN15+ IN31/IN15-	CH IN15+/IN15-DE	IN16/IN16+ IN32/IN16-	CH IN16+/IN16-DE

Table 2.1.3.3: Differential Input channel selections.

CAL1, CAL0 Calibration source

When calibrate mode is selected, these bits will switch the calibration MUX to one of the four calibration sources.

CAL1	CAL0	Source for A/D #1	Source for A/D #2
0	0	GND	GND
0	1	VREF (+5 V)	VREF (+5 V)
1	0	IN01/IN01+	IN17/IN09+
1	1	IN09/IN01-	IN25/IN09-

Table 2.1.3.1: Calibration Sources

CAL_MODE (Write only) I/O space $0B

This I/O space register enables one channel mode and calibration mode. See FIFO Control / Status register on page 16 for more information.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
N/A	N/A	N/A	N/A	N/A	CAL MODE	ONE Channel	N/A

ONE CHANNEL (Read/Write)

When this bit is set to 1, the output multiplexers are forced to the channel specified in the **LAST CHANNEL** register. Only a single conversion will be triggered for each trigger signal. If this bit is cleared to 0 (default upon reset) then the state machine functions normally, scanning channels 0 to the channel specified in the **LAST CHANNEL** register. Upon reset the last channel register is set to $10, which scan all channels.

LAST CHANNEL (Read-Write) I/O space $22

Access to this register is at I/O space address $22. The purpose of this register is to specify the number channels the A/D will acquire. This register is set to acquire all 16 channels per group (32 channel Single-Ended / 16 differential) upon reset. To change the number of channels acquired by the A/D write the amount of channels to address $22 following the table below. The channel number chosen will enable all the channels up to that number in the table. In differential mode a channel programmed as differential will be acquired twice. See example below for further explanation.

Important note: When using one channel mode the last channel register only acquires data from the channel programmed into the last channel register. The last channel register chart is different. Look for the One channel charts for single ended and differential below to decode properly.

LAST CHANNEL	Single Ended	LAST CHANNEL	Single Ended
$1	IN01/IN02	**$9**	IN17/IN18
$2	IN03/IN04	**$A**	IN19/IN20
$3	IN05IN06	**$B**	IN21/IN22
$4	IN07IN08	**$C**	IN23/IN24
$5	IN09/IN10	**$D**	IN25/IN26
$6	IN11/IN12	**$E**	IN27/IN28
$7	IN13/IN14	**$F**	IN29/IN30
$8	IN15/IN16	**$10**	IN31/IN32

Last Channel Single-Ended

LAST CHANNEL	Single Ended	LAST CHANNEL	Single Ended
$0	IN01/IN02	**$8**	IN17/IN18
$1	IN03/IN04	**$9**	IN19/IN20
$2	IN05IN06	**$A**	IN21/IN22
$3	IN07IN08	**$B**	IN23/IN24
$4	IN09/IN10	**$C**	IN25/IN26
$5	IN11/IN12	**$D**	IN27/IN28
$6	IN13/IN14	**$E**	IN29/IN30
$7	IN15/IN16	**$F**	IN31/IN32

Last Channel Register Single-ended with One Channel set

LAST CHAN	Differential Mode	Differential Mode
$1	IN01/IN01+ IN17/IN01-	IN02/IN02+ IN18/IN02-
$2	IN03/IN03+ IN19/IN02-	IN04/IN04+ IN20/IN04-
$3	IN05/IN05+ IN21/IN05-	IN06/IN06+ IN22/IN06-
$4	IN07/IN07+ IN23/IN07-	IN08/IN08+ IN24/IN08-
$5	IN09/IN09+ IN25/IN09-	IN10/IN10+ IN26/IN10-
$6	IN11/IN11+ IN27/IN11-	IN12/IN12+ IN28/IN12-
$7	IN13/IN13+ IN29/IN13-	IN14/IN14+ IN30/IN14-
$8	IN15/IN15+ IN31/IN15-	IN16/IN16+ IN32/IN16-

All Last Channel Differential Mode

LAST CHAN	Differential Mode	Differential Mode
$0	IN01/IN01+ IN17/IN01-	IN02/IN02+ IN18/IN02-
$1	IN03/IN03+ IN19/IN03-	IN04/IN04+ IN20/IN04-
$2	IN05/IN05+ IN21/IN05-	IN06/IN06+ IN22/IN06-
$3	IN07/IN07+ IN23/IN07-	IN08/IN08+ IN24/IN08-
$4	IN09/IN09+ IN25/IN09-	IN10/IN10+ IN26/IN10-
$5	IN11/IN11+ IN27/IN11-	IN12/IN12+ IN28/IN12-
$6	IN13/IN13+ IN29/IN13-	IN14/IN14+ IN30/IN14-
$7	IN15/IN15+ IN31/IN15-	IN16/IN16+ IN32/IN16-

