MicroLAB Systems

Ultimate DSP Development Solutions

TI Third Party Network Member DSP TEXAS INSTRUMENTS

DIGITAL SIGNAL PROCESSING

T/PDAS-AD16/16Q/1M-DA4/16Q/1M

16-bit 1 Msps Multichannel Instrumentation PIOX-16 AD/DA DCM
for *TORNADO* DSP Systems/Controllers

User's Guide

covers:
T/PDAS-AD16/16Q/1M-DA4/16Q/1M rev.1A

MicroLAB Systems Ltd
E-mail: info@mlabsys.com *WEB:* www.mlabsys.com *FTP:* ftp.mlabsys.com

CAUTION

Фирма *МикроЛАБ Системс Лтд* (*МЛС*) оставляет за собой право вносить любые изменения и прекращать выпуск и поддержку любых изделий и программного обеспечения, упоминаемых в настоящем документе, без какого-либо предварительного уведомления, если иное специально не оговаривается. *МЛС* рекомендует своим покупателям пользоваться подлинными и самыми последними версиями фирменных информационных документов и осуществлять предварительное консультирование с фирмой перед размещением заказа, чтобы быть уверенным, что настоящая информация достоверна и применима на текущий момент.

МЛС гарантирует качество и соответствие технических параметров поставляемой продукции приведенной технической спецификации. Всякое тестирование и проверка продукции производятся фирмой в степени и объемах, необходимых для поддержки настоящей гарантии. Какое-либо дополнительное тестирование и проверка продукции на соответствие другим требованиям проводятся лишь и только в случаях, выполняемых по специальным заказам, или тогда, когда это специально оговаривается.

МЛС не несет никакой ответственности за правильность функционирования и работоспособность оборудования и программного обеспечения, разработанного и изготовленного с применением продукции (или отдельных ее компонентов) фирмы, если это не подтверждено специальным фирменным сертификатом *МЛС*.

Продукция *МЛС* не предназначена для применения в аппаратуре, системах или изделиях для поддержки жизнидеятельности. Применение продукции *МЛС* в таком оборудовании категорически запрещено без специального письменного подтвержедения от *МЛС* или оригинального фирменного сертификата *МЛС*.

IMPORTANT NOTICE

Настоящая продукция предназначена для использования в составе лабораторного тестового и научно-исследовательского оборудования. *МЛС* не несет ответственности за работоспособность настоящей продукции в составе другого типа оборудования и/или в отличных от специфицированных условиях эксплуатации. При повреждениях настоящей продукции, вызванных ее применением в составе другого типа оборудования и/или условий эксплуатации, гарантийные обязательства аннулируются без какого-либо возмещения ущерба и ремонт производится за счет покупателя.

Настоящая продукция генерирует, использует и может излучать радиочастотную энергию, которая может создавать радиочастотные помехи для другой аппаратуры, несмотря на все конструктивные и другие меры, предпринятые для минимизации создаваемых помех. Однако, в случае возникновения помех для работы другой аппаратуры покупатель должен сам и за свой счет принять меры для их устранения или уменьшения.

ITEMS OF LICENSE AGREEMENT

Никакие части настоящего документа, аппаратные и программные части настоящей продукции не могут быть реассемблированы, ретрассированы и/или изменены с целью восстановления и/или изменения электрической схемы, конструкции, алгоритма работы или принципа функционирования любыми методами, воспроизведены, скопированы, запомнены в архивах с возможностью воспроизведения, а также переданы по средствам связи в любом виде и любыми методами, будь то электронные, механические, копировальные, фотографические, записывающие или другие, без предварительно выданного фирменного письменного разрешения от *МЛС*. Нарушение настоящего положения вне зависимости от приобретения настоящей продукции и/или документа трактуется как нарушение авторских прав и преследуется по закону.

Приобретение настоящей продукции автоматически означает согласие покупателя с положениями лицензионного соглашения, равно как и другими положениями закона об авторских правах. Нарушение настоящих положений, равно как и других положений закона об авторских правах, трактуется как нарушение авторских прав, преследуется по закону и автоматически ведет к аннулированию всех обязательств *МЛС* по поддержке настоящей продукции.

DOC: MLS-SPUG-189E revision 1A
January 2002

About this Document

This user's guide contains description for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* multi-channel instrumentation AD/DA daughter-card module (DCM) for *TORNADO* DSP systems/controllers.

This document does not include detail description neither for the on-board components nor for the corresponding software and hardware applications. To get the corresponding information refer to the following documentation:

1. ***TORNADO-3x. User's Guide.*** MicroLAB Systems, 1999.
2. ***TORNADO-P33. User's Guide.*** MicroLAB Systems, 2000.
3. ***TORNADO-6x. User's Guide.*** MicroLAB Systems, 1998.
4. ***TORNADO-P6x. User's Guide.*** MicroLAB Systems, 2002.
5. ***TORNADO-E3x. User's Guide.*** MicroLAB Systems, 2000.
6. ***TORNADO-E6x. User's Guide.*** MicroLAB Systems, 2002.

Warranty

The warranty period for all hardware and software products manufactured by MicroLAB Systems Ltd is *one year* after shipment. MicroLAB Systems Ltd guarantees free of charge repair or replacement for the manufacturer caused damaged products during warranty period. Software updates will be sent free of charge to the customer during warranty period.

Product registration procedure

MicroLAB Systems strongly recommends that you register each of your purchased hardware/software product in order to get free product updates and free technical support within the warranty period.

The registration procedure is as easy as the following:

- Open the PRODUCT REGISTRATION FORM, which is contained in the REGISTER.TXT text file available either on the MicroLAB Systems 'Technical & Programming Guide' CD-ROM or from the MicroLAB Systems FTP-site. If you are unable to locate the REGISTER.TXT file, then call/email to MicroLAB Systems.
- Fill in the applicable fields of the PRODUCT REGISTRATION FORM. It is important that you will to specify your name, post address, phone/fax, email address, purchased product name and serial number, and the product reception date.
- Return the PRODUCT REGISTRATION FORM to MicroLAB Systems either via email, fax or regular mail.

Note, that all product purchased from MicroLAB Systems shall be registered within 90 days after the date of the shipment.

If you need assistance, documentation or information...

Should you need technical assistance for purchased MicroLAB Systems Ltd products, or if you want to order additional documentation, or if you want to get latest information about MicroLAB Systems Ltd products, please email, call, fax or post to MicroLAB Systems Ltd customer support service:

address: 83 Dubninskaya str., #612, 127591, Moscow, RUSSIA
MicroLAB Systems Ltd

phone/fax: +7-(095)-900-6208
information request: info@mlabsys.com
technical support: support@mlabsys.com
product registration: register@mlabsys.com
WWW: http://www.mlabsys.com
FTP: ftp://ftp.mlabsys.com

Trademarks

TORNADO-3x, TORNADO-4x, TORNADO-54x, TORNADO-P6x, TORNADO-P3x, TORNADO-PX, TORNADO-SX, TORNADO-E, MIRAGE-P510D, UECMX, PIOX-Link are trademarks of *MicroLAB Systems Ltd*

TMS320, XDS510, Code Composer are trademarks of *Texas Instruments Inc*

Other trademarks and company names used are trademarks of their respective holders.

Contents

Figures

Tables

Chapter 1. Introduction

This chapter contains general description for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* 16-bit 1 Msps multi-channel instrumentation AD/DA PIOX-16 daughter-card module (DCM) for *TORNADO* DSP systems/controllers.

1.1 General Information

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM (fig.1-1) provides 16-channel 16-bit 1 Msps A/D and 4-channel 16-bit 1 Msps D/A on-board data acquisition and plugs into 16-bit parallel I/O expansion (PIOX-16) site of any of *TORNADO* DSP systems (*TORNADO-3x/54x/6x/P3x/P6x/etc*) and *TORNADO-E* stand-alone DSP controllers (*TORNADO-E3x/E6x/etc*) from MicroLAB Systems Ltd.

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM has been designed for high-frequency multi-channel and ultra-high accuracy analog I/O instrumentation applications.

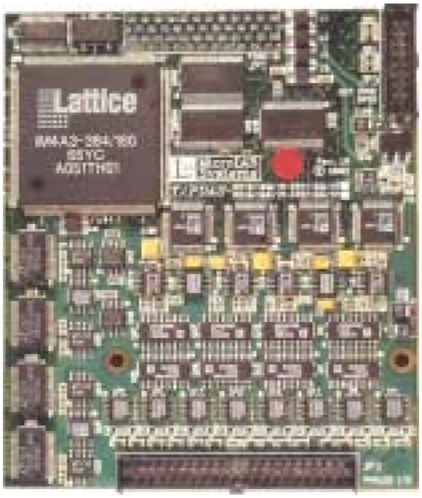

Fig. 1-1. T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM.

Installation onto TORNADO DSP System/Controller

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM installs as PIOX-16 DCM (fig.1-2) into the PIOX-16 site onto *TORNADO* DSP system/controller mainboard.

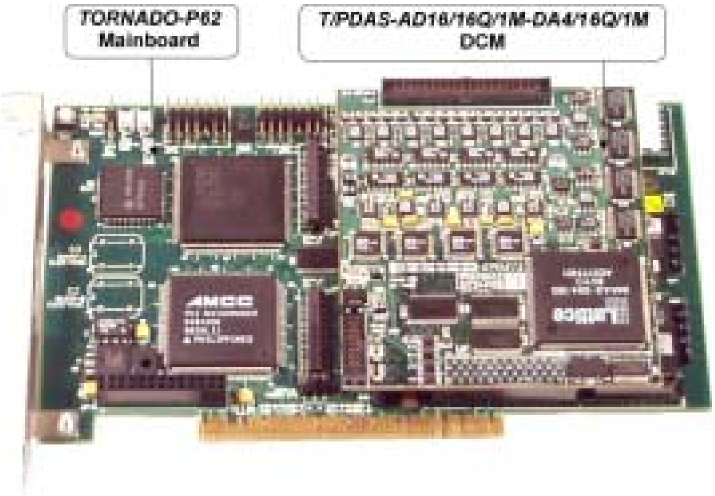

Fig. 1-2. T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM installed onto *TORNADO-P62* mainboard.

Overview

T/PDAS-AD16/16Q/1M-DA4/16Q/1M PIOX-16 AD/DA DCM comprises of the following main on-board components:

- analog input section comprising of four analog-to-digital conversion (ADC) channels with synchronous sampling
- each ADC channel includes 4:1/SE (2:1/DIFF) analog input multiplexer (for the total of either 16 single ended analog inputs or 8 differential analog inputs), and 16-bit 1 Msps ADC chip
- high-density FIFO for the ADC output data
- analog output section with comprising of four digital-to-analog conversion (DAC) channels with synchronous sampling
- each DAC channel includes 16-bit 1 Msps DAC chip
- high-density FIFO for DAC input data
- data acquisition control logic
- programmable high-resolution sampling frequency generator
- 4-bit general purpose I/O
- host PIOX-16 interface.

On-board quad-channel synchronously sampled analog input and analog output sections feature 16-bit resolution and excellent linearity at up to 1 MHz sampling frequency, which guarantee minimum signal distortion during A/D and D/A conversions at high sampling frequencies.

Each ADC channel provides software configured either 4:1 analog input multiplexer for single-ended (SE) input signal or 2:1 analog input multiplexer for differential (DIFF) input signal, which allows to select four particular analog input signals from total 16 analog input for A/D conversion for each sampling period. This allows to convert four either SE or DIFF analog inputs at up to 1 MHz maximum sampling frequency using one scan cycle, eight SE/DIFF analog inputs at up to 500 kHz maximum sampling frequency suing two scan cycles, twelve SE/DIFF analog inputs at up to 333 kHz maximum sampling frequency using three scan cycles, and sixteen SE analog inputs at up to 250 kHz maximum sampling frequency using four scan cycles.

ADC output data can be either read directly from host PIOX-16 interface in each sampling/scan cycle, or can be stored in on-board high-density ADC FIFO for further download to host DSP environment. DAC input data can be either loaded directly from host PIOX-16 interface in each sampling/scan cycle, or can be loaded from on-board high-density DAC FIFO, which can be preliminary uploaded from host DSP environment. Real-time ADC/DAC data download/upload to/from host DSP environment can be either performed directly by host DSP software, or can be performed in parallel with DSP program execution using DSP on-chip DMA controllers.

ADC and DAC data acquisition controllers can be software configured to operate in different data acquisition modes, which deliver outstanding flexibility and make *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM an ideal solution for virtually any high-performance DSP application with high-speed and high-accuracy multi-channel AD/DA, most typically instrumentation and power control applications.

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM also contains a set of on-board peripherals, which are controlled via host PIOX-16 interface. These on-board peripherals comprise of high-resolution sampling frequency generator and 4-bit general purpose digital I/O.

Host PIOX-16 interface provides access to on-board ADC, DAC, ADC FIFO, DAC FIFO and contains a set of control registers for data acquisition control, host interrupt selection, and for error processing.

External signal I/O

Connection of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM to external analog and digital I/O world is performed either via on-board I/O connectors or by means of external I/O board (*T/X-XIOB/PDAS1M*) (refer to fig.1-3 and to Appendix C).

T/X-XIOB/PDAS1M external I/O board converts *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* on-board I/O connectors into two groups: the 1st group contains analog I/O signals, whereas the 2nd group comprises of external clock and synchronization I/O pins and general purpose digital I/O pins.

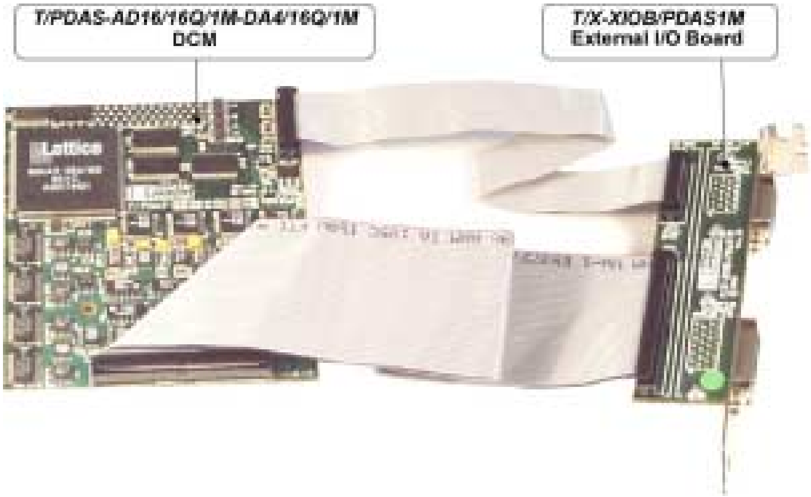

Fig. 1-3. T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM and *T/X-XIOB/PDAS1M* external I/O board.

T/X-XIOB/PDAS1M external I/O board installs either to the rear mounting bracket of host *TORNADO* DSP system in case *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM is installed onto *TORNADO* DSP system for PC (fig.1-4), or can install directly at the rear panel of host PC.

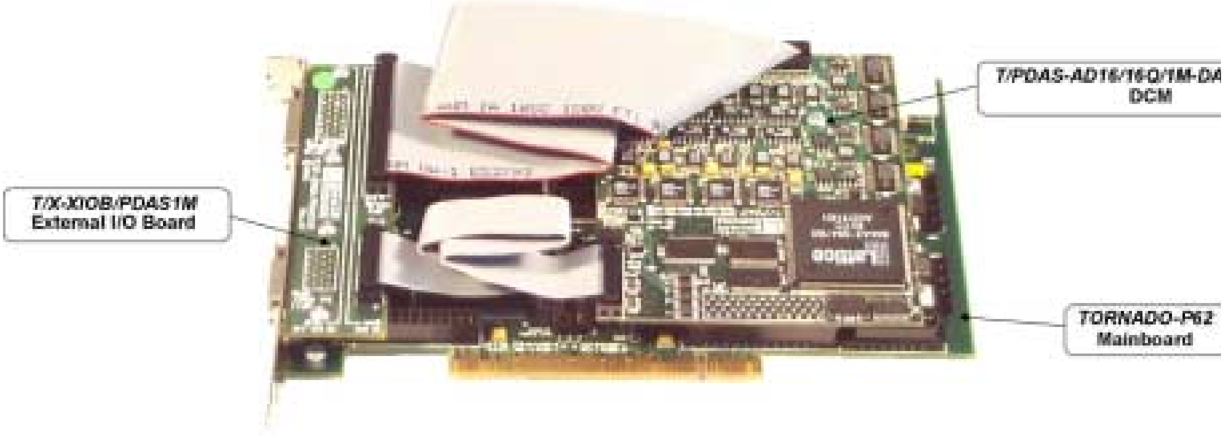

Fig. 1-4. *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM with *T/X-XIOB/PDAS1M* external I/O board installed onto *TORNADO-P62* PC plug-in DSP system.

Two external cable sets (*T/X-SIOCS/PDAS1M* and *T/X-AIOCS/PDAS1M*, refer to Appendix C for more details) come standard with *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM and provide direct connection to any external high-frequency signal sources and I/O synchronization using industry-standard connectors (fig.1-5). *T/X-SIOCS/PDAS1M* and *T/X-AIOCS/PDAS1M* external cable sets directly plug to the corresponding external I/O connectors at *T/X-XIOB/PDAS1M* external I/O board for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

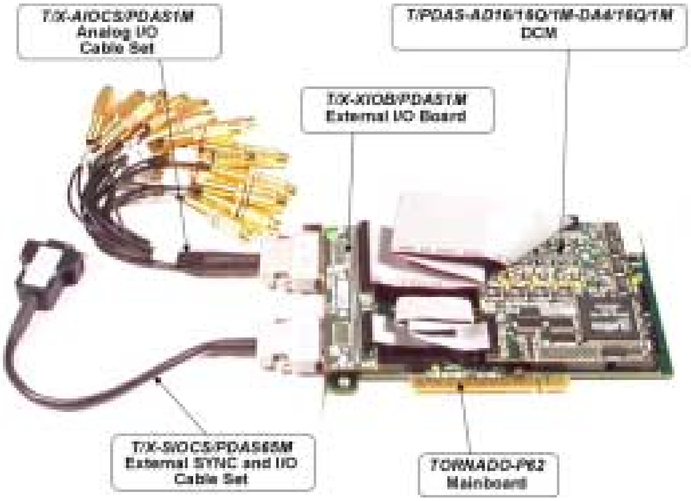

Fig. 1-5. *TORNADO-P62* DSP system with *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, *T/X-XIOB/PDAS1M* external I/O board and external cable sets.

Applications

T/PDAS-AD16/16Q/1M-DA4/16Q/1M PIOX-16 AD/DA DCM has been designed for multi-channel high-frequency and ultra-high accuracy analog and digital I/O instrumentation applications, as well as for other industrial and general signal processing applications, which assumes similar analog and digital signal I/O requirements and meets the specifications of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

1.2 Technical Specification

The following are detail technical specifications for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* PIOX-16 AD/DA DCM for *TORNADO* DSP systems/controllers.

CAUTION

Some of on-board components of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM may appear very hot during operation and can deliver skin sore in case of direct contact.

Although it is generally not required for normal operation of the *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, it is recommended to provide optional airflow over *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM board using optional fan/blower in case *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM is installed into the device enclosure with poor airflow conditions.

Parameter description	*parameter value*
analog inputs and ADC	
number of analog inputs	16 SE, or 8 DIFF, or combination of both
number of ADC channels	4 with synchronous sampling of all ADC
input impedance/capacitance for analog inputs	1 MOhm @ 75pF (*T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM only) 33 kOhm @ 75pF (with *T/X-XIOB/PDAS1M* external I/O board)
analog input range	± (5V ± 0.5%)
analog input signal over-voltage clipping voltage	± 6.5V
maximum over-voltage clipping current	± 100 mA total (for all clipped analog inputs)
analog input common voltage in DIFF mode	± 5V
ADC resolution	16 bits, no missing codes

Maximum ADC sampling frequency	
in HIGH SPEED mode	1 MHz
in NORMAL mode	800 kHz
Minimum ADC sampling frequency	
in HIGH SPEED mode	1 kHz
in NORMAL mode	-
A/D conversion time	0.75 µs (ADC high-speed mode) 1 µs (ADC normal speed mode)
ADC chip acquisition (sampling) time	0.25 µS
combined "ADC+ADCMUX" acquisition (sampling) time for analog input multiplexer and ADC chip	0.8 µS
SNR (F_{in}=20kHz, F_{in}=250kHz)	-88 dB (typ)
THD (F_{in}=20kHz, F_{in}=250kHz)	-98 dB (typ)
integral non-linearity	± 2.5 LSB (max)
gain error	± 0.5% full scale max
zero offset error	± 6 mV max
channel-to-channel isolation for neighbor channels (without *T/X-XIOB/PDAS1M* external I/O board)	-90db typ (R_{src} = 10 Ohm, F_{in} = 10 kHz) -76db typ (R_{src} = 10 kOhm, F_{in} = 10 kHz) -58db typ (R_{src} = 1 MOhm, F_{in} = 10 kHz)
duration of *standard ADC sampling cycle* for ADC asynchronous data acquisition mode	0.75 µs (ADC high-speed mode) 1 µs (ADC normal speed mode)
duration of *extended ADC sampling cycle* (ADC sampling cycle preceded by signal acquisition delay) for ADC asynchronous data acquisition mode	1.55 µs (ADC high-speed mode) 1.8 µs (ADC normal speed mode)
Scan cycle time for ADC synchronous data acquisition modes	1 µs (high-speed ADC mode without scan cycle delay) 2 µs (high-speed ADC mode with scan cycle delay) 1.25 µs (normal ADC mode without scan cycle delay) 2.25 µs (normal ADC mode with scan cycle delay)

analog outputs and DAC

number of DAC channels	4 with synchronous output sampling for all DAC
analog output range	± (5V ± 0.5%) @ 600 Ohm
output impedance	< 0.01 Ohm
minimum load impedance	600 Ohm

DAC resolution	16 bits
maximum sampling frequency	1 MHz
differential non-linearity	± 1.5 LSB max
integral non-linearity	± 1.5 LSB max
zero offset error	± 1.5 mV (max)
gain error	± 0.5% full scale max

ADC/DAC FIFO

ADC FIFO depth	256 Kwords
DAC FIFO depth	256 Kwords
programmable FIFO flags	PAE (partially empty) PAF (partially full)

ADC/DAC Data Acquisition Controllers

ADC/DAC data acquisition modes	ASYNCHRONOUS with direct ADC/DAC access SYNCHRONOUS with direct ADC/DAC access SYNCHRONOUS One-pass FIFO SYNCHRONOUS Pass-thru FIFO
ADC data download modes	direct ADC access ADC IRT with ADC direct read ADC IRT with ADC FIFO read
Time jitter for first ADC sampling event inside ADC sampling packet for all ADC synchronous data acquisition modes and for ADC sampling event for ADC asynchronous data acquisition mode with hardware defined sampling frequency source	± 10 ns (absolute maximum)
DAC data upload modes	direct DAC access DAC IRT with DAC direct load DAC IRT with DAC FIFO write
Duration of the DAC sampling cycle	100..120 ns
Time jitter for DAC output data load event for all DAC synchronous data acquisition modes and for DAC asynchronous data acquisition mode with DAC output load on hardware defined sampling frequency event	0 ns

Programmable Sampling Frequency Generator (PFG)

Programmable output frequency range	16 Hz .. 1 MHz

stability of reference clock frequency	± 50 PPM
PLL lock time to within 1% of output frequency	< 10 mS
period jitter	± 40 pS (MSV) ± 90 pS (absolute)

External Sampling Frequency (XFs) and Start Synchronization (ADC_XSSYNC_IN and DAC_XSSYNC_IN) inputs

maximum XFs input frequency	1 MHz
minimum duration of the '0' state of input signal at *XFs* pin of on-board JP3 connector, and of the '1' state of input signal at TM/XIO-0/1 pins of host PIOX-16 interface	200 nS
logical I/O level	3v/5v TTL
input impedance for XFs and XSYNC inputs	110 Ohm

general purpose digital I/O

number of programmable digital I/O bits	4 I/O bits with programmable direction
I/O signal level	3v/5v TTL
output load current	< 3.2 mA

Host PIOX-16 Interface

I/O ports	control registers I/O FIFO
access time	30 ns max (control register access) 60 ns max (ADC, DAC, FIFO access) 90ns max (DAC output data load)
PIOX-16 interrupt request inputs	software selectable from IRQ-0, IRQ-1, IRQ-2, IRQ-3

physical and power:

Dimensions	65 mm (2.55") x 85 mm (3.34")
power consumption via host PIOX I/F	+5v @0.7 A +12v @ 0.3 A -12v @ 0.1 A
external operating temperature	0°C .. +55°C
recommended optional fan/blower airflow (in case of poor airflow conditions)	>0.1 m^3/min

Chapter 2. System Architecture and Construction

This chapter contains detail technical description for architecture and construction of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* PIOX-16 AD/DA DCM.

2.1 System Architecture

System architecture and construction for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* PIOX-16 AD/DA DCM are presented at figures 2-1 and 2-2 correspondingly.

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM comprises of the following on-board architecture components:

- quad channel analog input section comprising of sixteen analog inputs (AIN-0..15) at JP2 analog I/O connector, sixteen analog input amplifiers (AIN-A0..15), four software configured either 4:1/SE single ended or 2:1/DIFF differential analog input multiplexers (AIN-MUX0..3), and four synchronously sampled 16-bit 1 Msps analog-to-digital converters (ADC0..3)
- ADC output data buffer (ADC-ODB) for direct read of ADC data via host PIOX-16 interface
- 256Kwords FIFO for buffering ADC output data streams (ADC-FIFO), which can be also read via host PIOX-16 interface
- quad-channel analog output section comprising of four synchronously sampled 16-bit 1 Msps digital-to-analog converters (DAC0..3), analog output amplifiers (AOUT-A0..3) and four analog outputs (AOUT-0..3) at JP2 analog I/O connector
- DAC input data buffer (DAC-IDB) for direct DAC data write via host PIOX-16 interface
- 256Kwords FIFO for buffering DAC input data streams (DAC-FIFO), which can be also written via host PIOX-16 interface
- high-resolution programmable sampling frequency generator (PFG)
- 4-bit general purpose programmable digital I/O (GPIO-0..3)
- data acquisition and control unit (DAQCU)
- external synchronization and I/O connector (JP3)
- host PIOX-16 interface controller and host PIOX-16 interface header (JP1) for installation onto *TORNADO* DSP systems/controllers.