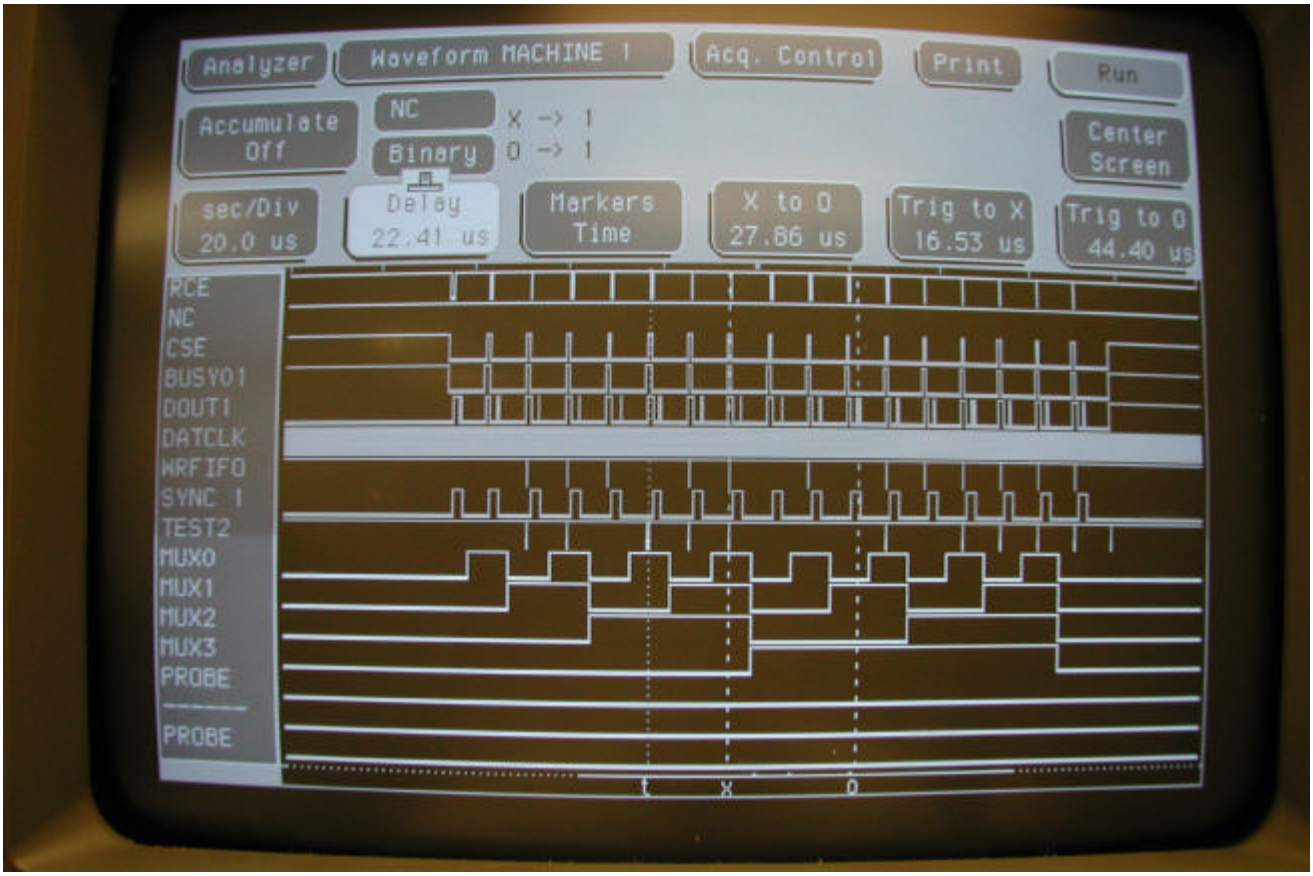
Last Channel Differential Mode with One Channel set

Example: With Last channel programmed as $10 (all channels), program IN01 in channel ram to be differential. Write to group #1 odd channel at address $00 = $04 and at address $08 = $04. Next program channel ram channel #12 to be differential, write to group #2 even channel at address $05 = $40 and at address $0D = $40. Notice that both the plus and minus inputs of a channel need to be programmed as differential in the channel ram. All other addresses in channel ram are filled with $00 for single-ended. Chart below will show the results of programming. Notice the differential channels are scanned twice.

CH-Ram Address Values	Group #1 (ODD) A/D Active Inputs	Group #2 (EVEN) A/D Active Inputs
$0 = 04	IN01/IN01+ IN17/IN01-	IN02/IN02+
$1 = 00	IN03/IN03+	IN04/IN04+
$2 = 00	IN05/IN05+	IN06/IN06+
$3 = 00	IN07/IN07+	IN08/IN08+
$4 = 00	IN09/IN09+	IN10/IN10+
$5 = 40	IN11/IN11+	IN12/IN12+ IN28/IN12-
$6 = 00	IN13/IN13+	IN14/IN14+
$7 = 00	IN15/IN15+	IN16IN16+
$8 = 04	IN01/IN01+ IN17/IN01-	IN18/IN18-
$9 = 00	IN19/IN05+	IN20/IN07-
$A = 00	IN21/IN09-	IN22/IN11-
$B = 00	IN23/IN13-	IN24/IN15-
$C = 00	IN25/IN02-	IN26/IN04-
$D = 40	IN27/IN06-	IN12/IN12+ IN24/IN12-
$E = 00	IN29/IN10-	IN30/IN14-
$F = 00	IN31/IN14-	IN32/IN16-

Last channel programming example

Here is an example in time showing the A/D access with the last channel programmed with the value $10, this represents all 16 channels to one of the A/D or 32 channels to both A/D



CAL_MODE (Write only)

When this bit is set to 1, the input multiplexers are switched to the calibration source. This includes the sources in Table 2.1.3.1

2.3 A/D CONVERTERS

There are two A/D converters. The A/D converters operate continuously at the selected sampling rate. The A/D are 200 KSPS A/D converter @ 16 bits.

Data output is serial, an FPGA converts the data in a parallel stream before being stored into the FIFO.

For more information goto www.analogdevices.com and look up AD977.

2.3.1 ACQUISITION MODE

The customer may desire to think of the IP module as similar to a Digital Storage Oscilloscope (DSO). A DSO can store and display several waveforms and can record signals prior to the trigger. The trigger point is completely configurable by

reprogramming the FIFO's programmable empty flag. This is easily done, and the software DSP examples and programming FIFO reference demonstrate how.

Setting up and operating this type of acquisition is easy. First, the *Clock* and *Trigger Sources* need to be configured. If internal clocking is used, then the divisor needs to be programmed as well. In this mode, *Arming Source* should be set to PAE. The FIFO should be reset, and the desired PAE point programmed into the FIFO.

When the application is ready to capture data, it should write to the **Start Acquisition** address (value is not important). This action will cause data to be stored into the FIFO. Conversions will be saved in the FIFO until the programmed number of samples raises PAE. At this point, the trigger is armed, and further writes to the FIFO result in corresponding reads from the FIFO, discarding the earliest data.

Once the trigger event is seen (external signal, IPSTROBE, or software write to **Trigger Event**), the earliest data will no longer be discarded, and the FIFO fills to full.

Sampling continues until the FIFO FF flag is low then stopping the acquisition. If an early end to acquisition is desired, a write to **Stop Acquisition** will accomplish this.

Interrupts may be generated by INTREQ0 going high (indicating that triggering is now active) and by INTREQ1 going low (indicating that the acquisition is now finished).

Further explanation can be found in programming FIFO in the back of the manual.

2.3.2 CONTINUOUS MODE

If continuous acquisition is desired, then the following procedure is used. Remember that in a real world scenario, it is not possible to read the data at the maximum rate that the IP is capable of. There is no way to predict the exact performance as it depends on the carrier board and the application.

Setting up and operating this type of acquisition is also easy. First, the *Clock Source* needs to be configured. If internal clocking is used, then the divisor needs to be programmed as well. For *Trigger Source*, select Software Strobe Only. Set *Arming Source* to Never Discard. The FIFO should be reset, and the desired PAE point programmed into the FIFO.

INTREQ0 can be generated by PAE going high (indicating that the programmed number of samples is available to read from the FIFO).

When it is desired to start acquisition, write to **Start Acquisition** and then **Trigger Event**. When it is desired to stop acquisition, write to **Stop Acquisition**.

At the back of the manual there is a procedure on how to program the FIFO (72265) in more detail.

3. INTERFACE TO THE IP CARRIER

The IP carrier controls this IP via a set of registers in the IOSPACE.

3.1 REGISTERS

The registers are accessed in 16 bit mode. All the registers are 8 bits wide except for the hardware FIFO that is 16 bits wide. The addresses are provided referenced to the base I/O space on a slave carrier as well as the addresses utilized on a DSP or intelligent carrier.

ALPHI Slave Carrier	ALPHI DSP Carrier	R/W	BITS	WS@ 32MHz	Register
0x00	0x00	R/W	7-0	2	Internal Clock Divisor 0
0x02	0x01	R/W	7-0	2	Internal Clock Divisor 1
0x04	0x02	R/W	7-0	2	Internal Clock Divisor 2
0x06	0x03	WS	N/A	2	IPRST
0x08	0x04	WS	N/A	2	Trigger Event
0x0A	0x05				Unused
0x0C	0x06	WS	N/A	2	Stop Acquisition
0x0E	0x07	R/W	7-0	2	Acquisition Control Register
0x10	0x08	R/W	7-0	2	Source Intreq1
0x12	0x09	W		2	Clear Intreq0 latch
0x14	0x0A	R/W	7-0		Read/Write trigger register
0x16	0x0B	R/W	7-0	2	FIFO Control / Status
0x18	0x0C	WS	N/A	2	Reset FIFO
0x1A	0x0D	R/W	15-0	2	Hardware FIFO
0x1C	0x0E				Unused
0x1E	0x0F				Unused
0x20	0x10				Unused
0x22	0x11				Unused
0x24	0x12				Unused
0x26	0x13	WS	N/A	2	Start Acquisition
0x28	0x14	R/W	7-0	2	Interrupt Vector register
0x44	0x22	R/W	7-0	2	Channel Status register

Table 3.1 IO Registers

3.1.1 INTERNAL CLOCK DIVISOR [0, 1, 2]

These three 8 bit registers combine to form one 24 bit register which serves as a divisor on the IP clock when internal sampling clock is selected.

Program the Internal sample clock

- At 8Mhz for 32 channels the minimum clock period should be more than 150 uS. To set the Internal Clock at 200uS do the following.
- $00 - Internal clock divisor 0, write $3F.
- $01 - Internal clock divisor 1, write $6.

- At 32Mhz for 32 channels the minimum clock period should be more than150 uS. To set the Internal Clock at 200uS do the following.
- $00 - Internal clock divisor 0, write $FF.
- $01 - Internal clock divisor 1, write $18.
- Divider at 32Mhz is four times the divider for 8Mhz.

$$SamplingRate = \frac{IPClockFreq}{N+1}$$

3.1.2 TRIGGER EVENT (WRITE ONLY) $4

A write to this register will cause the trigger to occur, regardless of the setting in the **Acquisition Control Register**.

3.1.3 STOP ACQUISITION (WRITE ONLY) $6

A write to this register will stop acquisition immediately. The FIFO will probably not have full capacity stored, but a full set of samples from all enabled channels will have been stored.

3.1.4 ACQUISITION CONTROL REGISTER (READ/WRITE) $7

Bits 7-6	Bits 5-4	Bits 3-2	Bits 1-0
Sampling Clock Source	Trigger Source	Stop Acquisition Source	Trigger Enable Source

This register allows for the configuration of the acquisition state machine.

Sampling Clock Source determines where the A/D converters get their start conversion source from.

Sampling Clock Source	Meaning
0	Internal Sampling Clock (IP Clock divided by Divisor)
1	IPSTROBE

Trigger Source determines what event causes the acquisition to switch from pre-triggered to post-triggered. A write to **Software Strobe** will always cause a trigger, provided the pre-trigger data has been acquired.

Trigger Source	Meaning
0	**Software Strobe** only
1	IPSTROBE

Stop Acquisition Source determines the cause of ending the acquisition. It should probably be set to FF.

Stop Acquisition Source	Meaning
0	FIFO FF Flag
1	FIFO PAF Flag

Trigger Enable Source determines how the state machine decides that the trigger should be enabled. For burst acquisition, set to Enable After PAE Satisfied. For continuous acquisition, set to Always Enabled.

Trigger Enable Source	Meaning
0	Enable After PAE Satisfied
1	Always Enabled

3.1.5 SOURCE INTREQ1 (READ/WRITE) $8

This register allows to enable one signal source that will generate an interrupt Intereq1.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
Not Used	Not Used	END_ ACQ_ OUT	FIFO EF	FIFO PAE	FIFO HF	FIFO PAF	FIFO FF

FIFO FF

This bit, when low (0), indicates that there are 16384 samples stored in the hardware FIFO. It is usually an indication that the acquisition is completed.

FIFO PAF

This bit, when low (0), indicates that there is a programmable number of samples stored in the hardware FIFO. It has no meaning for the acquisition modes discussed in this manual.

FIFO HF

This bit, when low (0), indicates that there is a 16384/2 samples number of samples stored in the hardware FIFO. It has no meaning for the acquisition modes discussed in this manual.

FIFO PAE

This bit, when low (0), indicates that there is less than a programmable number of samples stored in the hardware FIFO. When this bit goes high, it indicates that the pre-trigger sampling requirement has been met, and that it is now possible to trigger the acquisition. In continuous modes, this bit being high indicates that there are N samples ready to be read, where N-1 is the number programmed in the FIFO.

FIFO EF
This bit, when low (0), indicates that there are no samples stored in the hardware FIFO.

END_ACQ_OUT
When the acquisition cycle is ended a raising signal will generate an interrupt.

Note:

The interrupt Intreq1 is latched. To reset the latch, a read of the FIFO Control/Status register is needed.

3.1.6 READ / WRITE TRIGGER (READ/WRITE) $A

This register allows the control of the trigger when an acquisition to the hardware FIFO is being made.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
INT1 RESET RDFIFO	Event Enable	MASTER	INT1 RESET EF	8-32Mhz	ACQ ONCE	N/A	CONV Source

CONV SOURCE
This bit, when high (1), enables the event input to be the source for start acquisition.