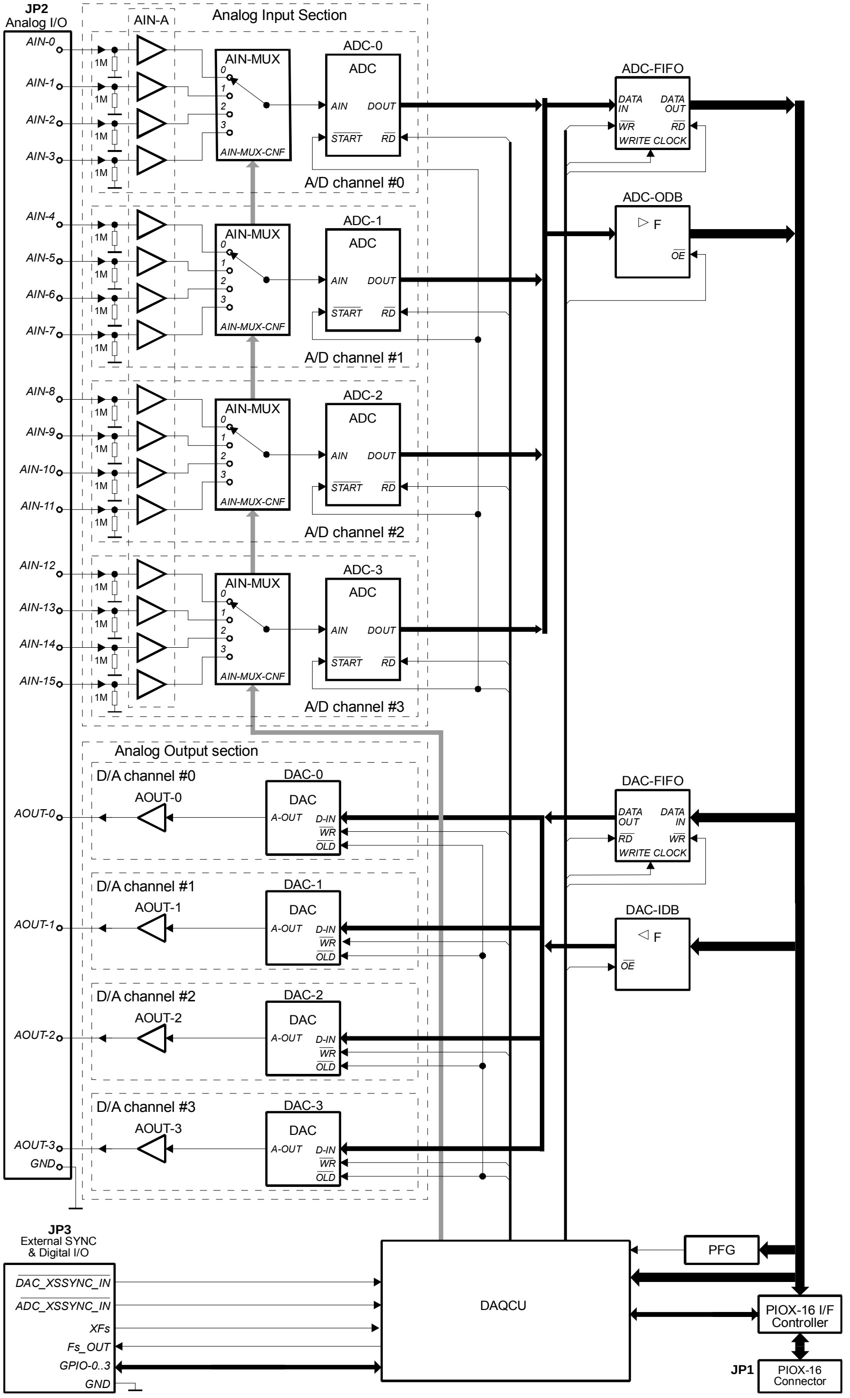

Fig.2-1. Block diagram of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

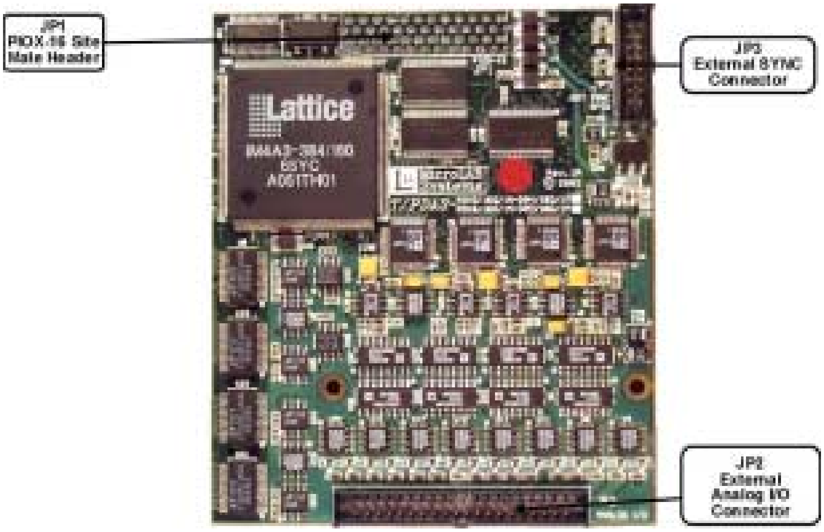

Fig.2-2. Construction of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

analog input section

Analog input section of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (fig.2-1) comprises of four identical ADC channels (#0..#3), and is designed for analog-to-digital conversion of analog signals from sixteen available analog inputs (AIN-0..15).

Each ADC channel can sample and perform A/D conversion of one of four assigned analog inputs, and all four ADC channels feature synchronous sampling of selected analog inputs. Each ADC channel comprises of the following components:

- four analog inputs (either of AIN-0..3, AIN-4..7, AIN-8..11, and AIN-12..15 correspondingly), which are available at JP2 analog I/O connector
- four analog input amplifiers (either of AIN-A0..3, AIN-A4..7, AIN-A8..11, and AIN-A12..15 correspondingly), one per each analog input
- software configured as either 4:1/SE single-ended or 2:1/DIFF differential analog input multiplexer (either of AIN-MUX0..3 correspondingly)
- 16-bit 1 Msps ADC chip (either of ADC-0..3 correspondingly).

For more details about analog input section of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM refer to section “Data Acquisition Control” later in this chapter.

ADC output data read path

Upon particular ADC data acquisition mode selected via ADC data acquisition controller, ADC output data can be either read directly via ADC output data buffers (ADC-ODB) by host PIOX-16 interface, or can be pushed into on-board high-density ADC FIFO, which can be further read via host PIOX-16 interface.

High-density on-board ADC FIFO can acquire any software configured number of ADC output data samples and is controlled by DAQCU and host PIOX-16 interface.

For more details about ADC FIFO and ADC data output path for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM refer to section "Data Acquisition Control" later in this chapter.

analog output section

Analog output section of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM comprises of four identical DAC channels (#0..#3), and is designed for digital-to-analog conversion of input data from either host PIOX-16 interface or from DAC FIFO to analog output signals (AOUT-0..3).

Each DAC channel can perform D/A conversion to one analog output, and all D/A channels feature synchronous DAC output data load (sampling). Each DAC channel comprises of the following components:

- 16-bit 1 Msps DAC chip (either of DAC-0..3 correspondingly)
- analog output amplifier (either of AOUT-A0..3 correspondingly).

For more details about analog output section of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM refer to section "Data Acquisition Control" later in this chapter.

DAC input data write path

Upon particular DAC data acquisition mode selected via DAC data acquisition controller, DAC input data can be either written directly via DAC input data buffers (DAC-IDB) by host PIOX-16 interface, or can be extracted from on-board high-density DAC FIFO, which must be previously written via host PIOX-16 interface.

High-density on-board DAC FIFO can acquire any software configured number of DAC output data samples and is controlled by DAQCU and host PIOX-16 interface.

For more details about DAC FIFO and DAC data input path for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM refer to section "Data Acquisition Control" later in this chapter.

Data acquisition and control unit (DAQCU), data acquisition controllers and IRT controllers

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM on-board data acquisition and control unit (DAQCU) provides individual accurate timing for ADC/DAC data acquisition, and performs ADC/DAC real-time data upload/download control. DAQCU is configured from host DSP software of *TORNADO* DSP system/controller.

DAQCU comprises of two independent and software configured ADC and DAC data acquisition controllers and two independent ADC and DAC interrupt retriggable transmission (IRT) controllers.

Each of ADC and DAC data acquisition controllers can run in asynchronous, synchronous with direct PIOX-16 access, synchronous FIFO one-pass, and synchronous FIFO pass-thru data acquisition modes. Different data acquisition modes feature different sampling time accuracy, different A/D multichannel support, and different ADC/DAC data read/write paths, thus providing outstanding flexibility to meet requirements of virtually any multichannel AD/DA instrumentation and similar applications.

ADC and DAC IRT controllers provide convenient way to download/upload real-time ADC/DAC data to/from DSP environment and have been designed to run in conjunction with host DSP on-chip DMA controllers in either of ADC/DAC data acquisition modes. ADC/DAC IRT controllers allow to significantly off-load DSP core from time consuming data transfer between *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM and DSP environment via PIOX-16 interface and leave enough time room for DSP core to perform actual digital signal processing of acquired real-time data.

For more details about operation of data acquisition controllers and IRT controllers for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM refer to section “Data Acquisition Control” later in this chapter.

A/D and D/A data acquisition start synchronization for synchronous DAQ modes

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM allows to start ADC and DAC data acquisition in synchronous data acquisition modes either immediately by setting the corresponding software flag, or on external synchronization event at the dedicated *DAC_XSYNCH_IN* and *ADC_XSYNC_IN* digital inputs, which are available via on-board JP3 external synchronization and I/O connector. External synchronization events can be selected either level or edge sensitive.

For more details about data acquisition synchronization for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM refer to section “Data Acquisition Control” later in this chapter.

sampling frequency

Sampling frequency is common for both ADC and DAC data acquisition controllers, and can be configured by host DSP software of host *TORNADO* DSP system/controller to source from either on-board high-resolution programmable sampling frequency generator (PFG), or from external sampling frequency input (XFs) from on-board JP3 connector, or from any of two timer outputs of host PIOX-16 interface (TM/XIO-0 and TM/XIO-1), which are typically the DSP on-chip timer outputs for all *TORNADO* DSP systems and controllers.

programmable sampling frequency generator (PFG)

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM provides on-board programmable high-resolution sampling frequency generator (PFG) in order to use it as the sampling frequency source for ADC and DAC data acquisition controllers.

On-board high-resolution PFG allows accurate setting of virtually any sampling frequency value within the 16Hz..1MHz frequency range. Sampling frequency, which value is outside of the PFG frequency range, can be supplied either from external sampling frequency input (XFs) of JP3 connector, or from PIOX-16 interface timer pins (TM/XIO-0 and TM/XIO-1).

For more details about PFG programming for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM refer to section “Data Acquisition Control” later in this chapter.

programmable general purpose I/O

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM provides four programmable general purpose I/O bits (GPIO-0..3), which are available at JP3 external synchronization connector. GPIO-0..3 I/O pins are 3v/5v TTL compatible and can be used as general purpose I/O for control and interfacing to external peripherals.

Direction and input/output data for GPIO-0..3 pins are individually programmable via host PIOX-16 interface. Active low condition at GPIO-0 and GPIO-1 I/O pins can be also used to generate interrupts to host PIOX-16 interface, which is required by multiple application to track external relay conditions.

For more details about GPIO of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM refer to section “Host PIOX-16 Interface” later in this chapter.

host PIOX-16 interface and controller

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM installs as PIOX-16 AD/DA DCM onto host *TORNADO* DSP system/controller using host 16-bit PIOX-16 interface (JP1 connector).

Host PIOX-16 interface controller provides control for access to on-board ADC, DAC, ADC FIFO, DAC FIFO, PFG and GPIO, and contains a set of control registers for DAQCU control.

For more details about host PIOX-16 interface of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM refer to section "Host PIOX-16 Interface" later in this chapter.

host PIOX-16 interrupts

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM can generate four interrupts requests to host PIOX-16 interface, which can be individually configured from host DSP software.

Each of host PIOX-16 interrupt requests feature individual software configured interrupt enable, interrupt polarity, and can source from a variety of data acquisition events, error events, external GPIO-0/1 transitions, IRT events and conditions, and FIFO flags.

For more details about host PIOX-16 interrupt requests for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM refer to sections "Host PIOX-16 Interface" and "Data Acquisition Control" later in this chapter.

2.2 Host PIOX-16 Interface

Host 16-bit PIOX-16 interface of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM provides access from host *TORNADO* DSP system/controller to on-board ADC, DAC, ADC FIFO, DAC FIFO, PFG, GPIO and DAQCU control registers.

On-board JP1 connector of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (fig.A-1) is used to install into PIOX-16 site of host *TORNADO* DSP system/controller.

host PIOX-16 interface address map

Host PIOX-16 interface address map for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM comprises of control register area, ADC/DAC direct data area, and ADC/DAC FIFO data area.

Table 2-1 provides host PIOX-16 interface address map for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

Table 2-1. Host PIOX-16 interface address map.

address area and register name	value on PIOX-16 reset condition	access mode	address (note 3)	Reference information
Control Register area: *DAQ_CNTR1_RG* registers (ADC/DAC data acquisition modes)	00H	r/w	*BA*+0000H (bits D0..D7)	table 2-2
Control Register area: *DAQ_CNTR2_RG* register (real-time data acquisition control)	00H	r/w	*BA*+0001H (bits D0..D7)	table 2-3
Control Register area: *DAQ_CNTR3_RG* register (data acquisition termination/IRT flags selector)	00H	r/w	*BA*+0002H (bits D0..D7)	table 2-4
Control Register area: *DAQ_SYNC_RG* register (sampling frequency selector and data acquisition synchronization options)	00H	r/w	*BA*+0003H (bits D0..D7)	table 2-5
Control Register area: *ADC_CNF_RG* register (individual ADC enable control	8FH	r/w	*BA*+0004H (bits D0..D7)	table 2-6
Control Register area: *ADC_IMUX_CNF1_RG* register (ADC input MUX configuration for asynchronous ADC data acquisition mode and for scan cycles #0/#1 for synchronous ADC data acquisition modes)	00H	r/w	*BA*+0006H (bits D0..D7)	tables 2-8a and 2-8b
Control Register area: *ADC_IMUX_CNF2_RG* register (ADC input MUX configuration for scan cycles #2/#3 for synchronous ADC data acquisition modes)	80H	r/w	*BA*+0007H (bits D0..D7)	tables 2-8a and 2-8b
Control Register area: *DAC_CNF_RG* register (individual DAC enable control	8FH	r/w	*BA*+0008H (bits D0..D7)	table 2-7
Control Register area: *FIFO_STAT_RG* register (ADC/DAC FIFO status flags)	00H	r	*BA*+0009H (bits D0..D7)	table 2-9
Control Register area: *ERR_STAT_RG* register (read-only) (data acquisition error status) *ERR_CLR_RG* register (write-only) (data acquisition error clear)	00H -	r w	*BA*+000AH (bits D0..D7)	table 2-10

Control Register area: *HIRQ0_SEL_RG* register (host PIOX-16 IRQ-0 interrupt control)	00H	r/w	*BA*+0010H (bits D0..D7)	tables 2-15, 2-16
Control Register area: *HIRQ1_SEL_RG* register (host PIOX-16 IRQ-1 interrupt control)	00H	r/w	*BA*+0011H (bits D0..D7)	tables 2-15, 2-16
Control Register area: *HIRQ2_SEL_RG* register (host PIOX-16 IRQ-2 interrupt control)	00H	r/w	*BA*+0012H (bits D0..D7)	tables 2-15, 2-16
Control Register area: *HIRQ3_SEL_RG* register (host PIOX-16 IRQ-3 interrupt control)	00H	r/w	*BA*+0013H (bits D0..D7)	tables 2-15, 2-16
Control Register area: *XIM_ERR_RG* register (expansion error interrupt mask)	00H	r/w	*BA*+0016H (bits D0..D7)	table 2-17
Control Register area: *PFG_CNTR1_RG* register (PFG control: V0..V7)	38H	r/w	*BA*+0018H (bits D0..D7)	table 2-13
Control Register area: *PFG_CNTR2_RG* register (PFG control: V8, R0..R6)	60H	r/w	*BA*+0019H (bits D0..D7)	table 2-13
Control Register area: *PFG_CNTR3_RG* register (PFG control: S0..S2, X0..X1)	00H	r/w	*BA*+001AH (bits D0..D7)	table 2-13
Control Register area: *GPIO_DATA_RG* register (GPIO I/O data)	-	r/w	*BA*+001CH (bits D0..D7)	table 2-14b
Control Register area: *GPIO_DIR_RG* register (GPIO direction control)	00H	r/w	*BA*+001DH (bits D0..D7)	table 2-14a
FIFO area: *ADC_FIFO_DATA_RG* register (ADC FIFO output data (LSW))	-	r	*BA*+8000H (bits D0..D15)	-
FIFO area: *ADC_FIFO_DATA_MSW_RG* register (ADC FIFO output data (MSW))	-	r	*BA*+000CH (bits D0..D7 only)	table 2-11
FIFO area: *DAC_FIFO_DATA_RG* register (DAC FIFO input data)	-	w	*BA*+8001H (bits D0..D15)	-
FIFO area: *ADC_FIFO_SIN_RG* register (PAF/PAE flags configuration data for ADC FIFO)	-	w	*BA*+8002H (bit D0)	table 2-12

FIFO area: *DAC_FIFO_SIN_RG* register (PAF/PAE flags configuration data for DAC FIFO)	-	w	*BA*+8003H (bit D0)	table 2-12
Control Register area: *ADC_DAQ_RESET_RG* register (Reset for ADC data acquisition controller)	-	w	*BA*+8004H (written data ignored)	-
Control Register area: *DAC_DAQ_RESET_RG* register (Reset for DAC data acquisition controller)	-	w	*BA*+8005H (written data ignored)	-
Control Register area: *ADDA_DAQ_RESET_RG* register (Simultaneous reset for both ADC and DAC data acquisition controllers)	-	w	*BA*+800AH (written data ignored)	-
ADC/DAC Direct Data area: *ADC0_DATA_RG* register (ADC-0 output data)	-	r	*BA*+8010H (bits D0..D15)	-
ADC/DAC Direct Data area: *ADC1_DATA_RG* register (ADC-1 output data)	-	r	*BA*+8011H (bits D0..D15)	-
ADC/DAC Direct Data area: *ADC2_DATA_RG* register (ADC-2 output data)	-	r	*BA*+8012H (bits D0..D15)	-
ADC/DAC Direct Data area: *ADC3_DATA_RG* register (ADC-3 output data)	-	r	*BA*+8013H (bits D0..D15)	-
ADC/DAC Direct Data area: *DAC0_DATA_RG* register (DAC-0 input data)	-	w	*BA*+8014H (bits D0..D15)	-
ADC/DAC Direct Data area: *DAC1_DATA_RG* register (DAC-1 input data)	-	w	*BA*+8015H (bits D0..D15)	-
ADC/DAC Direct Data area: *DAC2_DATA_RG* register (DAC-2 input data)	-	w	*BA*+8016H (bits D0..D15)	-
ADC/DAC Direct Data area: *DAC3_DATA_RG* register (DAC-3 input data)	-	w	*BA*+8017H (bits D0..D15)	-
ADC/DAC Direct Data area: *ADC_IRT_DATA_RG* register (ADC IRT output data)	-	r	*BA*+8018H (bits D0..D15)	-

ADC/DAC Direct Data area: *DAC_IRT_DATA_RG* register (DAC IRT input data)	-	w	*BA*+8019H (bits D0..D15)	-
Control Register area: *ADC_START_RG* register (ADC DAQ start command)	-	w	*BA*+800CH (written data ignored)	-
Control Register area: *DAC_OUTPUT_LD_RG* register (DAC chip output load command)	-	w	*BA*+800DH (written data ignored)	-
Control Register area: *DADC_START_DAC_OUTPUT_LD_RG* register (Simultaneous ADC data acquisition start and DAC chip output load command)	-	w	*BA*+800EH (written data ignored)	-

Notes:

1. Access modes: *r* - read only, *w* - write only, *r/w* - read/write.
2. '*BA*' denotes base address for host PIOX-16 interface within the DSP address map of host *TORNADO* DSP system/controller.
3. I/O registers relative address values are specified at the data word boundaries for *TORNADO* DSP system/controllers with TMS320C3x and TMS320C5xxx DSP. For *TORNADO* DSP system/controllers with TMS320C6xxx DSP these relative addresses shall be multiplied either by x4, or by x2 (depending upon the particular *TORNADO* DSP system/controller type) in order to meet the byte addressing for TMS320C6xxx DSP.

CAUTION

DSP software of host *TORNADO* DSP system/controller must perform access to *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM referenced to the corresponding PIOX-16 interface base address, which is specific for particular *TORNADO* DSP system/controller (refer to documentation for your *TORNADO* DSP system/control for more details about addressing PIOX-16 interface area).

Register relative addresses specified in table 2-1 for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM are true for *TORNADO* DSP systems/controllers with TMS320C3x and TMS320C5xxx DSP with allocation of PIOX-16 data words with +1 address increment.

For *TORNADO* DSP systems/controllers with TMS320C6xxx DSP, which allocate PIOX-16 data words with either +2 or +4 address increment, specified register relative addresses in table 2-1 for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM shall be multiplied by either x2 or x4 factor correspondingly (refer to documentation for your *TORNADO* DSP system/control for more details about addressing PIOX-16 interface area).

CAUTION

When accessing Control Register area of host PIOX-16 interface of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM from host *TORNADO* DSP system/controller, only data bits D0..D7 are valid.

When accessing ADC/DAC direct data and ADC/DAC FIFO data registers of host PIOX-16 interface of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM from host *TORNADO* DSP system/controller, all bits D0..D15 are valid.

CAUTION

Reset signal for on-board PIOX-16 interface of host *TORNADO* DSP system/controller must be released prior communication with *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM in order to take DCM out of the reset state condition, otherwise all control registers will stay in default reset state and written data will be ignored.

data acquisition control registers

On-board DAQCU data acquisition and control unit, which performs ADC/DAC data acquisition and ADC/DAC IRT control, can be configured by host DSP software via a set of the following control registers:

- *DAQ_CNTR1_RG* register, which is used to select ADC/DAC data acquisition modes and options
- *DAQ_CNTR2_RG* register, which is used for real-time control of ADC/DAC data acquisition process during synchronous ADC/DAC data acquisition modes
- *DAQ_CNTR3_RG* register, which is used to select ADC/DAC data acquisition termination flags and interrupt retriggable transmission (IRT) flags for ADC/DAQ synchronous FIFO data acquisition modes
- *DAQ_SYNC_RG* register, which is used to select sampling frequency source, ADC/DAQ data acquisition start synchronization mode, external synchronization mode, and ADC timing options
- *ADC_CNF_RG* register, which is used to select ADC channels, which will take part in ADC data acquisition
- *DAC_CNF_RG* register, which is used to select DAC channels, which will take part in DAC data acquisition
- *ADCQ_IMUX_CNF1_RG* and *ADCQ_IMUX_CNF2_RG* registers, which are used to configure ADC input multiplexers for all ADC channels
- *ADC_DAQ_RESET_RG*, *DAC_DAQ_RESET_RG* and *ADDA_DAQ_RESET_RG* registers, which are used to reset ADC and DAC data acquisition controllers
- *ADC_START_RG*, *DAC_OUTPUT_LD_RG* and *ADC_START_DAC_OUTPUT_LD_RG* registers, which are used to start A/D conversion in ADC asynchronous data acquisition mode and to load DAC chip outputs in DAC asynchronous data acquisition mode.

Data acquisition control registers details are presented below in this section. For details about ADC and DAC data acquisition control and timing refer to section “Data Acquisition Control” later in this chapter, whereas this subsection will describe details for data acquisition control registers.

DAQ_CNTR1_RG control register

DAQ_CNTR1_RG register must be used to set ADC and DAC data acquisition mode and options. *DAQ_CNTR1_RG* register is available for read/write and is allocated at the least significant byte of 16-bit data word of host PIOX-16 interface.

DAQ_CNTR1_RG register (r/w)

X	*DAC_ASYNC_DAQ_OUTPUT_LD_MODE-1* *(r/w, 0+)*	*DAC_ASYNC_DAQ_OUTPUT_LD_MODE-0* *(r/w, 0+)*	*0*	*ADC_ASYNC_DAQ__FS_EN* *(r/w, 0+)*	*DAC_DAQ_MODE-1* *(r/w, 0+)*	*DAC_DAQ_MODE-0* *(r/w, 0+)*	*ADC_DAQ_MODE-1* *(r/w, 0+)*	*ADC_DAQ_MODE-0* *(r/w, 0+)*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1	bit-0

Table 2-2 provides details about *DAQ_CNTR1_RG* register bits.

Table 2-2. DAQ_CNTR1_RG register bits.

register bits	access mode	value on PIOX-16 reset	Description
{*ADC_DAQ_MODE-1, ADC_DAQ_MODE-0*}	r/w	{0,0}	Define ADC data acquisition mode for ADC data acquisition controller. Refer to section "Data Acquisition Control" later in this chapter for more details about operation of ADC data acquisition controller. {*ADC_DAQ_MODE*-[1:0]}={0,0} value corresponds to *ADC asynchronous data acquisition mode* (*ADC-ASYNC-DAQ*) and *ADC FIFO configuration mode*, which is used for software initialized A/D conversions without automatically generated scan cycles and to configure ADC FIFO PAE/PAF flags offset values. In this mode, ADC data must be read directly via host PIOX-16 interface, and A/D conversion can be either initialized by DSP software write to either of *ADC_START_RG* and *ADC_START_DAC_OUTPUT_LD_RG* registers, or can be started on sampling frequency event as defined via bit *ADC_ASYNC_DAQ_FS_EN* bit of *DAQ_CNTR1_RG* register. {*ADC_DAQ_MODE*-[1:0]}={0,1} value corresponds to *ADC synchronous data acquisition mode with direct ADC data read via host PIOX-16 interface* (*ADC-SYNC-PX-DAQ*). In this mode, ADC data must be read continuously directly via host PIOX-16 interface. ADC data acquisition process in this mode features accurate ADC sampling timing on sampling frequency event with automatically generated scan cycles. External data acquisition start synchronization is supported. {*ADC_DAQ_MODE*-[1:0]}={1,0} value corresponds to *ADC synchronous FIFO one-pass data acquisition mode one-pass data acquisition mode (ADC-SYNC-FIFO-OPM-DAQ)*. This ADC data acquisition mode is used to acquire and store any predefined number of ADC data samples in ADC FIFO and to terminate automatically. Direct read of ADC data is not supported. ADC data acquisition process in this mode features accurate ADC sampling timing on sampling frequency event with automatically generated scan cycles. External data acquisition start synchronization is supported. {*ADC_DAQ_MODE*-[1:0]}={1,1} value corresponds to *ADC synchronous FIFO pass-through data acquisition mode (ADC-SYNC-FIFO-PTM-DAQ)*. This ADC data acquisition mode is used to continuously acquire and store ADC data samples in ADC FIFO. Direct read of ADC data is not supported. ADC data acquisition process in this mode feature accurate ADC sampling timing on sampling frequency event with automatically generated scan cycles. External data acquisition start synchronization is supported.