ACQ ONCE
This bit, when high (1), is to ReArm the acquisition upon FIFO empty (EF = 0).

8-32MHZ
This bit, when high (1), indicates to the state machine that the IP clock is set for 8Mhz.

Set IP-AD32-16 to run with IP-CLK at 8Mhz or 32Mhz.

- 8Mhz IP-CLK – Set carrier IP-CLK to 8Mhz. Next access Read/Write Trigger register at I/O space $0A and write $08 this sets the state machine for 8Mhz.
- 32Mhz IP-CLK – Set carrier IP-CLK to 32Mhz. Next access Read/Write Trigger register at I/O space $0A and write $00 this sets the state machine for 32Mhz.

INT1 RESET EF
This bit, when high (1), Is to remove INTREQ-1 when empty FIFO (EF = low).

MASTER
This bit, when high (1), will make the IP module the source for the sample clock, the signal will appear on IP-STROBE (P1 pin 46) as an output. The MASTER bit allows the synchronization of multiple IP's.

EVENT ENABLE
This bit, when high (1) enables the external event input on P2 pin 24. The event pin can be used to externally start conversion.

INT 1 RESET RDFIFO
This bit , when high (1) Is to remove INTREQ-1 when RDFIFO goes low.

3.1.7 FIFO CONTROL / STATUS (READ/WRITE) $B

This register allows for querying the current state of the hardware FIFO flags and a means to program the FIFO. Writing to bits 01 and 02 will setup calibration mode and one channel access. For more information go to page 8 - 9.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
FIFO FF	FIFO PAF	FIFO PAE	FIFO EF	HALF FULL	CAL MODE	ONE CHANNEL	FIFO REG

FIFO FF
This bit, when low (0), indicates that there are 16384 samples stored in the hardware FIFO. It is usually an indication that the acquisition is completed.

FIFO PAF
This bit, when low (0), indicates that there is a programmable number of samples stored in the hardware FIFO. It has no meaning for the acquisition modes discussed in this manual.

FIFO PAE
This bit, when low (0), indicates that there is less than a programmable number of samples stored in the hardware FIFO. When this bit goes high, it indicates that the pre-trigger sampling requirement has been met, and that it is now possible to trigger the acquisition. In continuous modes, this bit being high indicates that there are N samples ready to be read, where N-1 is the number programmed in the FIFO.

FIFO EF
This bit, when low (0), indicates that there are no samples stored in the hardware FIFO.

ONE CHANNEL
When this bit is set to 1, the output multiplexers are forced to the channel specified in the **LAST CHANNEL** register. Only a single conversion will be triggered for each trigger signal. If this bit is cleared to 0 then the state machine functions normally, scanning channels from 0 to the channel specified in the **LAST CHANNEL** register.

FIFO REG
This bit, when low (0), indicates reads of the **Hardware FIFO** will access the contents of the FIFO. When the bit is set high (1), indicates writes and reads of the FIFO will access the FIFO programming registers.

3.1.8 RESET THE HARDWARE FIFO (WRITE ONLY) $C

A write to this location will reset the hardware FIFO to empty. It is required to initialize the FIFO before any access is made.

3.1.9 HARDWARE FIFO (READ-WRITE) $D

If the **FIFO REG** bit is cleared (0), then a read of this location will respond with the oldest pair of samples in the FIFO. If the **FIFO REG** bit is set (1), then a read of this location will read the internal configuration registers of the FIFO.

Data is stored as 16 bit words. Only enabled channels will be found in the FIFO. If the **FIFO REG** bit is set (1), then a write to this location will program the internal configuration registers of the FIFO.

The FIFO is actually an IDT 72265 (16,384 x 18) or equivalent.

3.1.10 START ACQUISITION (WRITE ONLY) $13

A write to this register will start acquisition immediately. Data will be stored until the pre-trigger condition is met (PAE goes high).

If continuous acquisition is desired, then a write here must be followed with a write to **Trigger Acquisition**.

3.1.11 INTERRUPT VECTOR REGISTER (READ/WRITE) $14

An 8bit Interrupt Vector register is available for future applications

3.1.12 CHANNEL STATUS (READ/WRITE) $22

This register at address $22 allows for programming querying the current state of the last channel register previously discussed in the manual. For more information go to page 9 in this manual.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
N/A	N/A	N/A	MUX 4	MUX 3	MUX 2	MUX 1	MUX 0

4. PROGRAMMING (72265) FIFO

At the base address of the I/O access of your carrier board enter the following.

Program Acquisition mode

Acquisition will start upon receiving a software command

- Generate an interrupt at PAE pointer location (127 locations from beginning if not programmed)
- Fill the FIFO and stop when FF is activated (low)

The example will show how to program the FIFO pointer (PAE and PAF) if needed, start the acquisition, read the FIFO register pointer. Read data from FIFO and start over again.

Set IP-AD32-16 to run with IP-CLK at 8Mhz or 32Mhz.

- 8Mhz IP-CLK – Set carrier IP-CLK to 8Mhz. Next access Read/Write Trigger register at I/O space $0A and write $08 this sets the state machine for 8Mhz.
- 32Mhz IP-CLK – Set carrier IP-CLK to 32Mhz. Next access Read/Write Trigger register at I/O space $0A and write $00 this sets the state machine for 32Mhz.

Program the Internal sample clock

- At 8Mhz for 32 channels the minimum clock period should be more than 150 uS. To set the Internal Clock at 200uS do the following.
- $00 - Internal clock divisor 0, write $3F.
- $01 - Internal clock divisor 1, write $6.
- At 32Mhz for 32 channels the minimum clock period should be more than150 uS. To set the Internal Clock at 200uS do the following.
- $00 - Internal clock divisor 0, write $FF.
- $01 - Internal clock divisor 1, write $18.
- Divider at 32Mhz is four times the divider for 8Mhz.

Program PAE and PAF pointer (optional)

- PAE is equal to 127 upon hardware FIFO reset or Power on reset. PAF is equal to FIFO full less 127 location
- $0B - FIFO control/status, write $1 set FIFO_REG bit to “1” high allows access of the FIFO programming registers (PAE and PAF pointers).

Note: Read or write at this same address provides access to the registers in a circular mode.

- $0C - Hardware FIFO reset, write $0 a write to this location will reset the hardware FIFO and

Enable access to the FIFO registers. When reset goes high, status of FIFO_REG bit defines access to FIFO registers or data FIFO.

- Program FIFO register. (example 64 locations)
- $0D - Hardware FIFO, should read $7F (PAE = 128 locations)
- $0D - Hardware FIFO, write $3F (PAE = 64 locations).
- $0D - Hardware FIFO, should read $3F
- $0B - FIFO control/status, write $0 FIFO_Register bit = "0". Now we are in FIFO data normal mode (data are written inside the FIFO at end of conversion). Data can be read from the FIFO.

Clear Intreq0 latch

- $09 - Write $0 Note:Intreq0 sole source is PAE active(high)

Start FIFO acquisition.

- $13 - Start Acquisition, write $0. A write to this register will start acquisition immediately.
- After starting acquisition – you should see INT_REQ0 go low when PAE pointer is reached and stay low.
- A display of the signal WR_FIFO will show two accesses per A/D before INT-REQ0 goes low.
- Acquisition will continue until FIFO is FIFO FF active (default). PAF can be selected by programming register $7 Acquisition Control register

End acquisition

FIFO is full, lets read data from FIFO

- $0B - FIFO control/status, read FIFO status should read $30.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
FIFO FF	FIFO PAF	FIFO PAE	FIFO EF	FIFO HF	N/A	N/A	FIFO REG

- $0D - Hardware FIFO, read data in the FIFO = 1 time. This will clear the FIFO-FF.
- $0B - FIFO control/status, FIFO-FF should be back to a 1 – should read $B0.
- If read more than 127 times then the PAF (127 location before the FIFO full as default) will be cleared – should read $F0.
- If sufficient read is performed (more than the PAE location pointer) INTREQ0 can be re-programmed by clearing INT latch at address $09

INT_REQ1 initialization

User can program the INTREQ1 of the IP module to eventually “toggle” the reading of data if speed is not a requirement

Select source for INTREQ1

- $08 write $2 enables FIFO-PAF as source of interrupt.
- Now when starting acquisition you will have Intreq1 active low when PAF goes low.
- Interrupt is cleared when reading Control Register at address $0B. The Interrupt will be activated again if PAF has been released.

4.1 ACQUISITION IN A PRE-TRIGGER AND POST TRIGGER MODE.

Upon a start acquisition command, the FIFO will acquire data until PAE pointer is reached. Then the FIFO will stay IDLE waiting to receive a signal (End acquisition Idle) to resume storing data.

The acquisition is still storing the new data and discarding at the same time the oldest data.

After the acquisition has resumed, the FIFO will be filled up to FF or PAF as example above.

Proceed as example 1 to initialize the PAE, PAF pointer if needed.

Programmation of the PAE pointer as the idle location

Write $1 to the Acquisition control register address ($7). PAE will now be the source for the idle point.

Acquisition will start as above, storing data up to the idle pointer and discarding old data when new data is stored then it will wait for a signal to resume storing data until end of acquisition. In this example we use a software command address $4 (Trigger Event). Other sources are described in the manual.

Start Acquisition

- FIFO control/status address $0B , should read $F8
- $13 write $0 at address location $13

  A write to this register will start acquisition immediately.

  PAE interrupt is generated (INTREQ0).

  Display of the WRFIFO and RDFIFO lines when INTREQ0 goes low should show RDFIFO active WHEN WRFIFO is active.

RESUME acquisition by ending IDLE mode

- $04 - Trigger source, write $00 this command is used to resume acquisition.
- $0B - FIFO control/status, should read $30. The acquisition has ended with FF active. Other conditions can be programmed (see manual)

  0 Always enable

  1 Enable after PAE satisfied

4.2 REAL-TIME CONTINUOUS ACQUISITION

Using Internal Clock and reading of A/D data using DMA state Machine for read of all 32 locations

This example shows how to program the IP-AD3216 using the Internal clock as a sample clock. After starting initialization, the host arms the acquisition. The next sample clock will start the acquisition on the 32 channels simultaneously. An INTREQ-0 is then asserted on the first acquisition. The 32 A/D channels results are stored inside the FIFO. Setting up the hardware FIFO with PAE pointer set to 0 and PAF pointer set to 3FE0, INTREQ-1 will be activated when 32 samples are written, then the acquisition will stop.

The DMA state machine can now read the 32 results. At the end of the 32 reads the state machine is then ReArmed for the next sample clock.

INTREQ1 can be programmed to be removed upon the first read of the FIFO, or FIFO being empty (EF = 0). In this case PAE and EF are changing at the same time, due to the fact that PAE pointer is set to zero.

The following will explain how to program the IP-AD3216. Each address is written starting from the base address of the I/O space on the carrier board.

Program register for Internal Clock and Stop Acquisition Source

- $07 -write $04 this sets up Internal clock and Stop acquisition source to be PAF.

Bits 7-6	Bits 5-4	Bits 3-2	Bits 1-0
Sampling Clock Source	Trigger Source	Stop Acquisition Source	Trigger Enable Source

Sampling Clock Source determines where the A/D converters get their start conversion source from.

Sampling Clock Source	Meaning
00	Internal Sampling Clock (IP Clock divided by Divisor)
01	IPSTROBE

Stop Acquisition Source determines the cause of ending the acquisition.

Stop Acquisition Source	Meaning
00	FIFO FF Flag
01	FIFO PAF Flag

Set IP-AD32-16 to run with IP-CLK at 8Mhz or 32Mhz.

- 8Mhz IP-CLK – Set carrier IP-CLK to 8Mhz. Next access Read/Write Trigger register at I/O space $0A and write $08 this sets the state machine for 8Mhz.
- 32Mhz IP-CLK – Set carrier IP-CLK to 32Mhz. Next access Read/Write Trigger register at I/O space $0A and write $00 this sets the state machine for 32Mhz.

Program the Internal sample clock

- At 8Mhz for 32 channels the minimum clock period should be more than 150 uS. To set the Internal Clock at 200uS do the following.
- $00 - Internal clock divisor 0, write $3F.
- $01 - Internal clock divisor 1, write $6.
- At 32Mhz for 32 channels the minimum clock period should be more than150 uS. To set the Internal Clock at 200uS do the following.
- $00 - Internal clock divisor 0, write $FF.
- $01 - Internal clock divisor 1, write $18.
- Divider at 32Mhz is four times the divider for 8Mhz.