{*DAC_DAQ_MODE-1, DAC_DAQ_MODE-0*}	r/w	{0,0}	Define DAC data acquisition mode for DAC data acquisition controller. Refer to section "Data Acquisition Control" later in this chapter for more details about operation of DAC data acquisition controller. {*DAC_DAQ_MODE*-[1:0]}={0,0} value corresponds to *DAC asynchronous data acquisition mode* (*DAC-ASYNC-DAQ*) and *DAC FIFO configuration mode*, which is used for software initialized DAC output data load and to configure DAC FIFO PAE/PAF flags offset values. In this mode, DAC data must be written directly via host PIOX-16 interface, and DAC output data load can be either transparent, or initialized by DSP software on host DSP write to either of *DAC_OUTPUT_LD_RG* and *ADC_START_DAC_OUTPUT_LD_RG* registers, or can be performed on sampling frequency event, or performed at the end of DAC IRT data transfer cycle as defined by bits {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE*-[1:0] } of *DAQ_CNTR1_RG* register. {*DAC_DAQ_MODE*-[1:0]} ={0,1} value corresponds to *DAC synchronous data acquisition mode with direct DAC data write via host PIOX-16 interface* (*DAC-SYNC-PX-DAQ*). In this mode, DAC data must be written continuously directly via host PIOX-16 interface. DAC data acquisition process in this mode features accurate DAC output data load timing on sampling frequency event. External data acquisition start synchronization is supported. {*DAC_DAQ_MODE*-[1:0]} ={1,0} value corresponds to *DAC synchronous FIFO one-pass data acquisition mode one-pass data acquisition mode (DAC-SYNC-FIFO-OPM-DAQ)*. This DAC data acquisition mode is used to read any predefined number of DAC data samples from DAC FIFO and to terminate automatically. Direct data write to DAC is not supported. DAC data acquisition process in this mode features accurate DAC output data load timing on sampling frequency event. External data acquisition start synchronization is supported. {*DAC_DAQ_MODE*-[1:0]} ={1,1} value corresponds to *DAC synchronous FIFO pass-through data acquisition mode (DAC-SYNC-FIFO-PTM-DAQ)*. This DAC data acquisition mode is used to continuously extract DAC data samples from DAC FIFO. Direct data write to DAC is not supported. DAC data acquisition process in this mode feature accurate DAC output data load timing on sampling frequency event. External data acquisition start synchronization is supported.

ADC_ASYNC_DAQ_ FS_EN	r/w	0	Enable control for A/D conversion start on hardware defined sampling frequency event for to ADC asynchronous data acquisition mode (*ADC-ASYNC-DAQ*). This bit is ignored for all ADC synchronous data acquisition modes. *ADC_ASYNC_DAQ_FS_EN*=0 corresponds to disabled A/D conversion start on hardware defined sampling frequency event for ADC asynchronous data acquisition mode. A/D conversion start can be initialized by DSP software only by writing to *ADC_START_RG* register. This mode is useful when there is no sampling frequency and A/D conversion must start on DSP software defined events. ADC sampling event time jitter is completely defined by DSP software. *ADC_ASYNC_DAQ_FS_EN*=1 corresponds to enabled A/D conversion start on hardware defined sampling frequency event (selected by {*FS_SEL-1, FS_SEL-0*} bits of *DAQ_SYNC_RG* control register) for ADC asynchronous data acquisition mode. This allows extremely low time jitter for ADC sampling event and minimizes signal spectrum distortion. A/D conversion can still be initialized by DSP software by writing to *ADC_START_RG* register. This mode must be used when low signal spectrum distortion is required in ADC asynchronous data acquisition mode
{*DAC_ASYNC_DAQ_ OUTPUT_LD_MODE-1, DAC_ASYNC_DAQ_ OUTPUT_LD_MODE-0*}	r/w	{0,0}	Defines DAC output data load mode for DAC asynchronous data acquisition mode. These bits are ignored for all DAC synchronous data acquisition modes. {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE*-[1:0]}={0,0} value corresponds to *transparent DAC output data load mode* for all DAC-0..3 chips. In this mode, analog output of each DAC chip is updated immediately after DAC data is being written from host PIOX-16 interface. In this mode, time jitter for DAC output data load event is defined by DSP software. {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE*-[1:0]}={0,1} value corresponds to *DAC output data load on software defined event mode* , i.e. common DAC output data load signal for all DAC-0..3 chips is generated on host DSP software write to either of *DAC_OUTPUT_LD_RG* and *ADC_START_DAC_OUTPUT_LD_RG* registers. This mode provides synchronous sampling for all DAC outputs. In this mode, time jitter for DAC output data load event is defined by DSP software. {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE*-[1:0]}={1,0} value corresponds to *DAC output data load on hardware defined sampling frequency event mode*. This mode provides true synchronous sampling for all DAC outputs on hardware defined sampling frequency events. Sampling frequency source is selected via bits {*Fs_SEL-[1:0]*} of *DAQ_SYNC_RG* register. This mode provides most accurate DAC output data load timing without time jitter for DAC output sampling event, which results in minimum DAC output signal spectrum distortion. {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE*-[1:0]}={1,1} value corresponds to *DAC output data load at the end of DAC IRT data transfer mode*, i.e. immediately after DAC data has been written to all DAC chips. This mode provides synchronous sampling for all DAC outputs. This mode does not guarantee maximum time jitter for DAC output data load event.

Note:
1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

DAQ_CNTR2_RG control register

Except for the *ADC_RDY* status bit, all *DAQ_CNTR2_RG* register must be used to control and get status information for ADC/DAC data acquisition process for any of ADC/DAC synchronous data acquisition modes (ADC and DAC synchronous data acquisition processes correspondingly).

DAQ_CNTR2_RG register is available for read/write and is allocated at the least significant byte of 16-bit data word of host PIOX-16 interface.

***DAQ_CNTR2_RG* register (r/w)**

x	*DAC_SYNC_ DAQ_END (r,0+)* *DAC_SYNC_ DAQ_ABORT (w)*	*ADC_SYNC_ DAQ_END (r,0+)* *ADC_SYNC_ DAQ_ABORT (w)*	*DAC_SYNC_ DAQ_SSYNC_OK (r,0+)*	*ADC_SYNC_ DAQ_SSYNC_OK (r,0+)*	*ADC_SYNC_ DAQ_EOS (r)*	*ADC_RDY (r)*	*DAC_SYNC_ DAQ_RUN (r/w, 0+)*	*ADC_SYNC_ DAQ_RUN (r/w, 0+)*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1	bit-0

Table 2-3 provides details about *DAQ_CNTR2_RG* register bits.

Table 2-3. DAQ_CNTR2_RG register bits.

register bits	**access mode**	**default value on PIOX-16 reset**	**Description**
ADC_SYNC_ DAQ_RUN	R/w	0	Initializes ADC data acquisition process for any of ADC synchronous data acquisition modes (ADC synchronous data acquisition process). Refer to section "Data Acquisition Control" later in this chapter for more details about ADC data acquisition controller. Reading *ADC_SYNC_DAQ_RUN* =0 corresponds to no currently active ADC synchronous data acquisition process. Reading *ADC_SYNC_DAQ_RUN* =1 corresponds to currently active ADC synchronous data acquisition process. Writing '0' to *ADC_SYNC_DAQ_RUN* bit has no effect. Writing '1' to *ADC_SYNC_DAQ_RUN* bit while *ADC_SYNC_DAQ_RUN*=0 will initialize ADC synchronous data acquisition process. Writing '1' to *ADC_SYNC_DAQ_RUN* bit while *ADC_SYNC_DAQ_RUN*=1 has no effect. ADC synchronous data acquisition process can be aborted by host DSP software by either setting bit *ADC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register to the '1' state, or by resetting ADC data acquisition controller (write to either *ADC_DAQ_RESET_RG* or to *ADDA_DAQ_RESET_RG* registers), or by setting ADC asynchronous data acquisition mode. For ADC synchronous FIFO one-pass data acquisition mode (*ADC-SYNC-FIFO-OPM-DAQ*), ADC synchronous data acquisition process can also terminate normally on the ADC synchronous data acquisition termination flag event in accordance with bit *ADC_SYNC_DAQ_TF_SEL* bit of *DAQ_CNTR3_RG* register.

DAC_SYNC_ DAQ_RUN	R/w	0	Initializes DAC data acquisition process for any of DAC synchronous data acquisition modes (DAC synchronous data acquisition process). Refer to section “Data Acquisition Control” later in this chapter for more details about DAC data acquisition controller. Reading *DAC_SYNC_DAQ_RUN* =0 corresponds to no currently active DAC synchronous data acquisition process. Reading *DAC_SYNC_DAQ_RUN* =1 corresponds to currently active DAC synchronous data acquisition process. Writing ‘0’ to *DAC_SYNC_DAQ_RUN* bit has no effect. Writing ‘1’ to *DAC_SYNC_DAQ_RUN* bit while *DAC_SYNC_DAQ_RUN*=0 will initialize DAC synchronous data acquisition process. Writing ‘1’ to *DAC_SYNC_DAQ_RUN* bit while *DAC_SYNC_DAQ_RUN*=1 has no effect. DAC synchronous data acquisition process can be aborted by host DSP software by either setting bit *DAC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register to the ‘1’ state, or by resetting DAC data acquisition controller (write to either *DAC_DAQ_RESET_RG* or to *ADDA_DAQ_RESET_RG* registers), or by setting DAC asynchronous data acquisition mode. For DAC synchronous FIFO one-pass data acquisition mode (*DAC-SYNC-FIFO-OPM-DAQ*), DAC synchronous data acquisition process can be also terminated normally on the DAC synchronous data acquisition termination flag event in accordance with bit *DAC_SYNC_DAQ_TF_SEL* bit of *DAQ_CNTR3_RG* register.
ADC_RDY	r	0	Indicates ADC data ready condition for all ADC data acquisition modes. Refer to section “Data Acquisition Control” later in this chapter for more details about ADC data acquisition controller. *ADC_RDY*=0 denotes that ADC data is not yet valid, i.e. A/D conversion is in progress. This state is also set when ADC data acquisition controller has been reset. *ADC_RDY*=1 denotes that A/D conversion completes and valid ADC data is available. This condition can be used to generate interrupt request to host PIOX-16 interface.
ADC_SYNC_DAQ_ EOS	r	0	Indicates whether available ADC data belong to the last scan cycle inside sampling packet during ADC synchronous data acquisition process. This bit is updated when ADC data comes valid, i.e. when *ADC_RDY* bit of *DAQ_CNTR2_RG* registers is set to the ‘1’ state. Although this bit is valid for all ADC synchronous data acquisition modes, it is actually useful for *ADC-SYNC-PX-DAQ* data acquisition mode only when ADC data can be read directly via host PIOX-16 interface. This bit is not valid for ADC asynchronous data acquisition mode. Refer to section “Data Acquisition Control” later in this chapter for more details about ADC data acquisition controller. *ADC_SYNC_DAQ_EOS*=0 denotes that currently available ADC data does not belong to the last scan cycle of ADC sampling packet. *ADC_SYNC_DAQ_EOS*=1 denotes that currently available ADC data belongs to the last scan cycle of ADC sampling packet.

ADC_SYNC_DAQ_SSYNC_OK	r	0	Returns status information for start synchronization detector for ADC synchronous data acquisition process. *ADC_SYNC_DAQ_SSYNC_OK* bit is hold in the '0' state during ADC asynchronous data acquisition mode and when ADC synchronous data acquisition process is not active (in case *ADC_SYNC_DAQ_RUN* bit of *DAQ_CNTR2_RG* register is in the '0' state). Refer to section "Data Acquisition Control" later in this chapter for more details about ADC data acquisition controller. *ADC_SYNC_DAQ_SSYNC_OK*=0 denotes that either ADC synchronous data acquisition process is not active (in case *ADC_SYNC_DAQ_RUN* bit of *DAQ_CNTR2_RG* register is in the '0' state), or start synchronization event has not yet been detected during currently active ADC synchronous data acquisition process (in case *ADC_SYNC_DAQ_RUN* bit of *DAQ_CNTR2_RG* register is in the '1' state). *ADC_SYNC_DAQ_SSYNC_OK* =1 corresponds to already detected start synchronization event during currently active ADC synchronous acquisition process (in case *ADC_SYNC_DAQ_RUN* bit of *DAQ_CNTR2_RG* register is in the '1' state). This condition can be used to generate interrupt request to host PIOX-16 interface.
DAC_SYNC_DAQ_SSYNC_OK	r	0	Returns status information for start synchronization detector for DAC synchronous data acquisition process. *DAC_SYNC_DAQ_SSYNC_OK* bit is hold in the '0' state during DAC asynchronous data acquisition mode and when DAC synchronous data acquisition process is not active (in case *DAC_SYNC_DAQ_RUN* bit of *DAQ_CNTR2_RG* register is in the '0' state). Refer to section "Data Acquisition Control" later in this chapter for more details about ADC data acquisition controller. *DAC_SYNC_DAQ_SSYNC_OK*=0 denotes that either DAC synchronous data acquisition process is not active (in case *DAC_SYNC_DAQ_RUN* bit of *DAQ_CNTR2_RG* register is in the '0' state), or start synchronization event has not yet been detected during currently active DAC synchronous data acquisition process (in case *DAC_SYNC_DAQ_RUN* bit of *DAQ_CNTR2_RG* register is in the '1' state). *DAC_SYNC_DAQ_SSYNC_OK* =1 corresponds to already detected start synchronization event during currently active DAC synchronous acquisition process (in case *DAC_SYNC_DAQ_RUN* bit of *DAQ_CNTR2_RG* register is in the '1' state). This condition can be used to generate interrupt request to host PIOX-16 interface.

ADC_SYNC_DAQ_ ABORT *ADC_SYNC_DAQ_ END*	w r	- 0	During writes this bit can be used to abort currently active ADC synchronous data acquisition process. During reads this bit returns normal termination status of ADC synchronous data acquisition process for ADC synchronous FIFO one-pass data acquisition mode (*ADC-SYNC-FIFO-OPM-DAQ*). Refer to section "Data Acquisition Control" later in this chapter for more details about operation of ADC data acquisition controller. Writing *ADC_SYNC_DAQ_ABORT* =0 has no effect on ADC synchronous data acquisition process. Writing *ADC_SYNC_DAQ_ABORT* =1 will abort currently active ADC synchronous data acquisition process, and will clear bits *ADC_SYNC_DAQ_RUN* and *ADC_SYNC_DAQ_END* of *DAQ_CNTR2_RG* register, and bit *ADC_IRT_ERR* of *ERR_STAT_RG* register. Reading *ADC_SYNC_DAQ_END* =0 denotes that either ADC synchronous data acquisition process for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode has not been terminated normally yet on ADC synchronous data acquisition termination flag event in accordance with bit *ADC_SYNC_DAQ_TF_SEL* bit of *DAQ_CNTR3_RG* register, or ADC synchronous data acquisition process has not been initialized yet, or ADC synchronous data acquisition process has been aborted. *ADC_SYNC_DAQ_END* bit will stay in the '0' state during *ADC-ASYNC-DAQ*, *ADC-SYNC-PX-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes. Reading *ADC_SYC_DAQ_END* =1 corresponds to normally terminated ADC synchronous data acquisition process on ADC synchronous data acquisition termination event for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode in accordance with bit *ADC_SYNC_DAQ_TF_SEL* of *DAQ_CNTR3_RG* register. This condition can be used to generate interrupt request to host PIOX-16 interface. *ADC_SYNC_DAQ_END* bit will reset to the '0' state on start of ADC synchronous data acquisition process (by setting bit *ADC_SYNC_DAQ_RUN* of *DAQ_CNTR2_RG* register to the '1' state), when '1' is written to the *ADC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register, when ADC data acquisition controller is reset (by write to either *ADC_DAQ_RESET_RG* or to *ADDA_RESET_RG* registers), or when ADC asynchronous data acquisition mode is selected.

DAC_SYNC_DAQ_ ABORT *DAC_SYNC_DAQ_ END*	w r	- 0	During writes this bit can be used to abort currently active DAC synchronous data acquisition process. During reads this bit returns normal termination status of ADC synchronous data acquisition process for DAC synchronous FIFO one-pass data acquisition mode (*DAC-SYNC-FIFO-OPM-DAQ*). Refer to section "Data Acquisition Control" later in this chapter for more details about operation of ADC data acquisition controller. Writing *DAC_SYNC_DAQ_ABORT* =0 has no effect on DAC synchronous data acquisition process. Writing *DAC_SYNC_DAQ_ABORT* =1 will abort currently active DAC synchronous data acquisition process, and will clear bits *DAC_SYNC_DAQ_RUN* and *DAC_SYNC_DAQ_END* of *DAQ_CNTR2_RG* register, and bit *DAC_IRT_ERR* of *ERR_STAT_RG* register. Reading *DAC_SYNC_DAQ_END* =0 denotes that either DAC synchronous data acquisition process for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode has not been terminated normally yet on DAC synchronous data acquisition termination flag event in accordance with bit *DAC_SYNC_DAQ_TF_SEL* bit of *DAQ_CNTR3_RG* register, or DAC synchronous data acquisition process has not been initialized yet, or DAC synchronous data acquisition process has been aborted. *DAC_SYNC_DAQ_END* bit will stay in the '0' state during *DAC-ASYNC-DAQ*, *DAC-SYNC-PX-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes. Reading *DAC_SYC_DAQ_END* =1 corresponds to normally terminated DAC synchronous data acquisition process on DAC synchronous data acquisition termination event for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode in accordance with bit *DAC_SYNC_DAQ_TF_SEL* of *DAQ_CNTR3_RG* register. This condition can be used to generate interrupt request to host PIOX-16 interface. *DAC_SYNC_DAQ_END* bit will reset to the '0' state on start of DAC synchronous data acquisition process (by setting bit *DAC_SYNC_DAQ_RUN* of *DAQ_CNTR2_RG* register to the '1' state), when '1' is written to the *DAC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register, when DAC data acquisition controller is reset (by write to either *DAC_DAQ_RESET_RG* or to *ADDA_RESET_RG* registers), or when DAC asynchronous data acquisition mode is selected.

Note:
1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

DAQ_CNTR3_RG control register

DAQ_CNTR3_RG register must be used to select ADC/DAC data acquisition termination flags for ADC/DAC synchronous FIFO one-pass data acquisition mode (*ADC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-OPM-DAQ* correspondingly), and to select start/termination flags for ADC/DAC IRT controllers for all ADC synchronous FIFO data acquisition modes (*ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ*) and all DAC synchronous FIFO data acquisition modes (*DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ*).

DAQ_CNTR3_RG register is available for read/write and is allocated at the least significant byte of 16-bit data word of host PIOX-16 interface.

DAQ_CNTR3_RG register (r/w)

X	*0*	*0*	*0*	*0*	*DAC_SYNC_FIFO_ IRT_TF_SEL (r/w, 0+)*	*ADC_SYNC_FIFO_ IRT_TF_SEL (r/w, 0+)*	*DAC_SYNC_FIFO_ DAQ_TF_SEL (r/w, 0+)*	*ADC_SYNC_FIFO_ DAQ_TF_SEL (r/w, 0+)*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1	bit-0

Table 2-4 provides details about *DAQ_CNTR3_RG* register bits.

Table 2-4. DAQ_CNTR3_RG register bits.

register bits	access mode	default value on PIOX-16 reset	Description
ADC_SYNC_FIFO_ DAQ_TF_SEL	r/w	0	Selects ADC data acquisition termination flag for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode and selects ADC IRT transmission start flag for ADC IRT controller for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-OPM-DAQ* data acquisition modes. Refer to sections “Data Acquisition Control” and “ADC IRT and DAC IRT Controllers” later in this chapter for more details about ADC data acquisition controller and ADC IRT controller. *ADC_SYNC_FIFO_DAQ_TF_SEL*=0 denotes that ADC FIFO FF (FIFO full) flag is selected to terminate ADC synchronous data acquisition process in *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode and to start IRT transmission cycle for ADC IRT controller for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-OPM-DAQ* data acquisition modes. *ADC_SYNC_FIFO_DAQ_TF_SEL*=1 denotes that ADC FIFO PAF (FIFO partially-full) flag is selected to terminate ADC synchronous data acquisition process in *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode and to start IRT transmission cycle for ADC IRT controller for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-OPM-DAQ* data acquisition modes.
DAC_SYNC_FIFO_ DAQ_TF_SEL	r/w	0	Selects DAC data acquisition termination flag for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode and selects DAC IRT transmission start flag for DAC IRT controller for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-OPM-DAQ* data acquisition modes. Refer to sections “Data Acquisition Control” and “ADC IRT and DAC IRT Controllers” later in this chapter for more details about DAC data acquisition controller and DAC IRT controller. *DAC_SYNC_FIFO_DAQ_TF_SEL*=0 denotes that DAC FIFO EF (FIFO empty) flag is selected to terminate DAC synchronous data acquisition process in *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode and to start IRT transmission cycle for DAC IRT controller for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-OPM-DAQ* data acquisition modes. *DAC_SYNC_FIFO_DAQ_TF_SEL*=1 denotes that DAC FIFO PAE (FIFO partially-empty) flag is selected to terminate DAC synchronous data acquisition process in *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode and to start IRT transmission cycle for DAC IRT controller for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-OPM-DAQ* data acquisition modes.

ADC_SYNC_FIFO_IRT_TF_SEL	r/w	0	Selects ADC IRT transmission termination flag for ADC IRT controller for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-OPM-DAQ* data acquisition modes. Refer to sections "Data Acquisition Control" and "ADC IRT and DAC IRT Controllers" later in this chapter for more details about ADC data acquisition controller and ADC IRT controller. *ADC_SYNC_FIFO_DAQ_IRT_TF_SEL*=0 denotes that ADC FIFO EF (FIFO empty) flag is selected to terminate IRT transmission cycle for ADC IRT controller for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-OPM-DAQ* data acquisition modes. *ADC_SYNC_FIFO_DAQ_IRT_TF_SEL*=1 denotes that ADC FIFO PAE (FIFO partially-empty) flag is selected to IRT transmission cycle for ADC IRT controller for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-OPM-DAQ* data acquisition modes.
DAC_SYNC_FIFO_IRT_TF_SEL	r/w	0	Selects DAC IRT transmission termination flag for DAC IRT controller for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-OPM-DAQ* data acquisition modes. Refer to sections "Data Acquisition Control" and "ADC IRT and DAC IRT Controllers" later in this chapter for more details about DAC data acquisition controller and DAC IRT controller. *DAC_SYNC_FIFO_DAQ_IRT_TF_SEL*=0 denotes that DAC FIFO FF (FIFO full) flag is selected to terminate IRT transmission cycle for DAC IRT controller for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-OPM-DAQ* data acquisition modes. *DAC_SYNC_FIFO_DAQ_IRT_TF_SEL*=1 denotes that DAC FIFO PAF (FIFO partially-full) flag is selected to IRT transmission cycle for DAC IRT controller for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-OPM-DAQ* data acquisition modes.

Note:
1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

DAQ_SYNC_RG control register

DAQ_SYNC_RG register must be used to select sampling frequency source, start synchronization sources and mode for ADC/DAC synchronous data acquisition processes, and ADC timing options for different ADC data acquisition modes. *DAQ_SYNC_RG* register is available for read/write and is allocated at the least significant byte of 16-bit data word of host PIOX-16 interface.

***DAQ_SYNC_RG* register (r/w)**

X	*ADC_HIGH_SPEED_EN (r/w, 0+)*	*XSSYNC_MODE (r/w, 0+)*	*DAC_SYNC_DAQ_SSYNC_SEL (r/w, 0+)*	*ADC_SYNC_DAQ_SSYNC_SEL (r/w, 0+)*	*ADC_SYNC_PX_DAQ_SCCL_DLY_EN (r/w, 0+)*	*ADC_ASYNC_DAQ_ACQ_EN (r/w, 1+)*	*FS_SEL-1 (r/w, 0+)*	*FS_SEL-0 (r/w, 0+)*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1	bit-0

Table 2-5 provides details about *DAQ_SYNC_RG* register bits.

Table 2-5. Register bits of *DAQ_SYNC_RG* register.

register bits	**access mode**	**default value on PIOX-16 reset**	**Description**
{*FS_SEL-1, FS_SEL-0*}	r/w	{0,0}	Selects sampling frequency source for ADC and DAC data acquisition controllers. Refer to section “Data Acquisition Control” later in this chapter for more details about ADC data acquisition controller. {*FS_SEL*-[1:0]}={0,0} denotes that on-board programmable high-resolution sampling frequency generator (PFG) is used as the sampling frequency source. {*FS_SEL*-[1:0]}={0,1} denotes that external sampling frequency (XFs) input from on-board JP2 connector used as the sampling frequency source (refer to Appendix A). {*FS_SEL*-[1:0]}={1,0} denotes that PIOX-16 interface timer #0 pin (TM/XIO-0) used as the sampling frequency source. {*FS_SEL*-[1:0]}={1,1} denotes that PIOX-16 interface timer #1 pin (TM/XIO-1) used as the sampling frequency source.
ADC_ASYNC_DAQ_ ACQ_EN	r/w	1	Defines whether ADC data acquisition controller will generate either *standard ADC sampling cycles* or *extended ADC sampling cycles* in ADC asynchronous data acquisition mode. Refer to section “Data Acquisition Control” later in this chapter for more details about ADC data acquisition controller. *ADC_ASYNC_DAQ_ACQ_EN*=0 denotes that ADC data acquisition controller will generate standard *ADC sampling cycles*, i.e. ADC asynchronous data acquisition cycle begins with immediate A/D conversion start. However, in this case special care must be take in order to ensure that either at least 0.8 μS time has passed after analog input multiplexer has been updated, or at least 0.25 μS time has passed since last ADC data ready condition, whichever comes last. This is required in order to guarantee that analog input signal at the ADC input is acquired properly within sufficient accuracy. Generally, if analog input multiplexer is not switched to different analog inputs during ADC asynchronous data acquisition mode, this setting allows to perform A/D conversion at maximum 1 Msps performance. *ADC_ASYNC_DAQ_ACQ_EN*=1 denotes that ADC data acquisition controller will generate *extended ADC sampling cycles*, i.e. A/D conversion start is delayed after beginning of ADC asynchronous data acquisition cycle for 0.8μS in order to guarantee that analog input signal at the ADC input has been acquired properly within sufficient accuracy. This setting is recommended in case analog input multiplexer is assumed to switch to different analog inputs. This mode reduces maximum A/D conversion performance to approximately 645 ksps (1.55μS A/D sampling/conversion time). This is default setting on host PIOX-16 interface reset condition.