Program PAF as source for INTREQ-1

- $08 - write 2 This will cause INTREQ-1 to go low when PAF goes low.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
Not Used	Not Used	END_ ACQ_ OUT	FIFO EF	FIFO PAE	FIFO HF	FIFO PAF	FIFO FF

Program Read / Write register

- $0A write – $14 this enables (Acq Once bit #2) and (INTREQ-1 reset EF bit #4). Acq Once - the purpose is to Rearm the acquisition upon FIFO empty (EF = 0). INT1 reset EF - when this bit is enabled INTREQ1 is removed when EF goes low.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
INT1 RESET RDFIFO	Event Enable	Master	INT1 RESET EF	8-32Mhz	ACQ ONCE	IDLE Enable	CONV Source

Clear INTREQ-0 latch

- $09 write $0 Note: INTREQ-0 is the sole source for PAE active (high). INTREQ-0 is used only to inform the beginning of the first acquisition cycle. It will stay low until reset. It has no effect on further behavior of the state machine.

Program PAE and PAF pointer

PAE is equal to 127 (7F hex) upon hardware FIFO reset or Power on reset. PAF is equal to FIFO full less 127 location

- $0C - Hardware FIFO reset, write $0 a write to this location will reset the hardware FIFO and enable access to the FIFO registers.
- $0B - FIFO control/status, Should read $C8 now write $1 This sets the FIFO_REG bit to “1” (high) which allows access of the FIFO programming registers (PAE and PAF pointers).

Program FIFO register

- Note: Read or write at this same address provides access to the registers in a circular mode.

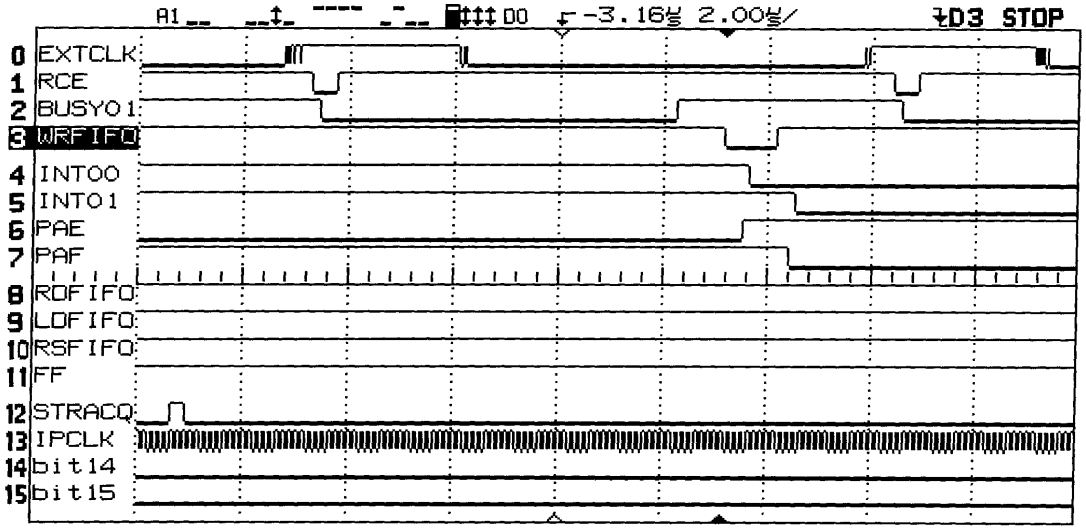
  First access programs PAE, after writing increment on the same address without making a read. The access programs PAF, after writing exit the register.
- $0D - Hardware FIFO, should read $7F (PAE = 128 locations)
- $0D - Hardware FIFO, write $0 for PAE then increment without a read of address.
- $0D - Hardware FIFO, write $3FE0 and exit register
- $0B - FIFO control/status, write $0 This sets the FIFO_REG bit = “0” (low). Now we are in FIFO data normal mode (data are written inside the FIFO at end of conversion). Data can be read from the FIFO.

Start FIFO Acquisition

- $13 - Start Acquisition, write $0 A write to this register will arm the state machine. The next Ext-CLK signal will start a conversion.

- After starting acquisition - should see INT_REQ0 go low when PAE pointer is reached and stay low.
- A display of the signal WR_FIFO will show 2 access for each A/D then INT-REQ1 goes low.
- Acquisition will continue until PAF goes low and INTREQ-1 goes low.
- See timing for example of first acquisition.

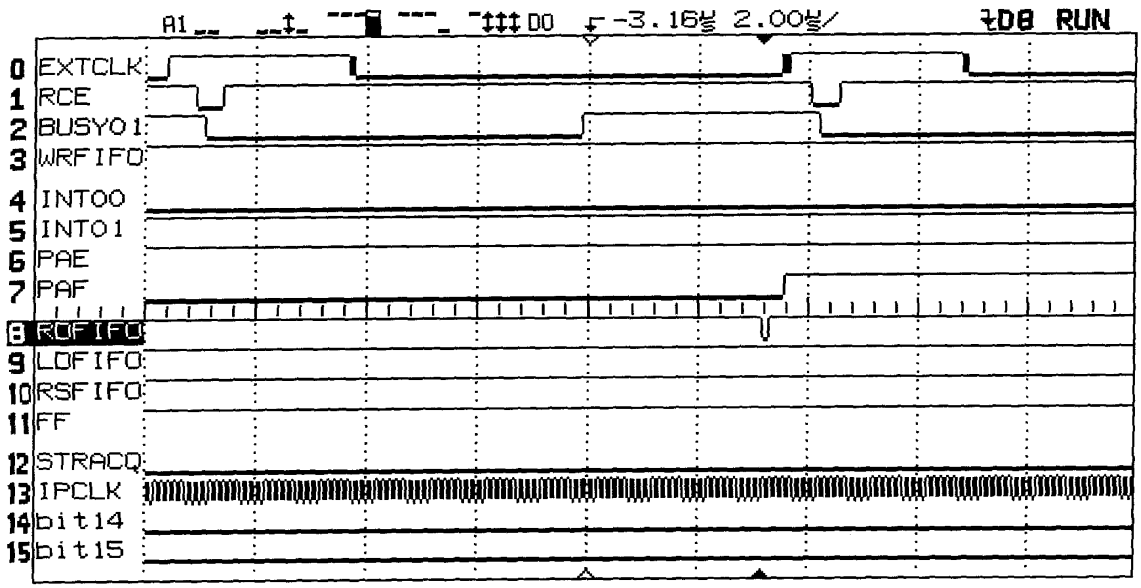
Start Acquisition timing



Read data from FIFO

- $0D – Hardware FIFO, Make first read of FIFO. Notice PAF goes high, meaning there is less than 32 data left in the FIFO.
- See timing for example of first read.First

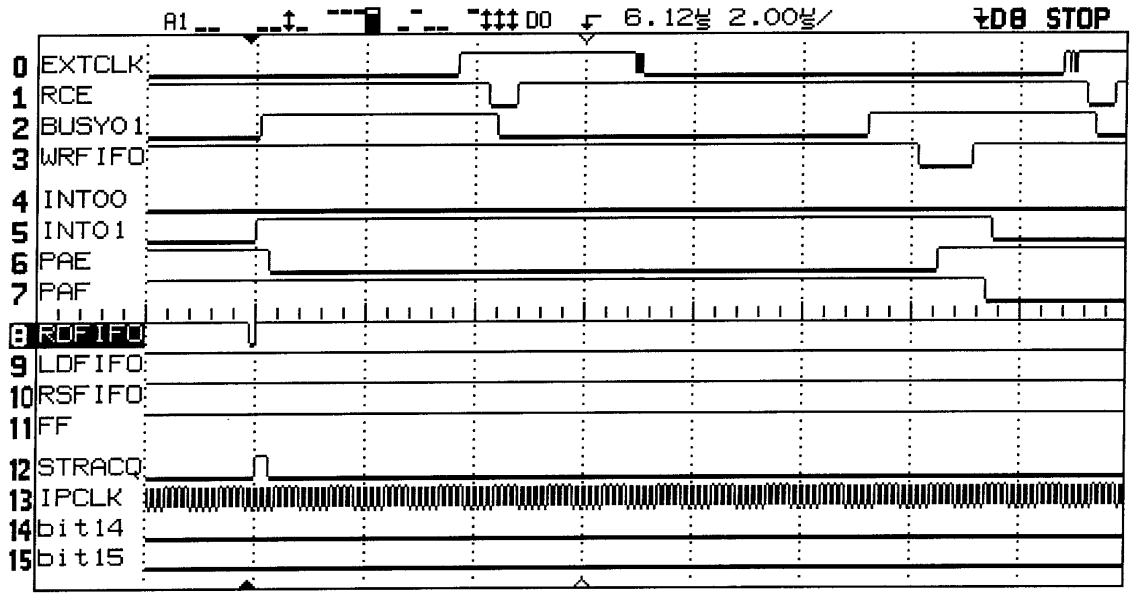
Read of hardware FIFO



Continue reading FIFO data

- $0D – Hardware FIFO, Continue reading the FIFO. After the 32nd read the PAE pointer goes low and INTREQ-1 goes high, Rearming itself for the next acquisition upon receiving sampling clock.
- See timing for example of the 32nd (last read).

Last read of hardware FIFO



4.3 ONE CHANNEL REAL-TIME ACQUISITION

This example shows how to program the IP-AD3216 using the Internal clock as a sample clock. After starting initialization, the host arms the acquisition. The next sample clock will start the acquisition on the channels selected. An INTREQ-0 is then asserted on the first acquisition. The selected A/D channels results are stored inside the FIFO. Setting up the hardware FIFO with PAE pointer set to 0 and PAF pointer set to 3FFE, INTREQ-1 will be activated when selected samples are written, then the acquisition will stop.

The selected channels results can now be read. At the end of the reads PAE then ReArmed for the next sample clock.

INTREQ1 can be programmed to be removed upon the first read of the FIFO, or FIFO being empty (EF = 0). In this case PAE and EF are changing at the same time, due to the fact that PAE pointer is set to zero.

The following will explain how to program the IP-AD3216. Each address is written starting from the base address of the I/O space on the carrier board.

Program register for Internal Clock and Stop Acquisition Source

- $07 -write $04 this sets up Internal clock and Stop acquisition source to be PAF.

Bits 7-6	Bits 5-4	Bits 3-2	Bits 1-0
Sampling Clock Source	Trigger Source	Stop Acquisition Source	Trigger Enable Source

Sampling Clock Source determines where the A/D converters get their start conversion source from.

Sampling Clock Source	Meaning
00	Internal Sampling Clock (IP Clock divided by Divisor)
01	IPSTROBE

Stop Acquisition Source determines the cause of ending the acquisition.

Stop Acquisition Source	Meaning
00	FIFO FF Flag
01	FIFO PAF Flag

Set IP-AD32-16 to run with IP-CLK at 8Mhz or 32Mhz.

- 8Mhz IP-CLK – Set carrier IP-CLK to 8Mhz. Next access Read/Write Trigger register at I/O space $0A and write $08 this sets the state machine for 8Mhz.
- 32Mhz IP-CLK – Set carrier IP-CLK to 32Mhz. Next access Read/Write Trigger register at I/O space $0A and write $00 this sets the state machine for 32Mhz.

Program the Internal sample clock

- At 8Mhz for 32 channels the minimum clock period should be more than 150 uS. To set the Internal Clock at 200uS do the following.
- $00 - Internal clock divisor 0, write $3F.
- $01 - Internal clock divisor 1, write $6.
- At 32Mhz for 32 channels the minimum clock period should be more than150 uS. To set the Internal Clock at 200uS do the following.
- $00 - Internal clock divisor 0, write $FF.
- $01 - Internal clock divisor 1, write $18.
- Divider at 32Mhz is four times the divider for 8Mhz.

Set IP-AD32-16 to run with IP-CLK at 8Mhz or 32Mhz.

- 8Mhz IP-CLK – Set carrier IP-CLK to 8Mhz. Next access Read/Write Trigger register at I/O space $0A and write $08 this sets the state machine for 8Mhz.
- 32Mhz IP-CLK – Set carrier IP-CLK to 32Mhz. Next access Read/Write Trigger register at I/O space $0A and write $00 this sets the state machine for 32Mhz.

Program the Internal sample clock

- At 8Mhz for 32 channels the minimum clock period should be more than 150 uS. To set the Internal Clock at 200uS do the following.
- $00 - Internal clock divisor 0, write $3F.
- $01 - Internal clock divisor 1, write $6.
- At 32Mhz for 32 channels the minimum clock period should be more than150 uS. To set the Internal Clock at 200uS do the following.
- $00 - Internal clock divisor 0, write $FF.
- $01 - Internal clock divisor 1, write $18.
- Divider at 32Mhz is four times the divider for 8Mhz.