ADC_SYNC_PX_ DAQ_SCCL_DLY_EN	r/w	0	Defines whether ADC scan cycle length is extended by 1μS in order to provide extra time for host DSP to read ADC data via host PIOX-16 interface. This option is valid for *ADC-SYNC-PX-DAQ* data acquisition mode only. Refer to section "Data Acquisition Control" later in this chapter for more details about ADC data acquisition controller. *ADC_SYNC_PX_DAQ_SCCL_DLY_EN*=0 denotes that duration of ADC scan cycle for *ADC-SYNC-PX-DAQ* data acquisition mode is normal and corresponds to maximum ADC performance. However, care must be taken in order to ensure that ADC data will be downloaded to host DSP environment via host PIOX-16 interface prior next ADC data will come valid in next ADC scan cycle. With this setting, duration of ADC scan cycle corresponds to maximum ADC performance and is 1μS for ADC high-speed mode and 1.25 μS for ADC normal speed mode. *ADC_SYNC_PX_DAQ_SCCL_DLY_EN*=1 denotes that duration of ADC scan cycle for *ADC-SYNC-PX-DAQ* data acquisition mode is extended by extra 1μS in order to provide extra time to host DSP to download ADC data to host DSP environment via host PIOX-16 interface prior next ADC data will come valid in next ADC scan cycle. With this setting, duration of ADC scan cycle is 2μS for ADC high-speed mode and 2.25 μS for ADC normal speed mode.
ADC_SYNC_DAQ_ SSYNC_SEL	r/w	0	Selects start synchronization source for ADC synchronous data acquisition process (for all ADC synchronous data acquisition modes). Refer to section "Data Acquisition Control" later in this chapter for more details about ADC data acquisition controller. *ADC_SYNC_DAQ_SSYNC_SEL*=0 corresponds to software start synchronization for ADC synchronous data acquisition process, i.e. ADC synchronous data acquisition process will start immediately after host DSP software will set bit *ADC_SYNC_DAQ_RUN* of *DAQ_CNTR2_RG* register to the '1' state. *ADC_SYNC_DAQ_SSYNC_SEL*=1 corresponds to external hardware defined start synchronization for ADC synchronous data acquisition process at *ADC_XSSYNC_IN* input pin of on-board JP3 connector (refer to Appendix A). ADC synchronous data acquisition process will start on external start synchronization event at *ADC_XSSYNC_IN* input pin after host DSP software will set bit *ADC_SYNC_DAQ_RUN* of *DAQ_CNTR2_RG* register to the '1' state. External start synchronization event at *ADC_XSSYNC_IN* input pin can be selected either as active low level or falling edge condition depending upon the state of *XSSYNC_MODE* bit of *DAQ_SYNC_RG* register, which is common for both ADC and DAC data acquisition controllers.

DAC_SYNC_DAQ_ SSYNC_SEL	r/w	0	Selects start synchronization source for DAC synchronous data acquisition process (for all DAC synchronous data acquisition modes). Refer to section “Data Acquisition Control” later in this chapter for more details about DAC data acquisition controller. *DAC_SYNC_DAQ_SSYNC_SEL*=0 corresponds to software start synchronization for DAC synchronous data acquisition process, i.e. DAC synchronous data acquisition process will start immediately after host DSP software will set bit *DAC_SYNC_DAQ_RUN* of *DAQ_CNTR2_RG* register to the ‘1’ state. *DAC_SYNC_DAQ_SSYNC_SEL*=1 corresponds to external hardware defined start synchronization for DAC synchronous data acquisition process at *DAC_XSSYNC_IN* input pin of on-board JP3 connector (refer to Appendix A). DAC synchronous data acquisition process will start on external start synchronization event at *DAC_XSSYNC_IN* input pin after host DSP software will set bit *DAC_SYNC_DAQ_RUN* of *DAQ_CNTR2_RG* register to the ‘1’ state. External start synchronization event at *DAC_XSSYNC_IN* input pin can be selected either as active low level or falling edge condition depending upon the state of *XSSYNC_MODE* bit of *DAQ_SYNC_RG* register, which is common for both ADC and DAC data acquisition controllers..
XSSYNC_MODE	r/w	0	Selects external start synchronization condition for both ADC and DAC data acquisition controllers at *ADC_XSSYNC_IN* and *DAC_XSSYNC_IN* input pins correspondingly of on-board JP3 connector (refer to Appendix A) in case external hardware defined start synchronization is selected as start synchronization source for ADC and/or DAC synchronous data acquisition processes. Refer to section “Data Acquisition Control” later in this chapter for more details about ADC and DAC data acquisition controllers. *XSSYNC_MODE*=0 corresponds to active low level condition as external start synchronization event for both ADC and DAC data acquisition controllers at *ADC_XSSYNC_IN* and *DAC_XSSYNC_IN* input pins correspondingly of on-board JP3 connector for ADC and DAC data acquisition controllers correspondingly in case external hardware defined start synchronization is selected as start synchronization source for ADC and/or DAC synchronous data acquisition processes. *XSSYNC_MODE*=1 corresponds to the falling edge condition as external start synchronization event for both ADC and DAC data acquisition controllers at *ADC_XSSYNC_IN* and *DAC_XSSYNC_IN* input pins correspondingly of on-board JP3 connector for ADC and DAC data acquisition controllers correspondingly in case external hardware defined start synchronization is selected as start synchronization source for ADC and/or DAC synchronous data acquisition processes.

ADC_HIGH_ SPEED_EN	r/w	0	Defines whether ADC high-speed mode is enabled. This option define A/D conversion time only for all ADC channels and does not effect ADC sampling time. This option is valid for all ADC data acquisition modes and directly defines ADC performance. Refer to section "Data Acquisition Control" later in this chapter for more details about ADC data acquisition controller. *ADC_HIGH_SPEED_EN*=0 denotes that all ADC channels operate in normal speed mode, which corresponds to maximum ADC sampling frequency 800 kHz. Although this setting delivers –20% lower ADC performance than the ADC high-speed mode does, it provides valid ADC output data starting from the first ADC sampling cycle, and there is no minimum restrictions for ADC sampling frequency value. This setting is recommended for sampling frequencies below 10 kHz. *ADC_HIGH_SPEED_EN*=1 denotes that all ADC channels operate in high-speed mode with maximum ADC sampling frequency as high as 1 MHz. However, in this ADC mode, ADC sampling frequency value must not go lower than 1 kHz. Also, with this setting, ADC output data for the first A/D conversion cycle for all ADC channels shall be discarded in case the time elapsed since previous A/D conversion cycle exceeds 1 mS. This setting is recommended for sampling frequencies above 10 kHz.

Note:
1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

ADC_CNF_RG and DAC_CNF_RG control registers

ADC_CNF_RG register must be used to select particular ADC channels, which will be involved into ADC data acquisition process for all ADC data acquisition modes. Correspondingly, *DAC_CNF_RG* register must be used to select particular DAC channels, which will be involved into DAC data acquisition process for all DAC data acquisition modes.

ADC_CNF_RG and *DAC_CNF_RG* registers are available for read/write and are allocated at the least significant byte of 16-bit data word of host PIOX-16 interface.

***ADC_CNF_RG* register (r/w)**

X	*1 (r)*	*0 (r)*	*0 (r)*	*0 (r)*	*ADC3_EN0 (r/w, 1+)*	*ADC2_EN0 (r/w, 1+)*	*ADC1_EN0 (r/w, 1+)*	*ADC0_EN0 (r/w, 1+)*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1	Bit-0

***DAC_CNF_RG* register (r/w)**

X	*1 (r)*	*0 (r)*	*0 (r)*	*0 (r)*	*DAC3_EN0 (r/w, 1+)*	*DAC2_EN0 (r/w, 1+)*	*DAC1_EN0 (r/w, 1+)*	*DAC0_EN0 (r/w, 1+)*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1	Bit-0

Tables 2-6 and 2-7 provide details about *ADC_CNF_RG* and *DAC_CNF_RG* registers bits.

Table 2-6. Register bits of *ADC_CNF_RG* register.

register bits	**access mode**	**default value on PIOX-16 reset**	**Description**
ADC0_EN *ADC1_EN* *ADC2_EN* *ADC3_EN*	r/w	1 1 1 1	Define what particular ADC channels (ADC-0..3) are involved into ADC data acquisition process for all ADC data acquisition modes. Refer to section “Data Acquisition Control” later in this chapter for more details about ADC data acquisition controller. *ADCx_EN*=0 (*x*=0..3) denotes that the ADC-x channel is not involved into ADC data acquisition process. *ADCx_EN*=1 (*x*=0..3) denotes that the ADC-x channel is involved into ADC data acquisition process.

Note:
1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

Table 2-7. Register bits of *DAC_CNF_RG* register.

register bits	**access mode**	**default value on PIOX-16 reset**	**Description**
DAC0_EN *DAC1_EN* *DAC2_EN* *DAC3_EN*	r/w	1 1 1 1	Define what particular DAC channels (ADC-0..3) are involved into DAC data acquisition process for all ADC data acquisition modes. Refer to section “Data Acquisition Control” later in this chapter for more details about DAC data acquisition controller. *DACx_EN*=0 (*x*=0..3) denotes that the DAC-x channel is not involved into DAC data acquisition process. *DACx_EN*=1 (*x*=0..3) denotes that the DAC-x channel is involved into DAC data acquisition process.

Note:
1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

It is important to note, that the contents of *ADC_CNF_RG* register actually does not define whether A/D conversion for particular ADC channel will be started or not started during ADC data acquisition process, except for ADC synchronous data acquisition process, which will be not initialized in case all ADC channels are disabled. Instead, the contents of *ADC_CNF_RG* register actually defines which particular and how many ADC channels will be automatically downloaded into host DSP environment in case ADC IRT controller in conjunction with host DSP on-chip DMA controller are being used for ADC data download to host DSP environment via host PIOX-16 interface. Also, for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes, *ADC_CNF_RG* register defines which particular ADC channels will be automatically read and pushed into ADC FIFO in each ADC scan cycle of ADC sampling packet.

Correspondingly, the contents of *DAC_CNF_RG* register defines which particular and how many DAC channels will be automatically uploaded in case DAC IRT controller in conjunction with host DSP on-chip DMA controller are being used for DAC data upload from host DSP environment via host PIOX-16 interface. Also, for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes, the contents of *DAC_CNF_RG* register defines which particular DAC channels will be automatically written from DAC FIFO in each DAC sampling cycle.

CAUTION

ADC_CNF_RG and *DAC_CNF_RG* registers are used by ADC/DAC data acquisition controllers in ADC/DAC synchronous FIFO data acquisition modes only in order to define what particular ADC/DAC channels will be pushed/pulled to/from ADC/DAC FIFO during ADC scan cycles and DAC sampling cycles correspondingly.

All-zero contents of *ADC_CNF_RG* and/or *DAC_CNF_RG* registers will stop corresponding ADC/DAC synchronous data acquisition process for all ADC/DAC synchronous data acquisition modes.

ADC_CNF_RG and *DAC_CNF_RG* registers are always used by ADC/DAC IRT controllers in order to define what particular ADC/DAC channels, which data will be transferred to/from host DSP environment (refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details).

For more details about ADC and DAC data acquisition controllers, refer to section "Data Acquisition Control" later in this chapter.

ADC_IMUX_CNF1_RG and ADC_IMUX_CNF2_RG control registers

ADC_IMUX_CNF1_RG and *ADC_IMUX_CNF2_RG* registers shall be used to configure analog input multiplexers for ADC asynchronous data acquisition mode (*ADC_IMUX_CNF1_RG* register only) and for each scan cycle of ADC sampling packet for all ADC synchronous data acquisition modes.

ADC_IMUX_CNF1_RG and *ADC_IMUX_CNF2_RG* registers are available for read/write and are allocated at the least significant byte of 16-bit data word of host PIOX-16 interface.

ADC_IMUX_CNF1_RG register (r/w)
(ADC asynchronous data acquisition mode only)

x	*x* *(r/w, 0+)*	*x* *(r/w, 0+)*	*x* *(r/w, 0+)*	*x* *(r/w, 0+)*	*x* *(r/w, 0+)*	*ASYNC_IMUX_DIFF* *(r/w, 0+)*	*ASYNC_IMUX_ CH_SEL-0* *(r/w, 0+)*	*ASYNC_IMUX_ CH_SEL-0* *(r/w, 0+)*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1	bit-0

ADC_IMUX_CNF1_RG register (r/w)
(all ADC synchronous data acquisition modes only)

x	*SCCL1_EOS* *(r/w, 0+)*	*SCCL1_IMUX_DIFF* *(r/w, 0+)*	*SCCL1_IMUX_CH_SEL-0* *(r/w, 0+)*	*SCCL1_IMUX_CH_SEL-0* *(r/w, 1+)*	*SCCL0_EOS* *(r/w, 0+)*	*SCCL0_IMUX_DIFF* *(r/w, 0+)*	*SCCL0_IMUX_CH_SEL-0* *(r/w, 0+)*	*SCCL0_IMUX_CH_SEL-0* *(r/w, 0+)*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1	bit-0

ADC_IMUX_CNF2_RG register (r/w)
(all ADC synchronous data acquisition modes only)

x	*1* *(r)*	*SCCL3 IMUX_DIFF* *(r/w, 0+)*	*SCCL3_IMUX_CH_SEL-0* *(r/w, 1+)*	*SCCL3_IMUX_CH_SEL-0* *(r/w, 1+)*	*SCCL2_EOS* *(r/w, 0+)*	*SCCL2_IMUX_DIFF* *(r/w, 0+)*	*SCCL2_IMUX_CH_SEL-0* *(r/w, 1+)*	*SCCL2_IMUX_CH_SEL-0* *(r/w, 0+)*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1	bit-0

Tables 2-8a and 2-8b provides details about *ADC_IMUX_CNF1_RG* and *ADC_IMUX_CNF2_RG* registers bits for different ADC data acquisition modes.

Table 2-8a. Register bits of *ADC_IMUX_CNF1_RG* register for ADC asynchronous data acquisition mode.

register bits	access mode	default value on PIOX-16 reset	Description
{*ASYNC_IMUX_CH_SEL-1*, *ASYNC_IMUX_CH_SEL-0*}	r/w	{0,0}	Configure input channel selector for analog input multiplexers (AIN-MUX0..3) of analog input section for ADC asynchronous data acquisition mode. Analog input multiplexers for each ADC channel (AIN-MUX0..3) are all configured identically. For single-ended analog input configuration of analog input multiplexers of ADC channels (bit *ASYNC_IMUX_DIFF* of *ADC_IMUX_CNF1_RG* register is set to the '0' state), these bits select particular analog input from four available for each ADC channel, which will be connected to the input of ADC chip of the corresponding ADC channel for further conversion to digital code. In case analog input multiplexer is configured in differential mode (bit *ASYNC_IMUX_DIFF* bit of *ADC_IMUX_CNF1_RG* register is set to the '1' state), then bit *ASYNC_IMUX_CH_SEL-1* is ignored, and bit *ASYNC_IMUX_CH_SEL-0* defines which of two available differential analog inputs for each ADC channel will be connected to the input of ADC chip of the corresponding ADC channel for further conversion to digital code. Refer to section "Data Acquisition Control" later in this chapter for more details about ADC data acquisition controller. {*ASYNC_IMUX_CH_SEL*-[1:0]}={0,0} denotes that AIN-0/AIN-4/AIN-8/AIN-12 analog inputs for single-ended configuration of analog input multiplexers, and (AIN-0 – AIN-2)/(AIN-4 – AIN-6)/(AIN-8 – AIN-10)/(AIN-12 – AIN-14) differential analog inputs for differential configuration of analog input multiplexers are connected to the inputs of ADC chips of the corresponding ADC channels. {*ASYNC_IMUX_CH_SEL*-[1:0]}={0,1} denotes that AIN-1/AIN-5/AIN-9/AIN-13 analog inputs for single-ended configuration of analog input multiplexers, and (AIN-1 – AIN-2)/(AIN-5 – AIN-7)/(AIN-9 – AIN-11)/(AIN-13 – AIN-15) differential analog inputs for differential configuration of analog input multiplexers are connected to the inputs of ADC chips of the corresponding ADC channels. {*ASYNC_IMUX_CH_SEL*-[1:0]}={1,0} denotes that AIN-2/AIN-6/AIN-10/AIN-14 analog inputs for single-ended configuration of analog input multiplexers, and (AIN-0 – AIN-2)/(AIN-4 – AIN-6)/(AIN-8 – AIN-10)/(AIN-12 – AIN-14) differential analog inputs for differential configuration of analog input multiplexers are connected to the inputs of ADC chips of the corresponding ADC channels. {*ASYNC_IMUX_CH_SEL*-[1:0]}={1,1} denotes that AIN-3/AIN-7/AIN-11/AIN-15 analog inputs for single-ended configuration of analog input multiplexers, and (AIN-1 – AIN-2)/(AIN-5 – AIN-7)/(AIN-9 – AIN-11)/(AIN-13 – AIN-15) differential analog inputs for differential configuration of analog input multiplexers are connected to the inputs of ADC chips of the corresponding ADC channels.

ASYNC_IMUX_DIFF	r/w	0	Configures analog input multiplexers of analog input section for ADC asynchronous data acquisition mode to operate in either single-ended or differential analog input configuration. Analog input multiplexers for each ADC channel (AIN-MUX0..3) are all configured identically. When configured in single-ended analog input configuration, analog input multiplexers of each ADC channel can select one of four available analog inputs (defined via bits *ASYNC_IMUX_CH_SEL*-[1:0]} of *ADC_IMUX_CNF1_RG* register), which will be connected to the input of ADC chip of the corresponding ADC channel for further conversion to digital code. Instead, in case analog input multiplexer is configured in differential mode, then bit *ASYNC_IMUX_CH_SEL-1* of *ADC_IMUX_CNF1_RG* register is ignored, and bit *ASYNC_IMUX_CH_SEL-0* defines which of two available differential analog inputs for each ADC channel will be connected to the input of ADC chip of the corresponding ADC channel for further conversion to digital code. Refer to section “Data Acquisition Control” later in this chapter for more details about ADC data acquisition controller. *ASYNC_IMUX_DIFF*=0 denotes that analog input multiplexers for each ADC channel are configured in single-ended analog input configuration. *ASYNC_IMUX_DIFF*=1 denotes that analog input multiplexers for each ADC channel are configured in differential analog input configuration.

Note:
1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

Table 2-8b. Register bits of *ADC_IMUX_CNF1_RG* and *ADC_IMUX_CNF2_RG* registers for all ADC synchronous data acquisition modes.

register bits	**access mode**	**default value on PIOX-16 reset**	**Description**
{*SCCL0_IMUX_CH_SEL-1, SCCL0_IMUX_CH_SEL-0*}	r/w	{0,0}	Configure input channel selector for analog input multiplexers (AIN-MUX0..3) of analog input section for scan cycles #0, #1, #2 and #3 correspondingly of ADC sampling packet for all ADC synchronous data acquisition modes. Analog input multiplexers for each ADC channel (AIN-MUX0..3) in each scan cycle are all configured identically. For single-ended analog input configuration of analog input multiplexers of ADC channels for particular scan cycle (bit *SCCLx_IMUX_DIFF* is set to the '0' state, *x*=0..3), these bits select particular analog input from four available for each ADC channel, which will be connected to the input of ADC chip of the corresponding ADC channel for further conversion to digital code. In case analog input multiplexer is configured in differential mode for particular scan cycle (bit *SCCLx_IMUX_DIFF* is set to the '1' state), then bit *SCCLx_IMUX_CH_SEL-1* is ignored, and bit *SCCLx_IMUX_CH_SEL-0* defines which of two available differential analog inputs for each ADC channel will be connected to the input of ADC chip of the corresponding ADC channel for further conversion to digital code. Refer to section "Data Acquisition Control" later in this chapter for more details about ADC data acquisition controller. {*SCCLx_IMUX_CH_SEL*-[1:0]}={0,0} denotes that AIN-0/AIN-4/AIN-8/AIN-12 analog inputs for single-ended configuration of analog input multiplexers, and (AIN-0 – AIN-2)/(AIN-4 – AIN-6)/(AIN-8 – AIN-10)/(AIN-12 – AIN-14) differential analog inputs for differential configuration of analog input multiplexers are connected to the inputs of ADC chips of the corresponding ADC channels in scan cycle #x (*x*=0..3) of ADC sampling packet. This is default setting for scan cycle #0 at host PIOX-16 interface reset condition. {*SCCLx_IMUX_CH_SEL*-[1:0]}={0,1} denotes that AIN-1/AIN-5/AIN-9/AIN-13 analog inputs for single-ended configuration of analog input multiplexers, and (AIN-1 – AIN-2)/(AIN-5 – AIN-7)/(AIN-9 – AIN-11)/(AIN-13 – AIN-15) differential analog inputs for differential configuration of analog input multiplexers are connected to the inputs of ADC chips of the corresponding ADC channels in scan cycle #x (*x*=0..3) of ADC sampling packet. This is default setting for scan cycle #1 at host PIOX-16 interface reset condition. {*SCCLx_IMUX_CH_SEL*-[1:0]}={1,0} denotes that AIN-2/AIN-6/AIN-10/AIN-14 analog inputs for single-ended configuration of analog input multiplexers, and (AIN-0 – AIN-2)/(AIN-4 – AIN-6)/(AIN-8 – AIN-10)/(AIN-12 – AIN-14) differential analog inputs for differential configuration of analog input multiplexers are connected to the inputs of ADC chips of the corresponding ADC channels in scan cycle #x (*x*=0..3) of ADC sampling packet. This is default setting for scan cycle #2 at host PIOX-16 interface reset condition. {*SCCLx_IMUX_CH_SEL*-[1:0]}={1,1} denotes that AIN-3/AIN-7/AIN-11/AIN-15 analog inputs for single-ended configuration of analog input multiplexers, and (AIN-1 – AIN-2)/(AIN-5 – AIN-7)/(AIN-9 – AIN-11)/(AIN-13 – AIN-15) differential analog inputs for differential configuration of analog input multiplexers are connected to the inputs of ADC chips of the corresponding ADC channels in scan cycle #x (*x*=0..3) of ADC sampling packet. This is default setting for scan cycle #3 at host PIOX-16 interface reset condition.
{*SCCL1_IMUX_CH_SEL-1, SCCL1_IMUX_CH_SEL-0*}		{0,1}	
{*SCCL2_IMUX_CH_SEL-1, SCCL2_IMUX_CH_SE -0*}		{1,0}	
{*SCCL3_IMUX_CH_SEL-1, SCCL3_IMUX_CH_SEL-0*}		{1,1}	

SCCL0_IMUX_DIFF *SCCL1_IMUX_DIFF* *SCCL2_IMUX_DIFF* *SCCL3_IMUX_DIFF*	r/w	0 0 0 0	Configure analog input multiplexers of analog input section for scan cycles #0, #1, #2 and #3 correspondingly of ADC sampling packet for all ADC synchronous data acquisition modes to operate in either single-ended or differential analog input configuration. Analog input multiplexers for each ADC channel (AIN-MUX0..3) in each scan cycle are all configured identically. When configured in single-ended analog input configuration for particular scan cycle, analog input multiplexers of each ADC channel can select one of four available analog inputs (defined via bits *SCCLx_IMUX_CH_SEL*-[1:0], *x*=0..3), which will be connected to the input of ADC chip of the corresponding ADC channel for further conversion to digital code. Instead, in case analog input multiplexer is configured in differential mode for particular scan cycle, then bit *SCCLx_IMUX_CH_SEL-1* is ignored, and bit *SCCLx_IMUX_CH_SEL-0* defines which of two available differential analog inputs for each ADC channel will be connected to the input of ADC chip of the corresponding ADC channel for further conversion to digital code. Refer to section “Data Acquisition Control” later in this chapter for more details about ADC data acquisition controller. *SCCLx_IMUX_DIFF*=0 denotes that analog input multiplexers for each ADC channel are configured in single-ended analog input configuration for scan cycle #x (*x*=0..3) of ADC sampling packet. *SCCLx_IMUX_DIFF*=1 denotes that analog input multiplexers for each ADC channel are configured in differential analog input configuration for scan cycle #x (*x*=0..3) of ADC sampling packet.
SCCL0_EOS *SCCL1_EOS* *SCCL2_EOS*	r/w	0 0 0	Define state of the end-of scan (EOS) flag for scan cycles #0, #1 and #2 correspondingly of ADC sampling packet for all ADC synchronous data acquisition modes. Note, that *SCCL3_EOS* flag for scan cycle #3, which is defined by bit #7 of *ADC_IMUX_CNF2_RG* register, always reads as ‘1’ and cannot be modified by host DSP software in order to ensure up to four scan cycles per each ADC sampling packet. ADC data acquisition controller, when configured in any of ADC synchronous data acquisition modes, generates scan cycles and analyzes *SCCLx_EOS* flags (*x*=0..2) starting from the scan cycle #0 within ADC sampling packet, until it detects first non-zero *SCCLx_EOS* flag. In case all *SCCLx_EOS* flags (*x*=0..2) are set to the ‘0’ state, then ADC data acquisition controller generates fourth scan cycle and terminates. Refer to section “Data Acquisition Control” later in this chapter for more details about ADC data acquisition controller. *SCCLx_EOS*=0 denotes that scan cycle #x (*x*=0..2) of ADC sampling packet is not the last one (in case *SCCLx_EOS* flags for previous scan cycles are all set to the ‘0’ state), and ADC data acquisition controller must proceed with next scan cycle within ADC sampling packet. In case any of EOS flags for previous scan cycles is set to the ‘1’ state, then *SCCLx_EOS* flag for scan cycle #x and all next scan cycles are ignored. *SCCLx_EOS*=1 denotes that scan cycle #x (*x*=0..2) of ADC sampling packet is the last one (in case *SCCLx_EOS* flags for previous scan cycles are all set to the ‘0’ state), and ADC data acquisition controller must terminate ADC sampling packet. In case any of EOS flags for previous scan cycles is set to the ‘1’ state, then *SCCLx_EOS* flag for scan cycle #x and all next scan cycles are ignored.

Note:
1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

Generation of reset signal for ADC/DAC data acquisition controllers and ADC/DAC IRT

ADC/DAC data acquisition controllers can be reset by host DSP software in order to terminate current data acquisition process, clear ADC and DAC FIFO read and write pointers, and to reset the corresponding ADC/DAC data acquisition controllers hardware.

ADC FIFO, ADC data acquisition controller and ADC IRT controller can be reset by writing to either *ADC_DAQ_RESET_RG* or *ADDA_DAQ_RESET_RG* write-only registers, whereas DAC FIFO, DAC data acquisition controller and DAC IRT controller can be reset by writing to either *DAC_DAQ_RESET_RG* or *ADDA_DAQ_RESET_RG* write-only registers. Note, that *ADDA_DAQ_RESET_RG* write-only register can be used to simultaneously reset ADC/DAC FIFO, ADC/DAC data acquisition controllers and ADC/DAC IRT controllers.

CAUTION

Data written to *ADC_DAQ_RESET_RG*, *DAC_DAQ_RESET_RG* and *ADDA_DAQ_RESET_RG* write-only registers is ignored, and only write action as it is has effect.

***ADC_DAQ_RESET_RG* register (w)**

bits 15..8	bit-7	Bit-6	Bit-5	Bit-4	bit-3	bit-2	bit-1	bit-0
x	x (w)	X (w)	X (w)	x (w)	x (w)	x (w)	x (w)	x (w)

***DAC_DAQ_RESET_RG* register (w)**

bits 15..8	bit-7	Bit-6	Bit-5	Bit-4	bit-3	bit-2	bit-1	bit-0
x	x (w)	x (w)	X (w)	x (w)	x (w)	x (w)	x (w)	x (w)

***ADDA_DAQ_RESET_RG* register (w)**

bits 15..8	bit-7	Bit-6	Bit-5	Bit-4	bit-3	bit-2	bit-1	bit-0
x	x (w)	x (w)	X (w)	x (w)	x (w)	x (w)	x (w)	x (w)

Refer to section "Data Acquisition Control" later in this chapter for more details about ADC and DAC data acquisition controllers. Resetting ADC/DAC data acquisition controller(s) will also reset the corresponding ADC/DAC IRT controller(s).

ADC_START_RG, DAC_OUTPUT_LD_RG and ADC_START_DAC_OUTPUT_LD_RG registers

ADC_START_RG and *DAC_OUTPUT_LD_RG* write-only registers are used in ADC and DAC asynchronous data acquisition modes only in order to start ADC sampling cycle and to perform DAC output data load. Note, that write to *DAC_OUTPUT_LD_RG* register has effect only in DAC asynchronous data acquisition mode and does not depend upon the state of bits {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE-*[1:0]} of *DAQ_CNTR1_RG* register.

Combined *ADC_START_DAC_OUTPUT_LD_RG* write-only register can be used to simultaneously start ADC sampling cycle and to perform DAC output data load.

CAUTION

Data written to *ADC_START_RG*, *DAC_OUTPUT_LD_RG* and *ADC_START_DAC_OUTPUT_LD_RG* write-only registers is ignored, and only write action as it is has effect.

ADC_START_RG register (w)

x	*x* *(w)*	*X* *(w)*	*X* *(w)*	*x* *(w)*	*x* *(w)*	*x* *(w)*	*x* *(w)*	*x* *(w)*
bits 15..8	bit-7	Bit-6	Bit-5	Bit-4	bit-3	bit-2	bit-1	bit-0

DAC_OUTPUT_LD_RG register (w)

x	*x* *(w)*	*X* *(w)*	*X* *(w)*	*x* *(w)*	*x* *(w)*	*x* *(w)*	*x* *(w)*	*x* *(w)*
bits 15..8	bit-7	Bit-6	Bit-5	Bit-4	bit-3	bit-2	bit-1	bit-0

ADC_START_DAC_OUTPUT_LD_RG register (w)

x	*x* *(w)*	*X* *(w)*	*X* *(w)*	*x* *(w)*	*x* *(w)*	*x* *(w)*	*x* *(w)*	*x* *(w)*
bits 15..8	bit-7	Bit-6	Bit-5	Bit-4	bit-3	bit-2	bit-1	bit-0

Refer to section “Data Acquisition Control” later in this chapter for more details about ADC and DAC data acquisition controllers. Resetting ADC/DAC data acquisition controller(s) will also reset the corresponding ADC/DAC IRT controller(s).

ADC/DAC FIFO status register

FIFO_STAT_RG registers must be used to obtain current status of EF/PAE/FF/PAF flags for ADC and DAC FIFO. Note, that ADC FIFO status flags are valid for both ADC synchronous FIFO data acquisition modes only (*ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ*), whereas DAC FIFO status flags are valid for both DAC synchronous FIFO data acquisition modes only (*DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ*).

For more details about ADC and DAC FIFO, ADC and DAC data acquisition controllers, refer to section "Data Acquisition Control" later in this chapter.

FIFO_STAT_RG register is available for read-only and is allocated at the least significant byte of 16-bit data word of host PIOX-16 interface.

***FIFO_STAT_RG* register (r)**

x	*DAC_ FIFO_PAE (r,1+)*	*DAC_ FIFO_EF (r,1+)*	*DAC_ FIFO_PAF (r,0+)*	*DAC_ FIFO_FF (r,0+)*	*ADC_ FIFO_PAE (r,1+)*	*ADC_ FIFO_EF (r,1+)*	*ADC_ FIFO_PAF (r,0+)*	*ADC_ FIFO_FF (r,0+)*
bits 15..8	Bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	Bit-1	bit-0

Table 2-9 provides details about *FIFO_STAT_RG* register bits.

Table 2-9. Register bits of *FIFO_STAT_RG* registers.

register bits	**access mode**	**default value on PIOX-16 reset**	**Description**
ADC_FIFO_FF *DAC_FIFO_FF*	r	0	Status of full flag (FF) for ADC and DAC FIFO correspondingly. FF flag is non-programmable. FF flag or each FIFO can be used to generate IRQ-0 and IRQ-1 host PIOX-16 interrupt requests. *xxxx_FIFO_FF*=0 corresponds to non-full condition of the corresponding FIFO. *xxxx_FIFO_FF*=1 corresponds to the full condition of the corresponding FIFO.
ADC_FIFO_PAF *DAC_FIFO_PAF*	r	0	Status of partially full flag (PAF) for ADC and DAC FIFO correspondingly. Offset value for ADC FIFO PAF flag can be programmed during ADC asynchronous data acquisition mode, whereas offset value for DAC FIFO PAF flag can be programmed during DAC asynchronous data acquisition mode. PAF flag of each FIFO can be used to generate IRQ-0 and IRQ-1 host PIOX-16 interrupt requests. *xxxx_FIFO_PAF*=0 corresponds to (2^{18}-M-1) or less number of unread samples inside the corresponding FIFO (M is the programmed offset for FIFO PAF flag). *xxxx_FIFO_PAF*=1 corresponds to (2^{18}-M) or more number of unread samples inside the corresponding FIFO (M is the programmed offset for FIFO PAF flag).

ADC_FIFO_EF *DAC_FIFO_EF*	r	1	Status of empty flag (EF) for ADC and DAC FIFO correspondingly. EF flag is non-programmable. EF flag of each FIFO can be used to generate IRQ-0 and IRQ-1 host PIOX-16 interrupt requests. *xxxx_FIFO_EF*=0 corresponds to non-empty condition of the corresponding FIFO. *xxxx_FIFO_EF*=0 corresponds to the empty condition of the corresponding FIFO.
ADC_FIFO_PAE *DAC_FIFO_PAE*	r	1	Status of partially empty flag (PAE) for ADC and DAC FIFO correspondingly. Offset value for ADC FIFO PAE flag can be programmed during ADC asynchronous data acquisition mode, whereas offset value for DAC FIFO PAE flag can be programmed during DAC asynchronous data acquisition mode. PAE flag of each FIFO can be used to generate IRQ-0 and IRQ-1 host PIOX-16 interrupt requests. *xxxx_FIFO_PAE*=0 corresponds to more than N unread samples inside the corresponding FIFO (N is the programmed offset for FIFO PAE flag). *xxxx_FIFO_PAE*=1 corresponds to N or less unread samples inside the corresponding FIFO (N is the programmed offset for FIFO PAE flag).

Note: 1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

error status (ERR_STAT_RG) and error clear (ERR_CLR_RG) registers

ERR_STAT_RG and *ERR_CLR_RG* registers shall be used correspondingly to read current error status and to clear errors for ADC FIFO during ADC synchronous FIFO data acquisition modes (*ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ*), for DAC FIFO during DAC synchronous FIFO data acquisition modes (*DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ*), and ADC/DAC IRT controllers for all ADC/DAC data acquisition modes.

ERR_STAT_RG and *ERR_CLR_RG* registers are available for read-only and write-only correspondingly and are allocated at the least significant byte of 16-bit data word of host PIOX-16 interface.

***ERR_STAT_RG* register (r)**
***ERR_CLR_RG* register (w)**

X	*0*	*0*	*DAC_ IRT_ERR (r,0+) CLR_DAC_ IRT_ERR (w)*	*ADC_ IRT_ERR (r,0+) CLR_ADC_ IRT_ERR (w)*	*DAC_ FIFO_UNF (r,0+) CLR_DAC_ FIFO_UNF (w)*	*DAC_ FIFO_OVF (r,0+) CLR_DAC_ FIFO_OVF (w)*	*ADC_ FIFO_UNF (r,0+) CLR_ADC_ FIFO_UNF (w)*	*ADC_ FIFO_OVF (r,0+) CR_ADC_ FIFO_OVF (w)*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	Bit-1	bit-0

Table 2-10 provides details about *ERR_STAT_RG* and *ERR_CLR_RG* registers bits.

Table 2-10. Register bits of *ERR_STAT_RG* and *ERR_CLR_RG* registers.

register bits	**access mode**	**default value on PIOX-16 reset**	**Description**
ADC_FIFO_OVF *CLR_ADC_FIFO_OVF*	r w	0 -	During reads this bit returns current status of ADC FIFO overflow error for ADC data acquisition controller. During writes this bit is used to clear ADC FIFO overflow error. Refer to section "Data Acquisition Control" later in this chapter for more details about ADC data acquisition controller. Read-only *ADC_FIFO_OVF* bit can be used to generate host PIOX-16 interrupt requests. Reading *ADC_FIFO_OVF*=0 corresponds to no detected ADC FIFO overflow error. Reading *ADC_FIFO_OVF*=1 corresponds to active ADC FIFO overflow error, which can be cleared by writing '1' to this bit (*CLR_ADC_FIFO_OVF*). Writing *CLR_ADC_FIFO_OVF*=0 has no effect on current state of ADC FIFO overflow error. Writing *CLR_ADC_FIFO_OVF*=1 clears currently active ADC FIFO overflow error.
ADC_FIFO_UNF *CLR_ADC_FIFO_UNF*	r w	0 -	During reads this bit returns current status of ADC FIFO underflow error for ADC data acquisition controller. During writes this bit is used to clear ADC FIFO underflow error. Refer to section "Data Acquisition Control" later in this chapter for more details about ADC data acquisition controller. Read-only *ADC_FIFO_UNF* bit can be used to generate host PIOX-16 interrupt requests. Reading *ADC_FIFO_UNF*=0 corresponds to no detected ADC FIFO underflow error. Reading *ADC_FIFO_UNF*=1 corresponds to active ADC FIFO underflow error, which can be cleared by writing '1' to this bit (*CLR_ADC_FIFO_UNF*). Writing *CLR_ADC_FIFO_UNF*=0 has no effect on current state of ADC FIFO underflow error. Writing *CLR_ADC_FIFO_UNF*=1 clears currently active ADC FIFO underflow error.

DAC_FIFO_OVF *CLR_DAC_FIFO_OVF*	R w	0 -	During reads this bit returns current status of DAC FIFO overflow error for DAC data acquisition controller. During writes this bit is used to clear DAC FIFO overflow error. Refer to section “Data Acquisition Control” later in this chapter for more details about DAC data acquisition controller. Read-only *DAC_FIFO_OVF* bit can be used to generate host PIOX-16 interrupt requests. Reading *DAC_FIFO_OVF*=0 corresponds to no detected DAC FIFO overflow error. Reading *DAC_FIFO_OVF*=1 corresponds to active DAC FIFO overflow error, which can be cleared by writing ‘1’ to this bit (*CLR_DAC_FIFO_OVF*). Writing *CLR_DAC_FIFO_OVF*=0 has no effect on current state of DAC FIFO overflow error. Writing *CLR_DAC_FIFO_OVF*=1 clears currently active DAC FIFO overflow error.
DAC_FIFO_UNF *CLR_DAC_FIFO_UNF*	R w	0 -	During reads this bit returns current status of DAC FIFO underflow error for DAC data acquisition controller. During writes this bit is used to clear DAC FIFO underflow error. Refer to section “Data Acquisition Control” later in this chapter for more details about DAC data acquisition controller. Read-only *DAC_FIFO_UNF* bit can be used to generate host PIOX-16 interrupt requests. Reading *DAC_FIFO_UNF*=0 corresponds to no detected DAC FIFO underflow error. Reading *DAC_FIFO_UNF*=1 corresponds to active DAC FIFO underflow error, which can be cleared by writing ‘1’ to this bit (*CLR_DAC_FIFO_UNF*). Writing *CLR_DAC_FIFO_UNF*=0 has no effect on current state of DAC FIFO underflow error. Writing *CLR_DAC_FIFO_UNF*=1 clears currently active DAC FIFO underflow error.
ADC_IRT_ERR *CLR_ADC_IRT_ERR*	r w	0 -	During reads this bit returns current error status of ADC IRT controller. During writes this bit is used to clear ADC IRT error. Refer to section “ADC IRT and DAC IRT Controllers” later in this chapter for more details about ADC IRT controller. Read-only *ADC_IRT_ERR* bit can be used to generate host PIOX-16 interrupt requests. Reading *ADC_IRT_ERR*=0 corresponds to no detected ADC IRT error. Reading *ADC_IRT_ERR*=1 corresponds to active ADC IRT error, which can be cleared by writing ‘1’ to this bit (*CLR_ADC_IRT_ERR*). Writing *CLR_ADC_IRT_ERR*=0 has no effect on current state of ADC IRT error. Writing *CLR_ADC_IRT_ERR*=1 clears currently active ADC IRT error.

DAC_IRT_ERR *CLR_DAC_IRT_ERR*	r w	0 -	During reads this bit returns current error status of DAC IRT controller. During writes this bit is used to clear DAC IRT error. Refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details about DAC IRT controller. Read-only *DAC_IRT_ERR* bit can be used to generate host PIOX-16 interrupt requests. Reading *DAC_IRT_ERR*=0 corresponds to no detected DAC IRT error. Reading *DAC_IRT_ERR*=1 corresponds to active DAC IRT error, which can be cleared by writing '1' to this bit (*CLR_DAC_IRT_ERR*). Writing *CLR_DAC_IRT_ERR*=0 has no effect on current state of DAC IRT error. Writing *CLR_DAC_IRT_ERR*=1 clears currently active DAC IRT error.

Note:
1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

ADC output data registers

While ADC data acquisition controller is configured in either ADC asynchronous data acquisition mode (*ADC-ASYNC-DAQ*) or in ADC synchronous data acquisition mode with direct ADC output data access via host PIOX-16 interface (*ADC-SYNC-PX-DAQ*), then ADC output data can be accessed directly via host PIOX-16 interface.

ADC0_DATA_RG, *ADC1_DATA_RG*, *ADC2_DATA_RG* and *ADC3_DATA_RG* read-only registers are used to directly read ADC-0, ADC-1, ADC-2 and ADC-3 output data correspondingly from host DSP software via host PIOX-16 interface.

In case ADC IRT controller in conjunction with host DSP on-chip DMA controller are used to download ADC output data to host DSP environment in either *ADC-ASYNC-DAQ* or *ADC-SYNC-PX-DAQ* DAC data acquisition modes, then *ADC_IRT_DATA_RG* read-only register must be specified as the source address without address increment/decrement feature for host DSP on-chip DMA controller in order to read ADC output data from all active ADC channels via ADC IRT controller.

ADC0_DATA_RG, *ADC1_DATA_RG*, *ADC2_DATA_RG*, *ADC3_DATA_RG*, and *ADC_IRT_DATA_RG* read-only registers occupy full 16-bit data words of host PIOX-16 interface with bit D15 being the sign bit.

***ADC0_DATA_RG* register (r)**

ADC0_DATA-15..0 *(r)*

bits 15..0

ADC1_DATA_RG register (r)

ADC1_DATA-15..0 *(r)*
bits 15..0

ADC2_DATA_RG register (r)

ADC2_DATA-15..0 *(r)*
bits 15..0

ADC3_DATA_RG register (r)

ADC3_DATA-15..0 *(r)*
bits 15..0

ADC_IRT_DATA_RG register (r)

ADC_IRT_DATA-15..0 *(r)*
bits 15..0

For more details about ADC data acquisition controller, refer to section “Data Acquisition Control” later in this chapter.

DAC input data registers

While DAC data acquisition controller is configured in either DAC asynchronous data acquisition mode (*DAC-ASYNC-DAQ*) or in DAC synchronous data acquisition mode with direct ADC output data access via host PIOX-16 interface (*DAC-SYNC-PX-DAQ*), then DAC input data can be written directly to DAC via host PIOX-16 interface.

DAC0_DATA_RG, *DAC1_DATA_RG*, *DAC2_DATA_RG* and *DAC3_DATA_RG* write-only registers are used to directly write input data to DAC-0, DAC-1, DAC-2 and DAC-3 correspondingly from host DSP software via host PIOX-16 interface.

In case DAC IRT controller in conjunction with host DSP on-chip DMA controller are used to upload DAC output data to host DSP environment in either *DAC-ASYNC-DAQ* or *DAC-SYNC-PX-DAQ* DAC data acquisition modes, then *DAC_IRT_DATA_RG* write-only register must be specified as the destination address without address increment/decrement feature for host DSP on-chip DMA controller in order to write input data for all active DAC channels via DAC IRT controller.

DAC0_DATA_RG, *DAC1_DATA_RG*, *DAC2_DATA_RG*, *DAC3_DATA_RG*, and *DAC_IRT_DATA_RG* write-only registers occupy full 16-bit data words of host PIOX-16 interface with bit D15 being the sign bit.

DAC0_DATA_RG register (w)

DAC0_DATA-15..0 *(w)*

bits 15..0

DAC1_DATA_RG register (w)

DAC1_DATA-15..0 *(w)*

bits 15..0

DAC2_DATA_RG register (w)

DAC2_DATA-15..0 *(w)*

bits 15..0

DAC3_DATA_RG register (w)

DAC3_DATA-15..0 *(w)*

bits 15..0

DAC_IRT_DATA_RG register (w)

DAC_IRT_DATA-15..0 *(w)*

bits 15..0

For more details about DAC data acquisition controller, refer to section "Data Acquisition Control" later in this chapter.

ADC FIFO data register

While ADC data acquisition controller is configured in either of ADC synchronous FIFO data acquisition modes (*ADC-SYNC-FIFO-OPM-DAQ* or *ADC-SYNC-FIFO-PTM-DAQ*), then temporary buffered real-time ADC output data is available at the ADC FIFO output via *ADC_FIFO_DATA_RG* read-only register. Note, that each read from *ADC_FIFO_DATA_RG* read-only register advances ADC FIFO read pointer, and ADC FIFO data cannot be re-read.

> **CAUTION**
>
> Direct ADC output data via *ADC0_DATA_RG*, *ADC1_DATA_RG*, *ADC2_DATA_RG* and *ADC3_DATA_RG* read-only registers are not available for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* ADC data acquisition modes.

Optional *ADC_FIFO_DATA_MSW_RG* read-only register is provided in order to read optional *ADC_SCCL_EOS* end-of-scan flag status associated with each ADC FIFO output data word.

> **CAUTION**
>
> *ADC_FIFO_DATA_MSW_RG* read-only register contains *ADC_SCCL_EOS* end-of-scan flag status information associated with the last read ADC FIFO data word.
>
> In order to get *ADC_SCCL_EOS* end-of-scan flag status information associated with the last read ADC FIFO data word, then *ADC_FIFO_DATA_MSW_RG* read-only register must be read prior next read from *ADC_FIFO_DATA_RG* read-only register.

In case host DSP on-chip DMA controller synchronized by ADC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM is used to download ADC output data to host DSP environment in either *ADC-SYNC-FIFO-OPM-DAQ* or *ADC-SYNC-FIFO-PTM-DAQ* ADC data acquisition modes, then *ADC_FIFO_DATA_RG* read-only register must be specified as the source address without address increment/decrement feature for host DSP on-chip DMA controller in order to read ADC output data from all enabled ADC channels.

ADC_FIFO_DATA_RG read-only register occupies full 16-bit data words of host PIOX-16 interface with bit D15 being the sign bit, whereas *ADC_FIFO_DATA_MSW_RG* read-only register is allocated at the least significant byte of 16-bit data word of host PIOX-16 interface.

***ADC_FIFO_DATA_RG* register (r)**

ADC_FIFO_DATA-15..0 *(r)*
bits 15..0

***ADC_FIFO_DATA_MSW_RG* register (r)**

X	*0*	*0*	*0*	*0*	*0*	*0*	*ADC_SCCL_EOS* *(r)*	*0*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	Bit-1	bit-0

Table 2-11 provides details about *ADC_FIFO_DATA_MSW_RG* registers bits.

Table 2-11. Register bits of *ADC_FIFO_DATA_MSW_RG* register.

Register bits	**access mode**	**Default value on PIOX-16 reset**	**Description**
ADC_SCCL_EOS	r	-	End-of-scan indicator for last read ADC FIFO data. Refer to section "Data Acquisition Control" later in this chapter for more details about ADC data acquisition controller. *ADC_SCCL_EOS*=0 denotes that last read ADC FIFO data does not correspond to the last scan cycle of ADC sampling packet. *ADC_SCCL_EOS*=1 denotes that last read ADC FIFO data corresponds to the last scan cycle of ADC sampling packet.

Note: 1. Access modes: r/w – read/write; r – read-only; w – write only.

For more details about ADC FIFO and ADC data acquisition controller, refer to section "Data Acquisition Control" later in this chapter.

DAC FIFO data register

While DAC data acquisition controller is configured in either of DAC synchronous FIFO data acquisition modes (*DAC-SYNC-FIFO-OPM-DAQ* or *DAC-SYNC-FIFO-PTM-DAQ*), then input data for DAC is available at DAC FIFO output via *DAC_FIFO_DATA_RG* write-only register. Note, that each write to *DAC_FIFO_DATA_RG* write-only register advances DAC FIFO write pointer, and DAC FIFO data cannot be re-written.

CAUTION

Direct DAC input data via *DAC0_DATA_RG*, *DAC1_DATA_RG*, *DAC2_DATA_RG* and *DAC3_DATA_RG* write-only registers are not available for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* ADC data acquisition modes.

In case DAC IRT controller in conjunction with host DSP on-chip DMA controller are used to upload ADC output data from host DSP environment in either *DAC-SYNC-FIFO-OPM-DAQ* or *DAC-SYNC-FIFO-PTM-DAQ* DAC data acquisition modes, then *DAC_FIFO_DATA_RG* write-only register must be specified as the destination address without address increment/decrement feature for host DSP on-chip DMA controller in order to write DAC input data for all active DAC channels via DAC IRT controller.

DAC_FIFO_DATA_RG write-only register occupies full 16-bit data words of host PIOX-16 interface with bit D15 being the sign bit.

DAC_FIFO_DATA_RG register (w)

DAC_FIFO_DATA-15..0 *(r)*
bits 15..0

ADC/DAC FIFO serial configuration registers

ADC_FIFO_SIN_RG and *DAC_FIFO_SIN_RG* write-only registers shall be used to program the offset values for PAE/PAF flags of ADC and DAC FIFO correspondingly while ADC and/or DAC data acquisition controllers are configured in the corresponding asynchronous data acquisition modes.

ADC and DAC FIFO have been designed to program PAE/PAF flags using serial programming technique via serial-in input (SIN), so *ADC_FIFO_SIN_RG* and *DAC_FIFO_SIN_RG* registers actually appear as 1-bit write-only registers allocated to the least significant bit of 16-bit data word of host PIOX-16 interface.

ADC_FIFO_SIN_RG register (w)

X	*x*	*X*	*X*	*x*	*X*	*X*	*x*	*ADC_FIFO_SIN* *(w)*
Bits 15..8	bit-7	bit-6	Bit-5	bit-4	bit-3	Bit-2	bit-1	bit-0

DAC_FIFO_SIN_RG register (w)

X	*x*	*X*	*X*	*x*	*X*	*X*	*x*	*DAC_FIFO_SIN* *(w)*
Bits 15..8	bit-7	bit-6	Bit-5	bit-4	bit-3	Bit-2	bit-1	bit-0

Table 2-12. Register bits of *ADC_FIFO_SIN_RG* and *DAC_FIFO_SIN_RG* registers.

register bits	**access mode**	**default value on PIOX-16 reset**	**description**
ADC_FIFO_SIN *DAC_FIFO_SIN*	w	-	FIFO serial-in (SIN) input data for serial programming of PAF/PAE flags for ADC and DAC FIFO correspondingly.

Note: 1. Access modes: r/w – read/write; r – read-only; w – write only.

Refer to section "Data Acquisition Control" later in this chapter for more details about how to program PAE/PAF flags of ADC and DAC FIFO.

PFG control registers

The output frequency of on-board high-resolution programmable sampling frequency generator (PFG) can be set from host DSP software via of 21-bit code comprising of V, R, S and X fields (refer to section "Data Acquisition Control" later in this chapter for more details about how to program the PFG output frequency).

PFG V, R, S and X fields can be set by host DSP software using *PFG_CNTR1_RG*, *PFG_CNTR2_RG* and *PFG_CNTR3_RG* registers, which are available for read/write and are allocated at the least significant byte of 16-bit data word of host PIOX-16 interface.

PFG_CNTR1_RG register (r/w)

x	*PFG_V7* *(r/w, 0+)*	*PFG_V6* *(r/w, 0+)*	*PFG_V5* *(r/w, 1+)*	*PFG_V4* *(r/w, 1+)*	*PFG_V3* *(r/w, 1+)*	*PFG_V2* *(r/w, 0+)*	*PFG_V1* *(r/w, 0+)*	*PFG_V0* *(r/w, 0+)*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	Bit-2	bit-1	bit-0

PFG_CNTR2_RG register (r/w)

x	*PFG_R6* *(r/w, 0+)*	*PFG_R5* *(r/w, 1+)*	*PFG_R4* *(r/w, 1+)*	*PFG_R3* *(r/w, 0+)*	*PFG_R2* *(r/w, 0+)*	*PFG_R1* *(r/w, 0+)*	*PFG_R0* *(r/w, 0+)*	*PFG_V8* *(r/w, 0+)*
bits 15..8	bit-7	bit-6	bit-5	bit-4	bit-3	Bit-2	bit-1	bit-0

PFG_CNTR3_RG register (r/w)

x	*0*	*0*	*PFG_X1* *(r/w, 0+)*	*PFG_X1* *(r/w, 0+)*	*0*	*PFG_S2* *(r/w, 0+)*	*PFG_S1* *(r/w, 0+)*	*PFG_S0* *(r/w, 0+)*
bits 15..8	bit-7	Bit-6	bit-5	bit-4	bit-3	Bit-2	bit-1	bit-0

Table 2-13 provides details about *PFG_CNTR1_RG*, *PFG_CNTR2_RG* and *PFG_CNTR3_RG* registers bits.