Program PAF as source for INTREQ-1

- $08 - write 2 This will cause INTREQ-1 to go low when PAF goes low.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
Not Used	Not Used	END_ ACQ_ OUT	FIFO EF	FIFO PAE	FIFO HF	FIFO PAF	FIFO FF

Program Read / Write register

- $0A write – $10 this enables (INTREQ-1 reset EF bit #4). INT1 reset EF - when this bit is enabled INTREQ1 is removed when EF goes low.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
INT1 RESET RDFIFO	Event Enable	Master	INT1 RESET EF	8-32Mhz	ACQ ONCE	IDLE Enable	CONV Source

Program PAE and PAF pointer

PAE is equal to 127 (7F hex) upon hardware FIFO reset or Power on reset. PAF is equal to FIFO full less 127 location

- $0C - Hardware FIFO reset, write $0 a write to this location will reset the hardware FIFO and enable access to the FIFO registers.
- $0B - FIFO control/status, Should read $C8 now write $1 This sets the FIFO_REG bit to "1" (high) which allows access of the FIFO programming registers (PAE and PAF pointers).

Program FIFO register

- Note: Read or write at this same address provides access to the registers in a circular mode.

  First access programs PAE, after writing increment on the same address without making a read. The access programs PAF, after writing exit the register.

- $0D - Hardware FIFO, should read $7F (PAE = 128 locations)
- $0D - Hardware FIFO, write $0000 for PAE then increment without a read of address.
- $0D - Hardware FIFO, write $3FFE and exit register
- $0B - FIFO control/status, write $0 This sets the FIFO_REG bit = "0" (low). Now we are in FIFO data normal mode (data are written inside the FIFO at end of conversion). Data can be read from the FIFO.

Clear INTREQ-0 latch

- $09 write $0 Note: INTREQ-0 is the sole source for PAE active (high). INTREQ-0 is used only to inform the beginning of the first acquisition cycle. It will stay low until reset. It has no effect on further behavior of the state machine.

LAST CHANNEL REGISTER (Read/Write) $22

- $22 write $00 for channels IN01/IN02

LAST CHANNEL	Single Ended	LAST CHANNEL	Single Ended
$0	IN01/IN02	**$8**	IN17/IN18
$1	IN03/IN04	**$9**	IN19/IN20
$2	IN05IN06	**$A**	IN21/IN22
$3	IN07IN08	**$B**	IN23/IN24
$4	IN09/IN10	**$C**	IN25/IN26
$5	IN11/IN12	**$D**	IN27/IN28
$6	IN13/IN14	**$E**	IN29/IN30
$7	IN15/IN16	**$F**	IN31/IN32

Last Channel Register Single-Ended Mode One Channel Set

Set One Channel bit

$0B - FIFO control/status, write $2 This sets the One channel bit to acquire data from the channel programmed in the last channel register.

BIT 07	BIT 06	BIT 05	BIT 04	BIT 03	BIT 02	BIT 01	BIT 00
FIFO FF	FIFO PAF	FIFO PAE	FIFO EF	HALF FULL	CAL MODE	ONE CHANNEL	FIFO REG

Start FIFO Acquisition

- $13 - Start Acquisition, write $0 A write to this register will arm the state machine. The next Ext-CLK signal will start a conversion.
- After starting acquisition - should see INT_REQ0 go low when PAE pointer is reached and stay low.
- A display of the signal WR_FIFO will show 2 access for each A/D then INT-REQ1 goes low.
- Acquisition will continue until PAF goes low and INTREQ-1 goes low.
- See timing for example of first acquisition.

Read data from FIFO

$0D – Hardware FIFO, Make first read of FIFO. PAF should go high and INTREQ-0 should go low.

Continue reading FIFO data

- $0D – Hardware FIFO, Continue reading the FIFO. After the 2nd read the PAE pointer goes low and INTREQ-1 goes high and acquisition is complete.

5. HARDWARE DETAILS

5.1 RESET SIGNALS

The **ATC-AD3216** is reset when the IP carrier issues a reset.

5.2 CONNECTORS

The connector placement is depicted below. There are no configuration jumpers.

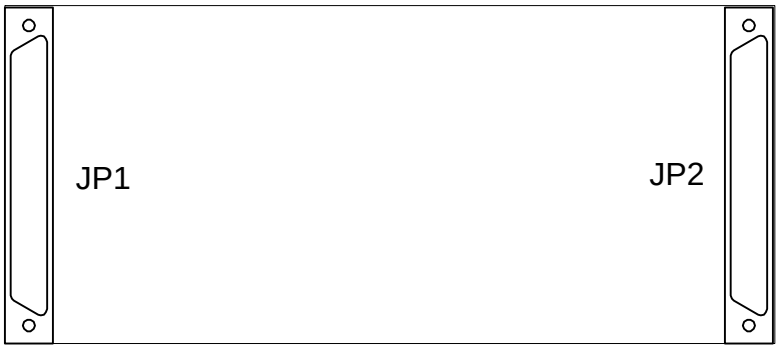


Figure 5.1: Connector Locations

5.2.1 CONNECTOR DESCRIPTIONS

IP External I/O Connector (JP3)

A 50 pin subminiature D shelled connector is used to route the analog signals to the IP. The IP carrier then takes these signals and presents them for customer use. See the documentation for the IP carrier for more details.

The signals are routed as follows.

Pin	Connection	Pin	Connection
1	IN01/IN01+	26	IN03/IN03+
2	IN017/IN01-	27	IN19/IN03-
3	GNDA	28	GNDA
4	IN05/IN05+	29	IN07/IN07+
5	IN21/IN05-	30	IN23/IN07-
6	GNDA	31	GNDA
7	IN09/IN09+	32	IN11/IN11+
8	IN25/IN09-	33	IN27/IN11-
9	GNDA	34	GNDA
10	IN13/IN13+	35	IN15/IN15+
11	IN29/IN13-	36	IN31/IN15-
12	GNDA	37	GNDA
13	IN02/IN02+	38	IN04/IN04+
14	IN18/IN02-	39	IN20/IN04-
15	GNDA	40	GNDA
16	IN06/IN06+	41	IN08/IN08+
17	IN22/IN06-	42	IN24/IN08-
18	GNDA	43	GNDA
19	IN10/IN10+	44	IN12/IN12+
20	IN26/IN10-	45	IN28/IN12-
21	GNDA	46	GNDA
22	IN14/IN14+	47	IN16/IN16+
23	IN30/IN14-	48	IN32/IN16-
24	-15VIN	49	+15VIN
25	EXTRIG	50	GND

Table 5.2: IP External I/O Connector (JP3)