Table 2-13. Register bits of *PFG_CNTR1_RG*, *PFG_CNTR2_RG* and *PFG_CNTR3_RG* registers.

register bits	**access mode**	**default value on PIOX-16 reset**	**Description**
{*V8..V0*}	r/w	38H	Defines 9-bit V-field for on-board PFG. Refer to section “Data Acquisition Control” later in this chapter for more details about how to program the PFG output frequency.
{*R6..R0*}	r/w	30H	Defines 7-bit R-field for on-board PFG. Refer to section “Data Acquisition Control” later in this chapter for more details about how to program the PFG output frequency.
{*S2..S0*}	r/w	0H	Defines 3-bit S-field for on-board PFG. Refer to section “Data Acquisition Control” later in this chapter for more details about how to program the PFG output frequency.
{*X1..X0*}	r/w	0H	Defines 2-bit X-field for on-board PFG. Refer to section “Data Acquisition Control” later in this chapter for more details about how to program the PFG output frequency.

Note: 1. Access modes: r/w – read/write; r – read-only; w – write only.

CAUTION

Host PIOX-16 interface reset condition will preset *PFG_CNTR1_RG*, *PFG_CNTR2_RG* and *PFG_CNTR3_RG* registers to default values, which correspond to 100 kHz PFG output frequency.

general purpose I/O (GPIO) control

GPIO-0..3 are general purpose I/O pins, which are available via on-board JP3 external synchronization and I/O connector. *GPIO-0..3* I/O pins are controlled by two control registers:

- *GPIO_DIR_RG* register (*GPIO* direction register)
- *GPIO_DATA_RG* register (*GPIO* data register)

GPIO_DIR_RG register must be used to define direction for *GPIO-0..3* I/O pins, whereas *GPIO_DATA_RG* register must be used to define output value and read current status of *GPIO-0..3* I/O pins. *GPIO_DIR_RG* and *GPIO_DATA_RG* registers are available for read/write and are allocated to the least significant byte of 16-bit data word of host PIOX-16 interface.

GPIO_DIR_RG register (r/w)

X	*0*	*0*	*0*	*0*	*GPIO-3_DIR (r/w, 0+)*	*GPIO-2_DIR (r/w, 0+)*	*GPIO-1_DIR (r/w, 0+)*	*GPIO-0_DIR (r/w, 0+)*
bits-15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	Bit-1	bit-0

GPIO_DATA_RG register (r/w)

X	*0*	*0*	*0*	*0*	*GPIO-3_DATA* *(r/w, 0+)*	*GPIO-2_DATA* *(r/w, 0+)*	*GPIO-1_DATA* *(r/w, 0+)*	*GPIO-0_DATA* *(r/w, 0+)*
bits-15..8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	Bit-1	bit-0

Tables 2-14a and 2-14b provides details about *GPIO_DIR_RG* and *GPIO_DATA_RG* registers bits.

Table 2-14a. Register bits of *GPIO_DIR_RG* register.

Register bits	**access mode**	**Default value on PIOX-16 reset**	**Description**
GPIO-0_DIR *GPIO-1_DIR* *GPIO-2_DIR* *GPIO-3_DIR*	r/w	0	Define direction for *GPIO-0..3* I/O pin correspondingly. *GPIO-x_DIR* =0 corresponds to the input direction for the corresponding *GPIO-x* I/O pin. In this case the current state of *GPIO-x* input pin can be read by host DSP software via the *GPIO-x_DATA* bit of *GPIO_DATA_RG* register. All writes to the *GPIO-x_DATA* bit of *GPIO_DATA_RG* register are ignored until the *GPIO-x* I/O pin will be configure as output pin. *GPIO-x_DIR* =1 corresponds to the output direction for the corresponding *GPIO-x* I/O pin. In this case the output value of *GPIO-x* input pin can be set by host DSP software by writing to the *GPIO-x_DATA* bit of *GPIO_DATA_RG* register, whereas current state of *GPIO-x* pin can be read by host DSP software via the *GPIO-x_DATA* bit of *GPIO_DATA_RG* register.

Note:
1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

Table 2-14b. Register bits of *GPIO_DATA_RG* register.

register bits	access mode	Default value on PIOX-16 reset	Description
GPIO-0_DATA *GPIO-1_DATA* *GPIO-2_DATA* *GPIO-3_DATA*	r/w	-	Returns current state of *GPIO-0..3* I/O pins correspondingly during read cycle. Defines output value for *GPIO-0..3* pins during write cycle in case *GPIO-0..3* pins are configured as outputs. Reading *GPIO-x_DATA* =0 corresponds to current '0' state for the corresponding *GPIO-x* I/O pin. Reading *GPIO-x_DATA* =1 corresponds to current '1' state for the corresponding *GPIO-x* I/O pin. Writing *GPIO-x_DATA* =0 sets the output '0' value for the corresponding *GPIO-x* I/O pin. However, the written output value will not take effect until the *GPIO-x* I/O pin will be configured as the output via *GPIO_DIR_RG* register. Writing *GPIO-x_DATA* =1 sets the output '1' value for the corresponding *GPIO-x* I/O pin. However, the written output value will not take effect until the *GPIO-x* I/O pin will be configured as the output via *GPIO_DIR_RG* register.

Note: 1. Access modes: r/w – read/write; r – read-only; w – write only.

CAUTION

In case any of *GPIO-0..3* I/O pins is configured as input, then writing to the corresponding bit of *GPIO_DATA_RG* register is ignored, however data written will be held in *GPIO_DATA_RG* register and will appear on the corresponding *GPIO-0..3* outputs as soon as it (they) will be configured as the output(s).

CAUTION

Active low condition at the *GPIO-0* and/or *GPIO-1* I/O pins can be used to generate host PIOX-16 interrupt requests.

host PIOX-16 interrupt control

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM offers software programmable source selectors and enable control for each of four host PIOX-16 interrupt requests (IRQ-0..3). Host PIOX-16 interrupt requests can be generated on a variety of data acquisition conditions and external events.

Since the total number of possible host DSP interrupt sources for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM is well above four sources, then individual interrupt source selector and enable control for each of host PIOX-16 interrupt requests allow flexibility for run-time host interrupt system configuration for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* in order to meet requirements of virtually any application.

Except for individual selection from a variety of interrupt request sources, *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM also provides enable (tri-state) and polarity control for each of IRQ-0..3 host PIOX-16 interface interrupt request outputs. This allows to use *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM with any *TORNADO* DSP system/controller with shared/non-shared and internally inversed/non-inversed interrupt requests for both host DSP interrupt processing and for ADC/DAC IRT data transfer synchronization.

Selection of interrupt source for IRQ-0..3 host PIOX-16 interrupt requests is performed by host DSP software by means of programming *HIRQ0_SEL_RG..HIRQ3_SEL_RG* interrupt control registers, which are available for read/write and are allocated at the least significant byte of 16-bit data word of host PIOX-16 interface.

HIRQ0_SEL_RG register (r/w)

x	*HIRQ0_EN* *(r/w, 0+)*	*HIRQ0_POL* *(r/w, 0+)*	*0*	*HIRQ0_SEL-4* *(r/w, 0+)*	*HIRQ0_SEL-3* *(r/w, 0+)*	*HIRQ0_SEL-2* *(r/w, 0+)*	*HIRQ0_SEL-1* *(r/w, 0+)*	*HIRQ0_SEL-0* *(r/w, 0+)*
bit-15...8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1	bit-0

HIRQ1_SEL_RG register (r/w)

x	*HIRQ1_EN* *(r/w, 0+)*	*HIRQ1_POL* *(r/w, 0+)*	*0*	*HIRQ1_SEL-4* *(r/w, 0+)*	*HIRQ1_SEL-3* *(r/w, 0+)*	*HIRQ1_SEL-2* *(r/w, 0+)*	*HIRQ1_SEL-1* *(r/w, 0+)*	*HIRQ1_SEL-0* *(r/w, 0+)*
bit-15...8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1	bit-0

HIRQ2_SEL_RG register (r/w)

x	*HIRQ2_EN* *(r/w, 0+)*	*HIRQ2_POL* *(r/w, 0+)*	*0*	*0*	*HIRQ2_SEL-3* *(r/w, 0+)*	*HIRQ2_SEL-2* *(r/w, 0+)*	*HIRQ2_SEL-1* *(r/w, 0+)*	*HIRQ2_SEL-0* *(r/w, 0+)*
bit-15...8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1	bit-0

HIRQ3_SEL_RG register (r/w)

x	*HIRQ3_EN* *(r/w, 0+)*	*HIRQ3_POL* *(r/w, 0+)*	*0*	*0*	*HIRQ3_SEL-3* *(r/w, 0+)*	*HIRQ3_SEL-2* *(r/w, 0+)*	*HIRQ3_SEL-1* *(r/w, 0+)*	*HIRQ3_SEL-0* *(r/w, 0+)*
bit-15...8	bit-7	bit-6	bit-5	Bit-4	bit-3	bit-2	bit-1	bit-0

Table 2-15 provides details about *HIRQ0_SEL_RG* .. *HIRQ3_SEL_RG* registers bits.

Table 2-15. Register bits of *HIRQ0_SEL_RG*, *HIRQ1_SEL_RG*, *HIRQ2_SEL_RG* and *HIRQ3_SEL_RG* registers.

Register bits	**access mode**	**default value on PIOX-16 reset**	**Description**

{*HIRQ0_SEL-4.. HIRQ0_SEL-0*}
{*HIRQ1_SEL-4.. HIRQ1_SEL-0*}
{*HIRQ2_SEL-3.. HIRQ2_SEL-0*}
{*HIRQ3_SEL-3.. HIRQ3_SEL-0*}
	r/w	0H	Select interrupt source for the corresponding host PIOX-16 interrupt request output in accordance with table 2-16.

HIRQ0_EN *HIRQ1_EN* *HIRQ2_EN* *HIRQ3_EN*	r/w	0	Enable (tri-state) control for the corresponding host PIOX-16 interrupt request output. *HIRQx-_EN* =0 corresponds to disabled IRQ-x host PIOX-16 interrupt request, i.e. IRQ-x host PIOX-16 interrupt request output is tri-stated is in inactive state. *HIRQx-_EN* =1 corresponds to active IRQ-x host PIOX-16 interrupt request output with an interrupt source being selected with the corresponding {*HIRQx_SEL-4..HIRQx_SEL-0*} register field and interrupt output polarity being defined by the corresponding *HIRQx_POL* register bit.
HIRQ0_POL *HIRQ1_POL* *HIRQ2_POL* *HIRQ3_POL*	r/w	0	Defines polarity (active signal level) or the corresponding host PIOX-16 interrupt request output. This bit must normally stay in default state except when the corresponding interrupt request output is used for ADC/DAC IRT data transfer control for *TORNADO* DSP systems/controllers. . Refer to section “ADC IRT and DAC IRT Controllers” later in this chapter for more details about ADC/DAC IRT controllers. *HIRQx-_POL* =0 corresponds to the positive polarity of IRQ-x host PIOX-16 interrupt request output. Since IRQ-0..3 inputs of host PIOX-16 interface are normally active low, then this setting will correspond to active low IRQ-x host PIOX-16 interrupt request output. This is default setting in case IRQ-x host PIOX-16 interrupt request output is used to interrupt request to host DSP of *TORNADO* DSP system/controller, and this is default setting for most *TORNADO* DSP systems/controllers in case IRQ-x host PIOX-16 interrupt request output is used to transfer ADC/DAC IRT synchronization event to host DSP on-chip DMA controller. *HIRQx-_POL* =1 corresponds to the negative (inverse) polarity of IRQ-x host PIOX-16 interrupt request output, i.e. active high IRQ-x host PIOX-16 interrupt request output. This setting must be used for some *TORNADO* DSP systems/controllers only in case IRQ-x host PIOX-16 interrupt request output is used to transfer ADC/DAC IRT synchronization event to host DSP on-chip DMA controller.

Note:
1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

Host PIOX-16 interface IRQ-0 and IRQ-1 interrupt source selectors of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM allow selection of one interrupt request source from 21 available interrupt request sources in accordance with full table 2-16, whereas host PIOX-16 interface IRQ-2 and IRQ-3 interrupt source selectors allow selection of one interrupt request source from 11 available interrupt request sources in accordance with the upmost part of table 2-16 only.

CAUTION

HIRQ0_SEL_RG..HIRQ3_SEL_RG interrupt request selector registers default to the 00H state on host PIOX-16 interface reset condition, which corresponds to disabled interrupt request source, positive output interrupt request polarity, and disabled interrupt request outputs.

Table 2-16. Interrupt request sources for IRQ-0 and IRQ-1 host PIOX-16 interrupt requests.

bits #4..#0 of HIRQ0_SEL_RG and HIRQ1_SEL_RG or bits #3..#0 of HIRQ0_SEL_RG and HIRQ1_SEL_RG					**interrupt request source**
bit #4	**bit #3**	**bit #2**	**bit #1**	**bit #0**	
common interrupt request sources for all IRQ-0..3 host PIOX-16 interrupt requests					
0	*0*	*0*	*0*	*0*	Interrupt is disabled. This is the default value on PIOX-16 reset condition.
0	*0*	*0*	*0*	*1*	Interrupt on the end of ADC synchronous data acquisition process for ADC synchronous FIFO one-pass data acquisition (*ADC-SYNC-FIFO-OPM-DAQ*), i.e. when bit *ADC_SYNC_DAQ_END* of *DAQ_CNTR2_RG* register is set to the '1' state.
0	*0*	*0*	*1*	*0*	Interrupt on the end of DAC synchronous data acquisition process for DAC synchronous FIFO one-pass data acquisition (*DAC-SYNC-FIFO-OPM-DAQ*), i.e. when bit *DAC_SYNC_DAQ_END* of *DAQ_CNTR2_RG* register is set to the '1' state.
0	*0*	*0*	*1*	*1*	Interrupt on detection of start synchronization event for ADC synchronous data acquisition process, i.e. when bit *ADC_SYNC_DAQ_SSYNC_OK* of *DAQ_CNTR2_RG* register is set to the '1' state.
0	*0*	*1*	*0*	*0*	Interrupt on detection of start synchronization event for DAC synchronous data acquisition process, i.e. when bit *DAC_SYNC_DAQ_SSYNC_OK* of *DAQ_CNTR2_RG* register is set to the '1' state.
0	*0*	*1*	*0*	*1*	Interrupt from ADC IRT controller (ADC IRT synchronization event).
0	*0*	*1*	*1*	*0*	Interrupt from DAC IRT controller(DAC IRT synchronization event).

0	*0*	*1*	*1*	*1*	Interrupt on active low condition at *GPIO-0* I/O pin.
0	*1*	*0*	*0*	*0*	Interrupt on active low condition at *GPIO-1* I/O pin.
0	*1*	*0*	*0*	*1*	Interrupt on logical OR of ADC/DAC FIFO and IRT error conditions in accordance with *XIM_ERR_RG* register.
0	*1*	*0*	*1*	*0*	Interrupt on detection of sampling frequency event (Fs).
0	*1*	*0*	*1*	*1*	Interrupt on ADC output data ready condition.
interrupt request sources for IRQ-0 and IRQ-1 host PIOX-16 interrupt requests only					
0	*1*	*1*	*1*	*0*	Interrupt at the end of ADC IRT data transfer cycle.
0	*1*	*1*	*1*	*1*	Interrupt at the end of DAC IRT data transfer cycle.
1	*0*	*0*	*0*	*0*	Interrupt on ADC FIFO full condition, i.e. when FF flag of ADC FIFO is set to the ‘1’ state.
1	*0*	*0*	*0*	*1*	Interrupt on ADC FIFO partially full condition, i.e. when PAF flag of ADC FIFO is set to the ‘1’ state.
1	*0*	*0*	*1*	*0*	Interrupt on ADC FIFO empty condition, i.e. when EF flag of ADC1-FIFO is set to the ‘1’ state.
1	*0*	*0*	*1*	*1*	Interrupt on ADC FIFO partially empty condition, i.e. when PAE flag of ADC1-FIFO is set to the ‘1’ state.
1	*0*	*1*	*0*	*0*	Interrupt on DAC FIFO full condition, i.e. when FF flag of DAC FIFO is set to the ‘1’ state.
1	*0*	*1*	*0*	*1*	Interrupt on DAC FIFO partially full condition, i.e. when PAF flag of DAC FIFO is set to the ‘1’ state.
1	*0*	*1*	*1*	*0*	Interrupt on DAC FIFO empty condition, i.e. when EF flag of DAC FIFO is set to the ‘1’ state.
1	*0*	*1*	*1*	*1*	Interrupt on DAC FIFO partially empty condition, i.e. when PAE flag of DAC FIFO is set to the ‘1’ state.

Notes:
1. Unlisted combinations are reserved and will result in disabled interrupt.
2. Highlighted selection corresponds to default setting on host PIOX-16 interface condition.

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM allows generation of host PIOX-16 interrupt requests on the logical OR of ADC/DAC FIFO and IRT error conditions in order to perform real-time monitoring of any combination of error conditions.

In case any of *HIRQ0_SEL_RG..HIRQ3_SEL_RG* registers is configured to select logical OR of ADC/DAC FIFO and IRT error conditions (bits #4..#0 or *HIRQ0_SEL_RG..HIRQ3_SEL_RG* are set to the {1,0,0,1} state), then the logical OR of error conditions, which appear in *ERR_STAT_RG* register, is generated using the *XIM_ERR_RG* expansion error interrupt mask register, which is available for read/write and is allocated at the least significant byte of 16-bit data word of host PIOX-16 interface.

XIM_ERR_RG register (r/w)

X	*0*	*0*	*XIM_DAC_ IRT_ERR (r/w,0+)*	*XIM_ADC_ IRT_ERR (r/w,0+)*	*XIM_DAC_ FIFO_UNF (r/w,0+)*	*XIM_DAC_ FIFO_OVF (r/w,0+)*	*XIM_ADC_ FIFO_UNF (r/w,0+)*	*XIM_ADC_ FIFO_OVF (r/w,0+)*
Bits 15..8	bit-7	bit-6	Bit-5	bit-4	Bit-3	bit-2	Bit-1	bit-0

CAUTION

Host PIOX-16 interrupt request on logical OR of error conditions is generated as logical OR of the corresponding enabled (unmasked) error conditions as they appear in *ERR_STAT_RG* register.

Table 2-17 provides details about *XIM_ERR_RG* register bits.

Table 2-17. Register bits of *XIM_ERR_RG* register.

register bits	**access mode**	**default value on PIOX-16 reset**	**Description**
XIM_ADC_FIFO_OVF	r/w	0	Expansion interrupt mask for ADC FIFO overflow error (bit *ADC_FIFO_OVF* of *ERR_STAT_RG* register). *XIM_ADC_FIFO_OVF*=0 corresponds to disabled host PIOX-16 interrupt request on ADC FIFO overflow error condition. *XIM_ADC_FIFO_OVF*=1 corresponds to enabled host PIOX-16 interrupt request on ADC FIFO overflow error condition.
XIM_ADC_FIFO_UNF	r/w	0	Expansion interrupt mask for ADC FIFO underflow error (bit *ADC_FIFO_UNF* of *ERR_STAT_RG* register). *XIM_ADC_FIFO_UNF*=0 corresponds to disabled host PIOX-16 interrupt request on ADC FIFO underflow error condition. *XIM_ADC_FIFO_UNF*=1 corresponds to enabled host PIOX-16 interrupt request on ADC FIFO underflow error condition.
XIM_DAC_FIFO_OVF	r/w	0	Expansion interrupt mask for DAC FIFO overflow error (bit *DAC_FIFO_OVF* of *ERR_STAT_RG* register). *XIM_DAC_FIFO_OVF*=0 corresponds to disabled host PIOX-16 interrupt request on DAC FIFO overflow error condition. *XIM_DAC_FIFO_OVF*=1 corresponds to enabled host PIOX-16 interrupt request on DAC FIFO overflow error condition.

XIM_DAC_FIFO_UNF	r/w	0	Expansion interrupt mask for DAC FIFO underflow error (bit *DAC_FIFO_UNF* of *ERR_STAT_RG* register). *XIM_DAC_FIFO_UNF*=0 corresponds to disabled host PIOX-16 interrupt request on DAC FIFO underflow error condition. *XIM_DAC_FIFO_UNF*=1 corresponds to enabled host PIOX-16 interrupt request on DAC FIFO underflow error condition.
XIM_ADC_IRT_ERR	r/w	0	Expansion interrupt mask for ADC IRT error (bit *ADC_IRT_ERR* of *ERR_STAT_RG* register). *XIM_ADC_IRT_ERR*=0 corresponds to disabled host PIOX-16 interrupt request on ADC IRT error condition. *XIM_ADC_IRT_OVF*=1 corresponds to enabled host PIOX-16 interrupt request on ADC IRT error condition.
XIM_DAC_IRT_ERR	r/w	0	Expansion interrupt mask for DAC IRT error (bit *DAC_IRT_ERR* of *ERR_STAT_RG* register). *XIM_DAC_IRT_ERR*=0 corresponds to disabled host PIOX-16 interrupt request on DAC IRT error condition. *XIM_DAC_IRT_OVF*=1 corresponds to enabled host PIOX-16 interrupt request on DAC IRT error condition.

Note:
1. Access modes: r/w – read/write; r – read-only; w – write only.
2. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

2.3 Data Acquisition Control

On-board DAQCU data acquisition and control unit of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, which provides ADC/DAC data acquisition and ADC/DAC IRT control, is configured and controlled via a set of control registers (refer to section “Host PIOX-16 Interface” earlier in this chapter and to table 2-1):

- *DAQ_CNTR1_RG* register, which is used to select ADC/DAC data acquisition modes and options
- *DAQ_CNTR2_RG* register, which is used for real-time control of ADC/DAC data acquisition process during synchronous ADC/DAC data acquisition modes
- *DAQ_CNTR3_RG* register, which is used to select ADC/DAC data acquisition termination flags and interrupt retriggable transmission (IRT) flags for ADC/DAQ synchronous FIFO data acquisition modes
- *DAQ_SYNC_RG* register, which is used to select sampling frequency source, ADC/DAQ data acquisition start synchronization mode, external synchronization mode, and ADC timing options
- *ADC_CNF_RG* register, which is used to select ADC channels, which will take part in ADC data acquisition
- *DAC_CNF_RG* register, which is used to select DAC channels, which will take part in DAC data acquisition
- *ADCQ_IMUX_CNF1_RG* and *ADCQ_IMUX_CNF2_RG* registers, which are used to configure ADC input multiplexers for all ADC channels
- *FIFO_STAT_RG* registers, which indicate current flags status of ADC/DAC FIFO for ADC/DAC synchronous FIFO data acquisition modes

- *ADC_DAQ_RESET_RG*, *DAC_DAQ_RESET_RG* and *ADDA_DAQ_RESET_RG* registers, which are used to reset ADC and DAC data acquisition controllers
- *ADC_START_RG*, *DAC_OUTPUT_LD_RG* and *ADC_START_DAC_OUTPUT_LD_RG* registers, which are used to start A/D conversion in ADC asynchronous data acquisition mode and to load DAC chip outputs in DAC asynchronous data acquisition mode.
- *ADC0_DATA_RG*, *ADC1_DATA_RG*, *ADC2_DATA_RG* and *ADC3_DATA_RG* registers, which are used to directly read ADC-0..3 output data for ADC asynchronous data acquisition mode (*ADC-ASYNC-DAQ*) and ADC synchronous data acquisition mode with direct ADC output data access via host PIOX-16 interface (*ADC-SYNC-PX-DAQ*)
- *DAC0_DATA_RG*, *DAC1_DATA_RG*, *DAC2_DATA_RG* and *DAC3_DATA_RG* registers, which are used to directly write DAC-0..3 input data for DAC asynchronous data acquisition mode (*DAC-ASYNC-DAQ*) and DAC synchronous data acquisition mode with direct DAC input data access via host PIOX-16 interface (*DAC-SYNC-PX-DAQ*)
- *ADC_FIFO_DATA_RG* and *DAC_FIFO_DATA_RG* registers, which are used to read ADC FIFO output data and to write to DAC FIFO input data
- *ADC_FIFO_SIN_RG* and *DAC_FIFO_SIN_RG* registers, which are used to program offset values for FIFO flags
- *ERR_STAT_RG* and *ERR_CLR_RG* registers, which are used to indicate current data acquisition errors and to clear these errors
- *PFG_CNTR1_RG*, *PFG_CNTR2_RG* and *PFG_CNTR3_RG* registers, which are used to configure on-board high-resolution programmable sampling frequency generator (PFG).

This section provides details about ADC and DAC data acquisition controllers, ADC and DAC data acquisition modes, and how to configure and control ADC and DAC data acquisition processes for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

analog input section

Analog input section of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (fig.2-1) comprises of four identical ADC channels (#0..#3), and is designed for synchronous analog-to-digital conversion of selected quad analog inputs from sixteen available analog inputs (AIN-0..15).

Upon the particular ADC data acquisition mode selected, ADC-0..3 output data can be further either read directly via host PIOX-16 interface, or are pushed into ADC FIFO.

Each of four ADC channels can sample and perform A/D conversion of one of four assigned analog inputs, and all four ADC channels feature synchronous sampling of selected analog inputs. Each ADC channel comprises of the following components:

- four analog inputs (either of AIN-0..3, AIN-4..7, AIN-8..11, and AIN-12..15 correspondingly), which are available at JP2 analog I/O connector
- four analog input amplifiers/buffers (either of AIN-A0..3, AIN-A4..7, AIN-A8..11, and AIN-A12..15 correspondingly), one per each analog input
- software configured as either 4:1/SE single-ended or 2:1/DIFF differential analog input multiplexer (either of AIN-MUX0..3 correspondingly)
- 16-bit 1 Msps ADC chip (either of ADC-0..3 correspondingly).

analog inputs and analog input amplifiers

All sixteen analog inputs (AIN-0..15) feature individual analog input amplifiers/buffers (AIN-A0..15) with gain factor 0 dB, which are used to interface to ± 5 V analog input signal range and to provide 1 MOhm @ 75 pF high input impedance and low input capacitance of analog inputs at extremely low input offset voltage. In order to exclude damage of on-board analog circuits, overvoltage protection at ± 6.5 V voltage clipping level is provided for each analog input.

Tremendous efforts have been done during design of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM in order to maximize channel-to-channel isolation (i.e. to minimize channel-to-channel signal penetration) for analog inputs while keeping analog input impedance at 1 MOhm high value. However, although *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM provides excellent channel-to-channel isolation for non-neighbor analog inputs (for example for AIN-4 and AIN-10 inputs, etc), due to the compact design of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM and extremely high resolution of used A/D converters, it has been found impossible to provides infinite channel-to-channel isolation for the neighbor analog inputs (for example for AIN-4 and AIN-5 inputs, for AIN-4 and AIN-3 inputs, etc), which strongly depends upon the output impedance of connected signal sources and signal frequency. Thus, providing connected output signal source impedance below 100 Ohm will deliver channel-to-channel signal penetration to this analog input from neighbor analog inputs to typically better than –88dB level for 10 kHz input signals.

CAUTION

In order to minimize channel-to-channel signal penetration for analog inputs, keep output impedance of connected analog signal sources as low as possible for all connected analog inputs. It is highly recommended to connect not used analog inputs to analog ground.

The physical nature of channel-to-channel signal penetration for neighbor analog inputs is the electromagnetic interference via JP2 on-board analog input connector. Thus, the larger is the difference between analog input numbers, the better is channel-to-channel isolation between these analog inputs.

It is important to note, that external analog I/O board (*T/X-XIOB/PDAS1M*), flat cable for connection *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM and *T/X-XIOB/PDAS1M* external analog I/O board, and external analog I/O cable (*T/X-AIOCS/PDAS1M*), which are all used to connect *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM to external analog I/O world (refer to fig.1-5 and Appendix C for more details), all decrease channel-to-channel isolation for analog inputs specified separately for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. Therefore, input impedance of all analog inputs is automatically decreased to 33 kOhm in case external analog I/O board (*T/X-XIOB/PDAS1M*) is connected to *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM via flat cable installed into JP2 on-board connector. This solution works well for low-impedance connected analog input sources, but provides limitation when high-impedance analog input sources are used.

CAUTION

In order to minimize channel-to-channel signal penetration for analog inputs for high impedance analog input sources, connect external analog input sources directly to *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM on-board JP2 analog I/O connector using individual shielded cables for each analog input source and extra cable plug.

Extra mated plugs for connecting to *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM on-board JP2 analog I/O connector are available either upon request from MicroLAB Systems or from Samtec Inc (www.samtec.com, p/n # TCSD-20-01-N).

analog input multiplexers

Each of four on-board ADC channels include analog input multiplexer (AIN-MUX0..3 correspondingly), which is used to select which particular analog input will be routed to the input of ADC chip of this ADC channel for A/D conversion. Analog input multiplexers of all ADC channels are always configured identically either by host DSP software in ADC asynchronous data acquisition mode, or by ADC data acquisition controller in each scan cycle of ADC sampling packet for all ADC synchronous data acquisition modes.

Analog input multiplexers of all ADC channels can be configured as either 4:1/SE single-ended or 2:1/DIFF differential analog input multiplexers (figures 2-3a and 2-3b). This allows to select either one of four available single-ended analog inputs or one of two available differential analog inputs for each ADC channel, and this configuration can be updated even in each scan cycle of ADC sampling packet for all ADC synchronous data acquisition modes.

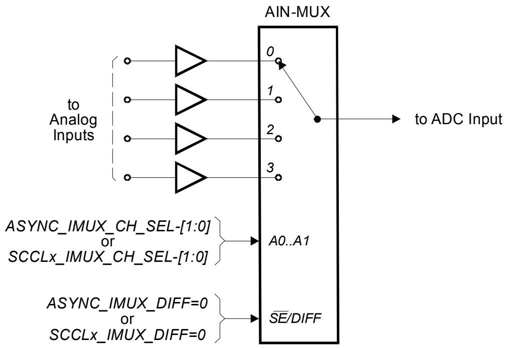

Fig.2-3a. Single-ended configuration of analog input multiplexer (4:1/SE) for each ADC channel.

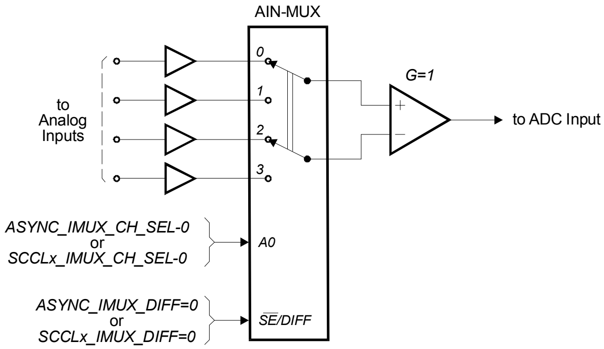

Fig.2-3b. Differential configuration of analog input multiplexer (2:1/DIFF) for each ADC channel.

For ADC asynchronous data acquisition mode, analog input multiplexers are configured by host DSP software via bits {*ASYNC_IMUX_CH_SEL*-[1:0]} and *ASYNC_IMUX_DIFF* of *ADC_IMUX_CNF1_RG* register (refer to table 2-8a). Bit *ASYNC_IMUX_DIFF* selects between 4:1/SE or 2:1/DIFF configuration of analog input multiplexers, whereas bits {*ASYNC_IMUX_CH_SEL*-[1:0]} select particular either SE or DIFF analog input, which will be routed to analog input of ADC chip. Note, that for 2:1/DIFF configuration of analog input multiplexers, only bit *ASYNC_IMUX_CH_SEL-0* is used to select particular one of two available differential analog inputs, and bit *ASYNC_IMUX_CH_SEL-1* is ignored.

For all ADC synchronous data acquisition modes, analog input multiplexers are configured by ADC data acquisition controller for each scan cycle inside ADC sampling packet in accordance with bits {*SCCLx_IMUX_CH_SEL*-[1:0]} and *SCCLx_IMUX_DIFF* (*x*=0..3, denotes scan cycle number) of *ADC_IMUX_CNF1_RG* and *ADC_IMUX_CNF2_RG* registers (refer to table 2-8b), which shall be set by host DSP software prior initializing ADC synchronous data acquisition process. Similar to ADC asynchronous data acquisition mode, bits *SCCLx_IMUX_DIFF* select between 4:1/SE or 2:1/DIFF configuration of analog input multiplexers for particular scan cycles in ADC sampling packet, whereas bits {*SCCLx_IMUX_CH_SEL*-[1:0]} select particular either SE or DIFF analog input, which will be routed to analog input of ADC chip for particular scan cycles in ADC sampling packet. Note, that for 2:1/DIFF configuration of analog input multiplexers, only bit *SCCLx_IMUX_CH_SEL-0* is used to select particular one of two available differential analog inputs, and bit *SCCLx_IMUX_CH_SEL-1* is ignored.

Table 2-18 below provides details about what particular analog input channels are selected for different configurations of analog input multiplexers for all ADC data acquisition modes.

Table 2-18. Analog input multiplexer configurations.

bit ***ASYNC_IMUX_DIFF*** **or** ***SCCLx_IMUX_DIFF***	**bit** ***ASYNC_IMUX_CH_SEL-1*** **or** ***SCCLx_IMUX_CH_SEL-1***	**bit** ***ASYNC_IMUX_CH_SEL-0*** **or** ***SCCLx_IMUX_CH_SEL-0***	**Analog inputs selected for ADC channels**			
			ADC-0	***ADC-1***	***ADC-2***	***ADC-3***
Single-ended configuration of AIN-MUX0..3						
0	0	0	AIN-0	AIN-4	AIN-8	AIN-12
0	0	1	AIN-1	AIN-5	AIN-9	AIN-13
0	1	0	AIN-2	AIN-6	AIN-10	AIN-14
0	1	1	AIN-3	AIN-7	AIN-11	AIN-15
Differential configuration of AIN-MUX0..3						
1	X	0	(AIN-0 - AIN-2)	(AIN-4 - AIN-6)	(AIN-8 - AIN-10)	(AIN-12 - AIN-14)
1	X	1	(AIN-1 - AIN-3)	(AIN-5 - AIN-7)	(AIN-9 - AIN-11)	(AIN-13 - AIN-15)

Note: 1. Highlighted configuration correspond to default settings on PIOX-16 reset condition.

Analog input multiplexers feature small output signal settling time, which together with ADC chip sampling time is about 0.8 µS, that well matches maximum throughoutput ADC channel performance of 1 Msps during pipelined operation.

A/D converter chip (ADC chip)

On-board ADC chips feature 16-bit resolution and excellent linearity at up to 1 MHz sampling frequency, which guarantee ultra-low signal distortion during A/D conversion at high sampling frequencies.

All four on-board ADC chips provide synchronous acquisition (sampling) of analog input signals, which are routed from the outputs of corresponding analog input multiplexers.

CAUTION

ADC-only acquisition (sampling) time, which required for proper acquisition of ADC analog inputs to sufficient accuracy, is 0.25 µS.

Summary “ADC+ADC_IMUX”acquisition time, which includes both acquisition time for both ADC analog input multiplexers and ADC-only acquisition time, and which is required to properly acquire ADC analog inputs after ADC input multiplexer have been updated, is 0.8 µS.

On-board ADC chips can be configured to operate either in high-speed mode or in normal mode, which is defined via bit *ADC_HIGH_SPEED_EN* of *DAQ_SYNC_RG* register (table 2-5).

In case bit *ADC_HIGH_SPEED_EN* of *DAQ_SYNC_RG* register is set into the ‘0’ state (default value on host PIOX-16 interface reset condition), then all four ADC chips are configured in ADC normal speed mode. ADC normal mode is the recommended settings for sampling frequencies below 10 kHz.

CAUTION

Although maximum sampling frequency for ADC normal mode is 800 kHz, however this mode provides valid ADC output data starting from the first ADC sampling/scan cycle, and there is no minimum value restrictions for ADC sampling frequency value, i.e. ADC sampling frequency can be as low as 0 Hz.

In case bit *ADC_HIGH_SPEED_EN* of *DAQ_SYNC_RG* register is set into the ‘1’ state, then all four ADC chips are configured in ADC high-speed mode. Sampling frequency for ADC normal mode is as high as 1 MHz, however there are several restrictions applied for this mode. ADC high-speed mode is the recommended settings for sampling frequencies above 10 kHz.

CAUTION

For *ADC high-speed mode*, ADC output data, which correspond to the first A/D conversion cycle, shall be discarded in case time elapsed after previous A/D conversion exceeds 1 mS value.

For *ADC high-speed mode*, minimum sampling frequency is 1 kHz.

ADC output data read path

Upon particular ADC data acquisition mode selected via ADC data acquisition controller, ADC output data can be either read directly via ADC output data buffers (ADC-ODB) by host PIOX-16 interface (refer to fig.2-1), or can be pushed into on-board high-density ADC FIFO, which can be further read via host PIOX-16 interface.

ADC0_DATA_RG..ADC3_DATA_RG read-only registers of host PIOX-16 interface (refer to table 2-1 and section 2-2 earlier in this chapter) shall be used in order to read real-time ADC-0..3 output data correspondingly for ADC asynchronous data acquisition mode (*ADC-ASYNC-DAQ*) and ADC synchronous with direct PIOX-16 access data acquisition mode (*ADC-SYNC-PX-DAQ*), which allow direct read of ADC output data from host PIOX-16 interface. In these modes, *ADC_IRT_DATA_RG* read-only register must be used by host DSP on-chip DMA controller in order to download real-time ADC output data for all enabled ADC channels using ADC IRT controller (refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details).

ADC_FIFO_DATA_RG and *ADC_FIFO_DATA_MSW_RG* read-only registers of host PIOX-16 interface (refer to table 2-1 and section 2-2 earlier in this chapter) shall be used in order to read temporary stored ADC-0..3 real-time output data from ADC FIFO for ADC synchronous FIFO data acquisition modes (*ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ*), which do not allow direct read of ADC output data from host PIOX-16 interface and perform temporary buffering of ADC real-time output data in on-board high-density ADC FIFO. In these modes, *ADC_FIFO_DATA_RG* read-only register must be also used by host DSP on-chip DMA controller in order to download real-time ADC output data for all enabled ADC channels using ADC IRT controller (refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details).

analog output section

Analog output section of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM comprises of four identical DAC channels (#0..#3), and is designed for digital-to-analog conversion of input data from either host PIOX-16 interface or from DAC FIFO to analog output signals (AOUT-0..3).

Each DAC channel can perform D/A conversion to one analog output, and all D/A channels feature synchronous DAC output data load (sampling).

Each DAC channel comprises of the following components:

- 16-bit 1 Msps DAC chip (either of DAC-0..3 correspondingly)
- analog output amplifier (either of AOUT-A0..3 correspondingly) and analog output (AOUT-0..3 correspondingly).

D/A converter chips (DAC chips)

On-board DAC chips feature 16-bit resolution and excellent linearity at up to 1 MHz sampling frequency, which guarantee ultra-low output signal distortions at high sampling frequencies. There is no minimum value restriction for DAC sampling frequency value, which can be as low as 0 Hz. All four on-board DAC chips provide synchronous sampling of output signal.

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM uses double-buffered DAC chips in order to provide flexible support for different DAC output signal sampling modes. Internal D/A converter chip architecture is presented at figure 2-4.

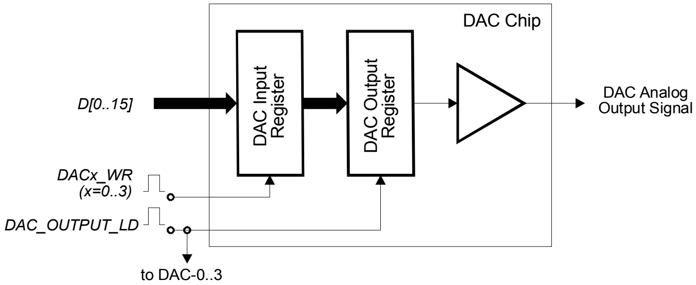

Fig.2-4. Internal D/A converter chip architecture.

DAC chip input data from either host PIOX-16 interface or from DAC FIFO is written into DAC chip input register for each DAC chips individually when *DACx_WR* data strobe signal comes active. Individual *DACx_WR* data write strobes for each DAC chip are generated by either host PIOX-16 interface controller, or DAC IRT controller, or DAC data acquisition controller depending upon what particular DAC must be loaded with new data. However, in order to update DAC output analog signal in accordance with new written DAC data in DAC input data register, common *DAC_OUTPUT_LD* data strobe must be issued by DAC data acquisition controller in order to copy contents of DAC input register to DAC output register.

Double-buffered internal DAC architecture is used by *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM in order to separate DAC input data write process from actual D/A conversion (D/A output signal sampling) and to provide flexibility in selection of different D/A output signal sampling mode.

DAC_OUTPUT_LD data strobe, which is common for all DAC channels, is generated by DAC data acquisition controller in accordance with DAC data acquisition mode and the state of {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register (table 2-2). Refer to subsection "DAC data acquisition controller" later in this section for more details.

DAC input data write path

Upon particular DAC data acquisition mode selected via DAC data acquisition controller, DAC input data can be either written directly via DAC input data buffers (DAC-IDB) by host PIOX-16 interface (refer to fig.2-1), or can be extracted from on-board high-density DAC FIFO, which must be previously written via host PIOX-16 interface.

DAC0_DATA_RG..DAC3_DATA_RG write-only registers of host PIOX-16 interface (refer to table 2-1 and section 2-2 earlier in this chapter) shall be used in order to write real-time DAC-0..3 input data correspondingly for DAC asynchronous data acquisition mode (*DAC-ASYNC-DAQ*) and DAC synchronous with direct PIOX-16 access data acquisition mode (*DAC-SYNC-PX-DAQ*), which allow direct write of DAC input data from host PIOX-16 interface. In these modes, *DAC_IRT_DATA_RG* write-only register must be used by host DSP on-chip DMA controller in order to upload real-time DAC input data for all enabled DAC channels using DAC IRT controller (refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details).

DAC_FIFO_DATA_RG write-only register of host PIOX-16 interface (refer to table 2-1 and section 2-2 earlier in this chapter) must be used in order to write temporary stored DAC-0..3 real-time input data to DAC FIFO for DAC synchronous FIFO data acquisition modes (*DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ*), which do not allow direct write of DAC input data from host PIOX-16 interface and perform temporary buffering of DAC real-time input data in on-board high-density DAC FIFO. In these modes, *DAC_FIFO_DATA_RG* write-only register must be used by host DSP on-chip DMA controller in order to upload real-time DAC input data for all enabled DAC channels using DAC IRT controller (refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details).

analog output amplifiers and analog outputs

Analog output amplifiers are used to buffer analog outputs of D/A converter chips and to protect D/A converter chips from possible damage due to misuse of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

Analog output amplifiers feature excellent linearity, ultra-low offset voltage, high dynamic parameters, output short circuit protection, and operate at load impendence as low as 600 Ohm. Analog output voltage range for each DAC channel is ± 5 V.

Analog outputs (*AOUT-0..3*) of the corresponding DAC channels of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM are available via on-board JP2 analog I/O connector.

sampling frequency selector

Sampling frequency is common for ADC and DAC data acquisition controllers and can be configure by host DSP software via bits {*FS_SEL-1, FS_SEL-0*} of *DAQ_SYNC_RG* register (table 2-5) to select from the following sources:

- on-board high-resolution programmable sampling frequency generator (PFG)
- external sampling frequency input (*XFs*) from on-board JP3 connector
- any of two timer inputs of host PIOX-16 interface (TM/XIO-0 and TM/XIO-1).

On-board PFG high-resolution programmable sampling frequency generator is the recommended selection for sampling frequency source. PFG can be configured by host DSP software to provide sampling frequency within 16 Hz .. 1 MHz range with high resolution setting. For more details about PFG refer to the corresponding subsection later in this section.

External sampling frequency input (*XFs*) is available at JP3 external synchronization connector (refer to Appendix A and Appendix C) and feature low impedance 110 Ohm logical input in order to minimize signal distortions when connecting to external signal sources.

CAUTION

External sampling frequency input (*XFs*) of JP3 connector is active low input with the sampling frequency event detected at the falling edge of *XFs* input signal.

Minimum duration of active low state of signal at external sampling frequency input (*XFs*) of JP3 connector must be at least 200 nS in order to provide proper detection of sampling frequency event.

Host PIOX-16 interface timer pins (TM/XIO-0 and TM/XIO-1) can be also used as the sampling frequency source for ADC and DAC acquisition controllers of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. However, since these timer pins outputs are outputs of on-board DSP chip of host *TORNADO* DSP system/controller, these DSP on-chip timers are typically used for software purposes as real-time clock.

CAUTION

In case sampling frequency selector is configured to source sampling frequency input from either TM/XIO-0 or TM/XIO-1 outputs of host PIOX-16 interface, then sampling frequency event detected at the falling edge at TM/XIO-0/1 input signals.

Minimum duration of active low state of signals at TM/XIO-0 or TM/XIO-1 outputs of host PIOX-16 interface least 200 nS in order to provide proper detection of sampling frequency event. This corresponds to the 'clock output mode' of host TMS320C3x/C6xxx DSP on-chip timers #0/#1.

Current sampling frequency is always available at the sampling frequency output (*Fs_OUT*) of on-board JP3 connector, and it can be used either to synchronize external circuits or to connect to the XFs input of other *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

CAUTION

Sampling frequency output (*Fs_OUT*) of on-board JP3 connector provides active low output signal with the sampling frequency event corresponding to the falling edge at *Fs_OUT* output.

PFG programmable sampling frequency generator

On-board PFG programmable frequency generator must be used for accurate setting of sampling frequency for ADC and DAC data acquisition controllers within the 16 Hz .. 1 MHz frequency range with high-resolution setting.

On-board PFG is configured by host DSP software via 9-bit V-field (V0..V8), 7-bit R-field (R0..R6), 2-bit S-field (S0..S2) and 2-bit X-field (X0..X1), which are available via *PFG_CNTR1_RG*, *PFG_CNTR2_RG* and *PFG_CNTR3_RG* registers (table 2-13) of host PIOX-16 interface as the following:

$$Fout = \frac{50*(V+8)}{(R+2)*64*Sv*Xv} \text{ (MHz)}$$

where:

Fout	output frequency value (MHz) of the corresponding PFG
V	numeric value of 9-bit V-field
R	numeric value of 7-bit R-field
Sv	value of S-divider, which is defined by 3-bit S-field in accordance with table 2-19a
Xv	value of X-divider, which is defined by 2-bit X-field in accordance with table 2-19b.

Table 2-19a. S-divider settings for PFG.

value of the S-field			**value of S-divider (*Sv*)**
S2	***S1***	***S0***	
0	*0*	*0*	:10
0	*0*	*1*	:2
0	*1*	*0*	:8
0	*1*	*1*	:4
1	*0*	*0*	:5
1	*0*	*1*	:7
1	*1*	*0*	:9
1	*1*	*1*	:6

Table 2-19b. X-divider settings for PFG.

value of the X-field	**value of the X-field**	**value of X-divider (*Xv*)**
X1	***X0***	
0	*0*	:1
0	*1*	:32
1	*0*	:1024
1	*1*	reserved, do not use

CAUTION

Host PIOX-16 interface reset condition will preset *PFG_CNTR1_RG*, *PFG_CNTR2_RG* and *PFG_CNTR3_RG* registers to default values, which correspond to 100 kHz PFG output frequency.

The following restrictions are applicable for the V- and R- fields when programming on-board PFG sampling frequency generator:

$$4 \le V \le 511$$

$$1 \le R \le 123$$

$$1 \le \frac{5*(V+8)}{(R+2)} \le 32$$

CAUTION

The time interval between succeeding settings of V- and R-fields for on-board PFG sampling frequency generator must be 10 mS or greater in order to meet lock time specification for PFG on-chip PLL.

ADC data acquisition controller

ADC data acquisition process is performed under the control of ADC data acquisition controller, which is the part of on-board DAQCU.

Figure 2-5 presents block-diagram for on-board ADC data acquisition controller, ADC FIFO and logic for ADC IRT controller, which function together with ADC FIFO and ADC data acquisition controller, and is used to transfer real-time data streams between either ADC directly or ADC FIFO and host PIOX-16 interface. For details about ADC IRT controller refer to section “ADC IRT and DAC IRT Controllers” later in this chapter.

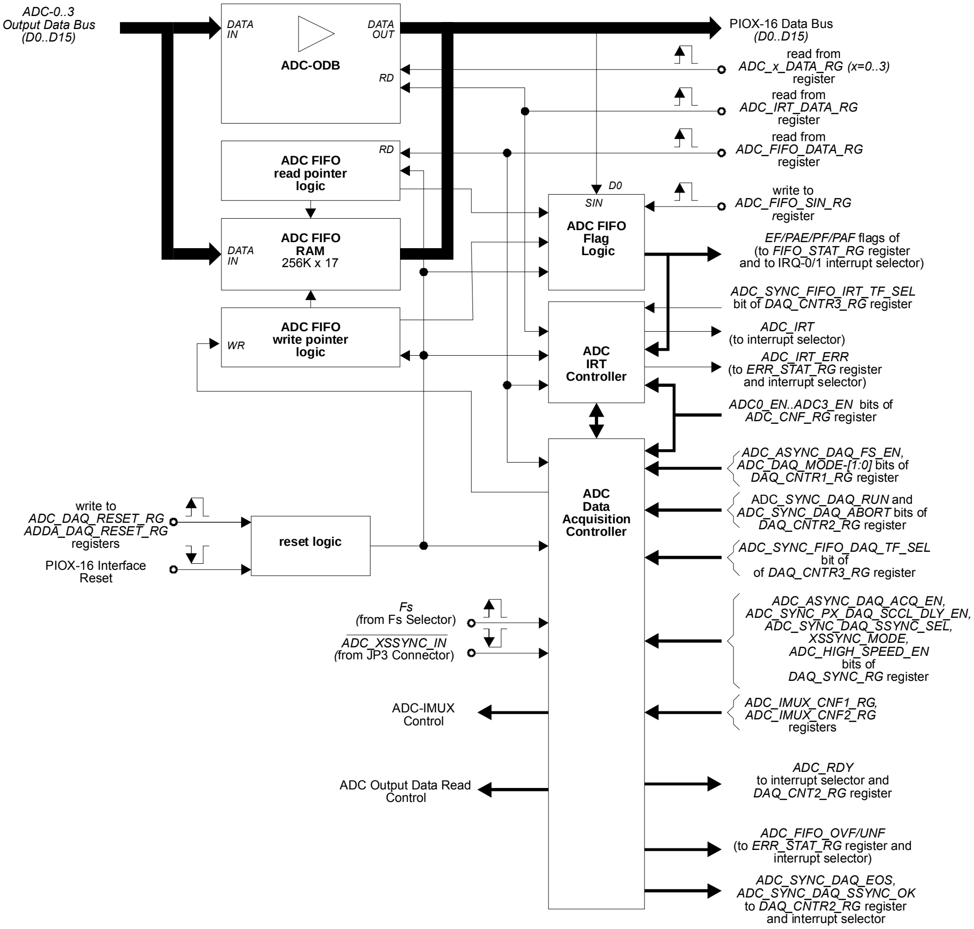

Fig.2-5. ADC data acquisition controller, ADC FIFO and ADC IRT controller.

ADC data acquisition controller provides accurate timing for ADC data acquisition process, detects external ADC data acquisition synchronization (ADC synchronous data acquisition modes only), starts A/D conversion, updates analog input multiplexers (ADC synchronous data acquisition modes only), provides ADC output data transfer into ADC FIFO (ADC synchronous FIFO data acquisition modes only), and detects real-time data acquisition error conditions (ADC synchronous FIFO data acquisition modes only).

ADC data acquisition controller is configured and controlled by host DSP software via a set of control registers, which are the part of host PIOX-16 interface (refer to section 2.2 earlier in this chapter). The following is the list of PIOX-16 interface control registers and register bits, which are used to configure and perform real-time control of ADC data acquisition controller:

- {*ADC_DAQ_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register, which are used to select ADC data acquisition mode (refer to table 2-2)
- *ADC_SYNC_DAQ_RUN* and *ADC_SYNC_DAQ_ABORT* bits of *DAQ_CNTR2_RG* register, which are used to start and abort ADC synchronous data acquisition process (refer to table 2-3)

- *ADC_SYNC_DAQ_SSYNC_OK*, *ADC_SYNC_DAQ_END* and *ADC_SYNC_DAQ_EOS* read-only bits of *DAQ_CNTR2_RG* register, which are used to monitor status of ADC synchronous data acquisition process (refer to table 2-3)
- *ADC_SYNC_FIFO_DAQ_TF* bit of *DAQ_CNTR3_RG* register, which is used to select ADC FIFO flag for normal termination of ADC synchronous FIFO one-pass data acquisition process (refer to table 2-4)
- *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register, which define what particular ADC channels (ADC-0..3) are involved into ADC data acquisition process for all ADC data acquisition modes (table 2-6)
- {*SCCLx_IMUX_CH_SEL-[1:0]*}, *SCCLx_IMUX_DIFF* and *SCCLx_EOS* bits (*x*=0..3) of *ADC_IMUX_CNF1_RG* and *ADC_IMUX_CNF2_RG* registers, which are used to configure analog input multiplexers for each scan cycle of ADC synchronous data acquisition process, and to define last scan cycle for ADC sampling packets for all ADC synchronous data acquisition modes (table 2-8b)
- *ADC_ASYNC_DAQ_FS_EN* bit of *DAQ_CNTR1_RG* register, which is used to enable sampling frequency event in order to start ADC data sampling cycle for ADC asynchronous data acquisition mode (table 2-2)
- *ADC_ASYNC_DAQ_ACQ_EN* bit of *DAQ_SYNC_RG* register (table 2-5), which is used to select between *standard ADC sampling cycle* and *extended ADC sampling cycles* for ADC asynchronous data acquisition mode, i.e. which is used to delay of A/D conversion start by 0.8 µS from the beginning of ADC data sampling cycle in order to properly acquire multiplexed analog input signals during ADC asynchronous data acquisition mode
- *ADC_SYNC_PX_DAQ_SCCL_DLY_EN* bit of *DAQ_SYNC_RG* register, which is used to prolong scan cycles of ADC sampling packet by 1 µS for ADC synchronous data acquisition mode with direct ADC output access via host PIOX-16 interface (*ADC-SYNC-PX-DAQ*) in order to allow extra time for host DSP software to download ADC output data prior next ADC output data will come valid (table 2-5)
- *ADC_RDY* read-only bit of *DAQ_CNTR1_RG* register, which is used to indicate that ADC output data is ready (table 2-3) (is typically used for ADC asynchronous data acquisition mode only)
- *ADC_SYNC_DAQ_SSYNC_SEL* bit of *DAQ_SYNC_RG* register, which is used to selects start synchronization source for ADC synchronous data acquisition process for all ADC synchronous data acquisition modes (table 2-5)
- *XSSYNC_MODE* bit of *DAQ_SYNC_RG* register, which is used to select external start synchronization condition at *ADC_XSSYNC_IN* input pin of on-board JP3 connector (refer to Appendix A) in case external hardware defined start synchronization is selected as start synchronization source for ADC synchronous data acquisition process (table 2-5)
- *ADC_HIGH_SPEED_EN* bit of *DAQ_SYNC_RG* register, which is used to select between ADC normal and ADC high-speed operation modes (table 2-5)
- *ADC_DAQ_RESET_RG* and *ADDA_DAQ_RESET_RG* write-only registers, which are used to reset ADC data acquisition controller
- *ADC_START_RG* and *ADC_START_DAC_OUTPUT_LD_RG* write-only registers, which are used to start ADC sampling cycle from host DSP software for ADC asynchronous data acquisition mode
- *ADC_FIFO_FF*, *ADC_FIFO_PAF*, *ADC_FIFO_EF* and *ADC_FIFO_PAE* bits of *FIFO_STAT_RG* register, which are used to get current status of ADC FIFO FF/PAF/EF/PAE flags for ADC synchronous FIFO one-pass/pass-thru data acquisition modes (refer to table 2-9)
- *ADC_FIFO_OVF* and *ADC_FIFO_UNF* read-only bits of *ERR_STAT_RG* read-only register, and *CLR_ADC_FIFO_OVF* and *CLR_ADC_FIFO_UNF* write-only bits of *ERR_CLR_RG* write-only

register, which are used to get current status of ADC FIFO overflow/underflow errors and to clear these errors for ADC synchronous FIFO one-pass/pass-thru data acquisition modes (refer to table 2-10).

ADC data acquisition controller can operate in one of four ADC data acquisition modes, which is selected via {*ADC_DAQ_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register (table 2-2):

- *ADC ASYNCHRONOUS data acquisition mode* (*ADC-ASYNC-DAQ*), which is used in case ADC sampling cycles shall be initialized on either software defined or sampling frequency events without automatically generated scan cycles. This mode feature simplified timing and is typically used in case ADC sampling events are generated by host DSP software. In this mode, ADC output data is directly available via host PIOX-16 interface via *ADCx_DATA_RG* registers (*x*=0..3), and A/D conversion can be either initialized by DSP software on write to either of *ADC_START_RG* and *ADC_START_DAC_OUTPUT_LD_RG* registers, or can be started on sampling frequency event as defined via bit *ADC_ASYNC_DAQ_FS_EN* bit of *DAQ_CNTR1_RG* register. In this mode, *ADC_IRT_DATA_RG* read-only register must be used by host DSP on-chip DMA controller in order to download real-time ADC output data for all enabled ADC channels using ADC IRT controller (refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details). This mode is also known as *ADC FIFO configuration mode* and is used to set offsets for ADC FIFO partially empty (PAE) and partially full (PAF) flags, which are used by ADC data acquisition controller during *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode and by ADC IRT controller during both *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes. For more details about *ADC-ASYNC-DAQ* data acquisition mode refer to the corresponding subsection below.
- *ADC SYNCHRONOUS WITH DIRECT ADC DATA READ VIA PIOX-16 INTERFACE data acquisition mode* (*ADC-SYNC-PX-DAQ*), which allows to continuously perform A/D conversion of up to 16 analog inputs with accurate sampling timing, while offering direct ADC real-time output data access via host PIOX-16 interface. As for all ADC synchronous data acquisition modes, this mode features accurate sampling timing, ADC sampling packet initialization on sampling frequency event, A/D conversion of up to 16 analog inputs via automatically generated scan cycles inside one ADC sampling packet, and provides support for external hardware defined start synchronization for ADC synchronous data acquisition process. Once started, ADC synchronous data acquisition process in this mode continuously performs A/D conversion until terminated by host DSP software via *ADC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register (table 2-3). Each ADC sampling packet comprises of up to four automatically generated scan cycles as defined by bits *SCCLx_EOS* (*x*=0..2) of *ADC_IMUX_CNF1_RG* and *ADC_IMUX_CNF2_RG* registers. For each scan cycle inside ADC sampling packet, analog input multiplexer is updated in accordance with {*SCCLx_IMUX_CH_SEL*-[1:0]} and *SCCLx_IMUX_DIFF* (*x*=0..3) bits of *ADC_IMUX_CNF1_RG* and *ADC_IMUX_CNF2_RG* registers. In this mode, real-time ADC output data is directly available via host PIOX-16 interface via *ADCx_DATA_RG* registers (*x*=0..3), whereas *ADC_IRT_DATA_RG* read-only register must be used by host DSP on-chip DMA controller in order to download real-time ADC output data for all enabled ADC channels using ADC IRT controller (refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details). This mode must be used in case continuous A/D conversion of up to 16 analog inputs with accurate sampling timing is required, while direct ADC real-time output data read via host PIOX-16 interface is a preferred selection. For more details about *ADC-SYNC-PX-DAQ* data acquisition mode refer to the corresponding subsection below.
- *ADC SYNCHRONOUS FIFO ONE-PASS data acquisition mode* (*ADC-SYNC-FIFO-OPM-DAQ*), which allows to acquire pre-defined number of ADC sampling packets, store real-time ADC data output data in ADC FIFO, and then automatically terminate ADC data acquisition process. As for all ADC synchronous data acquisition modes, this mode features accurate sampling timing, ADC sampling packet initialization on sampling frequency event, A/D conversion of up to 16 analog inputs

via automatically generated scan cycles inside one ADC sampling packet, and provides support for external hardware defined start synchronization for ADC synchronous data acquisition process. Once started, ADC synchronous data acquisition process in this mode performs A/D conversion until terminated normally on ADC FIFO either FF or PAF flag condition as defined by *ADC_SYNC_FIFO_DAQ_TF* bit of *DAQ_CNTR3_RG* register (table 2-4). ADC synchronous data acquisition process can be also terminated by host DSP software via *ADC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register (table 2-3). Each ADC sampling packet comprises of up to four automatically generated scan cycles as defined by bits *SCCLx_EOS* (*x*=0..2) of *ADC_IMUX_CNF1_RG* and *ADC_IMUX_CNF2_RG* registers. For each scan cycle inside ADC sampling packet, analog input multiplexer is updated in accordance with {*SCCLx_IMUX_CH_SEL*-[1:0]} and *SCCLx_IMUX_DIFF* (*x*=0..3) bits of *ADC_IMUX_CNF1_RG* and *ADC_IMUX_CNF2_RG* registers. ADC output data for each scan cycle is automatically pushed into ADC FIFO for all enabled ADC channels as defined by *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6). In this mode, real-time ADC output data read via host PIOX-16 interface via *ADCx_DATA_RG* registers (*x*=0..3) is not supported. Stored ADC real-time output data for enabled ADC channels can be read from *ADC_FIFO_DATA_RG* and *ADC_FIFO_DATA_MSW_RG* registers after ADC synchronous data acquisition process terminates. Also, *ADC_FIFO_DATA_RG* register must be used by host DSP on-chip DMA controller in order to download real-time ADC output data for all enabled ADC channels using ADC IRT controller (refer to section “ADC IRT and DAC IRT Controllers” later in this chapter for more details). This mode must be used in case user application requires to acquire pre-defined number of ADC sampling packets, store ADC output data in ADC FIFO and then automatically terminate ADC data acquisition process. For more details about *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode refer to the corresponding subsection below.

- *ADC SYNCHRONOUS FIFO PASS-THRU data acquisition mode* (*ADC-SYNC-FIFO-PTM-DAQ*), which allows to continuously perform A/D conversion of up to 16 analog inputs with accurate sampling timing and push real-time ADC data output data into ADC FIFO. As for all ADC synchronous data acquisition modes, this mode features accurate sampling timing, ADC sampling packet initialization on sampling frequency event, A/D conversion of up to 16 analog inputs via automatically generated scan cycles inside one ADC sampling packet, and provides support for external hardware defined start synchronization for ADC synchronous data acquisition process. Once started, ADC synchronous data acquisition process in this mode continuously performs A/D conversion until terminated by host DSP software via *ADC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register (table 2-3). Each ADC sampling packet comprises of up to four automatically generated scan cycles as defined by bits *SCCLx_EOS* (*x*=0..2) of *ADC_IMUX_CNF1_RG* and *ADC_IMUX_CNF2_RG* registers. For each scan cycle inside ADC sampling packet, analog input multiplexer is updated in accordance with {*SCCLx_IMUX_CH_SEL*-[1:0]} and *SCCLx_IMUX_DIFF* (*x*=0..3) bits of *ADC_IMUX_CNF1_RG* and *ADC_IMUX_CNF2_RG* registers. ADC output data for each scan cycle is automatically pushed into ADC FIFO for all enabled ADC channels as defined by *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6). In this mode, real-time ADC output data read via host PIOX-16 interface via *ADCx_DATA_RG* registers (*x*=0..3) is not supported. Temporary stored ADC real-time output data for enabled ADC channels can be read from *ADC_FIFO_DATA_RG* and *ADC_FIFO_DATA_MSW_RG* registers while ADC synchronous data acquisition process is running. ADC FIFO is continuously monitored for overflow and underflow error conditions, and the corresponding error flags are available via bits *ADC_FIFO_OVF* and *ADC_FIFO_UNF* bits of *ERR_STAT_RG* register (table 2-10). Also, *ADC_FIFO_DATA_RG* register must be used by host DSP on-chip DMA controller in order to download real-time ADC output data for all enabled ADC channels using ADC IRT controller (refer to section “ADC IRT and DAC IRT Controllers” later in this chapter for more details). This mode must be used in case user application requires to continuously process ADC output data for up to 16 analog inputs with temporary ADC output data buffering in ADC FIFO. For more details about *ADC-SYNC-FIFO-PTM-DAQ* data acquisition mode refer to the corresponding subsection below.

Generally, all above listed ADC data acquisition modes can be grouped as ADC asynchronous data acquisition mode (*ADC-ASYNC-DAQ*) and all ADC synchronous data acquisition modes (*ADC-SYNC-PX-DAQ*, *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ*). ADC asynchronous and synchronous data acquisition modes principally differ in ADC sampling timing, and a set of software defined parameters, which are used to configure ADC data acquisition process. However, the main difference between these ADC data acquisition modes is that all ADC synchronous modes provide high ADC sampling timing accuracy and allow to perform A/D conversion of up to 16 analog inputs using automatically generated scan cycles inside ADC sampling packet, whereas ADC asynchronous mode features simple timing and software initialized ADC sampling cycles although providing A/D conversion of only four analog inputs per one ADC sampling cycle.

Also, ADC data acquisition modes can be grouped as either ADC data acquisition modes with direct access to ADC real-time output data from host PIOX-16 interface (*ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ*) and ADC synchronous FIFO data acquisition modes (*ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ*) with temporary store of ADC real-time output data in ADC FIFO. These ADC real-time output data I/O options allow to use *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM in different applications, which either do not require high sampling frequencies and/or complex signal processing algorithms, and therefore allow direct read of ADC real-time output data into host DSP environment, as well as in applications, which require high sampling frequencies and/or complex signal processing algorithm, and therefore cannot perform real-time ADC output data download at full speed in each sampling frequency period and need to temporary store ADC output data in high-density FIFO for further download into host DSP environment.

On-board ADC FIFO, which is used to temporary store ADC real-time output data in *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes, is 256Kx18 static RAM array with the corresponding read/write pointer logic and programmable FIFO flag logic. ADC FIFO flag logic generates four output flags, which are used to indicate current FIFO status and to control operation of ADC data acquisition controller and ADC IRT controller. For more details about FIFO flag logic refer to the corresponding subsection below.

For more details and timing diagrams for each ADC data acquisition mode refer to the corresponding subsections below.

DAC data acquisition controller

DAC data acquisition process is performed under the control of DAC data acquisition controller, which is the part of on-board DAQCU.

Figure 2-6 presents block-diagram for on-board DAC data acquisition controller, DAC FIFO and logic for DAC IRT controller, which function together with DAC FIFO and DAC data acquisition controller, and is used to transfer real-time data streams between either DAC directly or DAC FIFO and host PIOX-16 interface. For details about DAC IRT controller refer to section “ADC IRT and DAC IRT Controllers” later in this chapter.

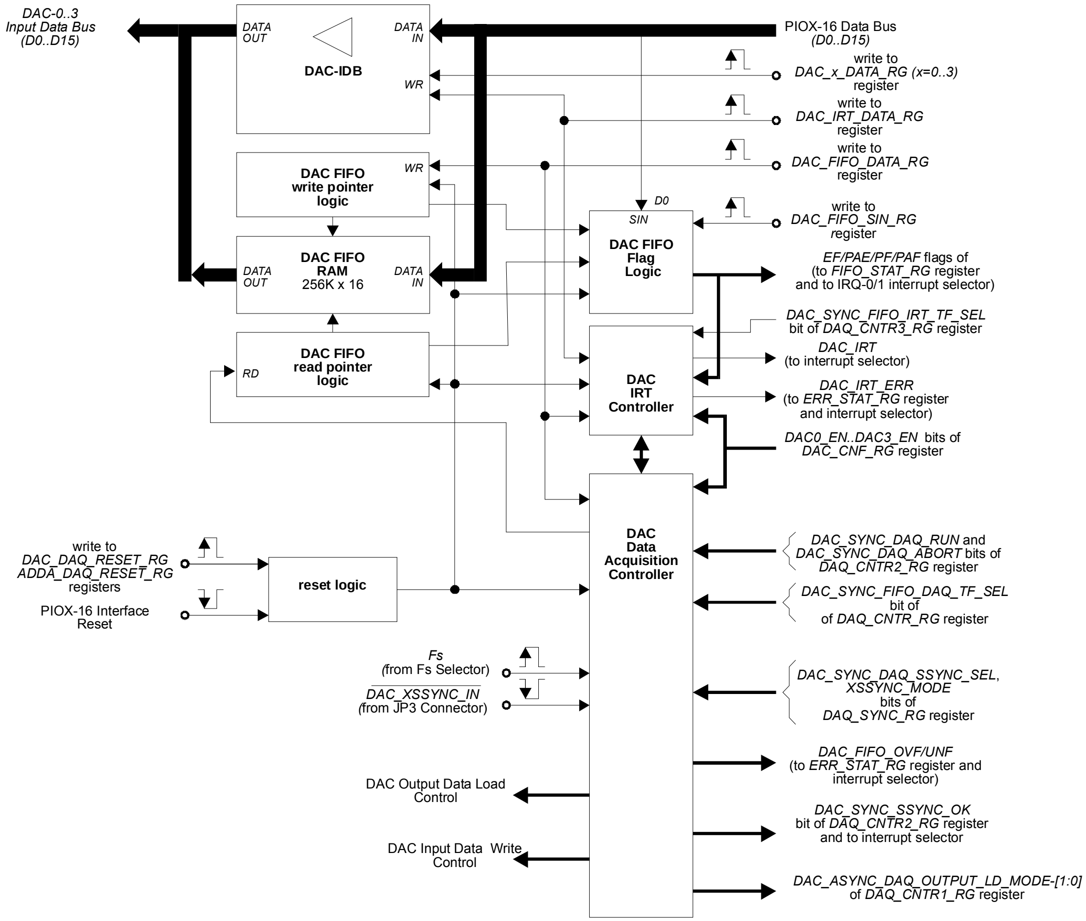
DAC-0..3
Input Data Bus
(D0..D15)
DATA OUT
DATA IN
DAC-IDB
WR
PIOX-16 Data Bus
(D0..D15)
write to DAC_x_DATA_RG (x=0..3) register
write to DAC_IRT_DATA_RG register
write to DAC_FIFO_DATA_RG register
DAC FIFO write pointer logic
WR
D0
SIN
write to DAC_FIFO_SIN_RG register
DAC FIFO Flag Logic
DAC FIFO RAM 256K x 16
DATA OUT
DATA IN
EF/PAE/PF/PAF flags of (to FIFO_STAT_RG register and to IRQ-0/1 interrupt selector)
DAC_SYNC_FIFO_IRT_TF_SEL bit of DAQ_CNTR3_RG register
DAC_IRT (to interrupt selector)
DAC FIFO read pointer logic
RD
DAC IRT Controller
DAC_IRT_ERR (to ERR_STAT_RG register and interrupt selector)
DAC0_EN..DAC3_EN bits of DAC_CNF_RG register
DAC Data Acquisition Controller
DAC_SYNC_DAQ_RUN and DAC_SYNC_DAQ_ABORT bits of DAQ_CNTR2_RG register
write to DAC_DAQ_RESET_RG ADDA_DAQ_RESET_RG registers
reset logic
PIOX-16 Interface Reset
DAC_SYNC_FIFO_DAQ_TF_SEL bit of of DAQ_CNTR_RG register
Fs (from Fs Selector)
DAC_XSSYNC_IN (from JP3 Connector)
DAC_SYNC_DAQ_SSYNC_SEL, XSSYNC_MODE bits of DAQ_SYNC_RG register
DAC_FIFO_OVF/UNF (to ERR_STAT_RG register and interrupt selector)
DAC Output Data Load Control
DAC_SYNC_SSYNC_OK bit of DAQ_CNTR2_RG register and to interrupt selector
DAC Input Data Write Control
DAC_ASYNC_DAQ_OUTPUT_LD_MODE-[1:0] of DAQ_CNTR1_RG register

Fig.2-6. DAC data acquisition controller, DAC FIFO and DAC IRT controller.

DAC data acquisition controller provides accurate timing for DAC data acquisition process, detects external DAC data acquisition synchronization (DAC synchronous data acquisition modes only), performs DAC output data load, provides transfer of DAC FIFO output data to DAC inputs (DAC synchronous FIFO data acquisition modes only), and detects real-time data acquisition error conditions (DAC synchronous FIFO data acquisition modes only).

DAC data acquisition controller is configured and controlled by host DSP software via a set of control registers, which are the part of host PIOX-16 interface (refer to section 2.2 earlier in this chapter). The following is the list of PIOX-16 interface control registers and register bits, which are used to configure and perform real-time control of DAC data acquisition controller:

- {*DAC_DAQ_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register, which are used to select DAC data acquisition mode (refer to table 2-2)
- *DAC_SYNC_DAQ_RUN* and *DAC_SYNC_DAQ_ABORT* bits of *DAQ_CNTR2_RG* register, which are used to start and abort DAC synchronous data acquisition process (refer to table 2-3)

- *DAC_SYNC_DAQ_SSYNC_OK*, *DAC_SYNC_DAQ_END* and *DAC_SYNC_DAQ_EOS* read-only bits of *DAQ_CNTR2_RG* register, which are used to monitor status of DAC synchronous data acquisition process (refer to table 2-3)
- *DAC_SYNC_FIFO_DAQ_TF* bit of *DAQ_CNTR3_RG* register, which is used to select DAC FIFO flag for normal termination of DAC synchronous FIFO one-pass data acquisition process (refer to table 2-4)
- *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register, which define what particular DAC channels (DAC-0..3) are involved into DAC data acquisition process for all DAC data acquisition modes (table 2-7)
- {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register, which are used to select DAC output data load mode for DAC asynchronous data acquisition mode (table 2-2)
- *DAC_SYNC_DAQ_SSYNC_SEL* bit of *DAQ_SYNC_RG* register, which is used to selects start synchronization source for DAC synchronous data acquisition process for all DAC synchronous data acquisition modes (table 2-5)
- *XSSYNC_MODE* bit of *DAQ_SYNC_RG* register, which is used to select external start synchronization condition at *DAC_XSSYNC_IN* input pin of on-board JP3 connector (refer to Appendix A) in case external hardware defined start synchronization is selected as start synchronization source for DAC synchronous data acquisition process (table 2-5)
- *DAC_DAQ_RESET_RG* and *ADDA_DAQ_RESET_RG* write-only registers, which are used to reset DAC data acquisition controller
- *DAC_OUTPUT_LD_RG* and *ADC_START_DAC_OUTPUT_LD_RG* write-only registers, which are used to generate DSP software controlled DAC output load for DAC asynchronous data acquisition mode
- *DAC_FIFO_FF*, *DAC_FIFO_PAF*, *DAC_FIFO_EF* and *DAC_FIFO_PAE* bits of *FIFO_STAT_RG* register, which are used to get current status of DAC FIFO FF/PAF/EF/PAE flags for DAC synchronous FIFO one-pass/pass-thru data acquisition modes (refer to table 2-9)
- *DAC_FIFO_OVF* and *DAC_FIFO_UNF* read-only bits of *ERR_STAT_RG* read-only register, and *CLR_DAC_FIFO_OVF* and *CLR_DAC_FIFO_UNF* write-only bits of *ERR_CLR_RG* write-only register, which are used to get current status of DAC FIFO overflow/underflow errors and to clear these errors for DAC synchronous FIFO one-pass/pass-thru data acquisition modes (refer to table 2-10).

DAC data acquisition controller can operate in one of four DAC data acquisition modes, which is selected via {*DAC_DAQ_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register (table 2-2):

- *DAC ASYNCHRONOUS data acquisition mode* (*DAC-ASYNC-DAQ*), which is used for software and sampling frequency event initialized DAC output data load cycles. This mode feature simplified timing and is typically used in case DAC output data load events are generated by host DSP software. In this mode, DAC input data can be directly written via host PIOX-16 interface via *DACx_DATA_RG* registers (*x*=0..3), and D/A conversion can be either initialized by DSP software on write to either of *DAC_OUTPUT_LD_RG* and *ADC_START_DAC_OUTPUT_LD_RG* registers, or can be started on the sampling frequency event as defined via bits {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE*-[1:0]} of *DAQ_CNTR1_RG* register. In this mode, *DAC_IRT_DATA_RG* write-only register must be used by host DSP on-chip DMA controller in order to upload real-time DAC input data for all enabled DAC channels using DAC IRT controller (refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details). This mode is also known as *DAC FIFO configuration mode* and is used to set offsets for DAC FIFO partially empty (PAE) and partially full (PAF) flags, which are used by DAC data acquisition controller during *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode and by DAC IRT controller during both *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes. For more details about *DAC-ASYNC-DAQ* data acquisition mode refer to the corresponding subsection below.

- *DAC SYNCHRONOUS WITH DIRECT DAC DATA WRITE VIA PIOX-16 INTERFACE data acquisition mode* (*DAC-SYNC-PX-DAQ*), which allows to continuously perform D/A conversion with accurate sampling timing, while offering direct DAC real-time input data write via host PIOX-16 interface. As for all DAC synchronous data acquisition modes, this mode features accurate sampling timing, DAC output data load on sampling frequency event, and provides support for external hardware defined start synchronization for DAC synchronous data acquisition process. Once started, DAC synchronous data acquisition process in this mode continuously performs D/A conversion until terminated by host DSP software via *DAC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register (table 2-3). In this mode, real-time DAC input data can be directly written via host PIOX-16 interface via *DACx_DATA_RG* registers (*x*=0..3), whereas *DAC_IRT_DATA_RG* write-only register must be used by host DSP on-chip DMA controller in order to upload real-time DAC input data for all enabled DAC channels using DAC IRT controller (refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details). This mode must be used in case continuous D/A conversion with accurate sampling timing is required, while direct DAC real-time input data write via host PIOX-16 interface is a preferred selection. For more details about *DAC-SYNC-PX-DAQ* data acquisition mode refer to the corresponding subsection below.
- *DAC SYNCHRONOUS FIFO ONE-PASS data acquisition mode* (*DAC-SYNC-FIFO-OPM-DAQ*), which allows store real-time DAC data input data in DAC FIFO, to perform D/A conversion for pre-defined number of sampling frequency periods, and then automatically terminate DAC data acquisition process. As for all DAC synchronous data acquisition modes, this mode features accurate sampling timing, DAC output data load on sampling frequency event, and provides support for external hardware defined start synchronization for DAC synchronous data acquisition process. Once started, DAC synchronous data acquisition process in this mode performs D/A conversion until terminated normally on DAC FIFO either EF or PAE flag condition as defined by *DAC_SYNC_FIFO_DAQ_TF* bit of *DAQ_CNTR3_RG* register (table 2-4). DAC synchronous data acquisition process can be also terminated by host DSP software via *DAC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register (table 2-3). DAC input data during each sampling frequency period is automatically extracted from DAC FIFO for all enabled DAC channels as defined by *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-7). In this mode, real-time DAC input data write via host PIOX-16 interface via *DACx_DATA_RG* registers (*x*=0..3) is not supported. Stored DAC real-time input data for enabled DAC channels must be written to *DAC_FIFO_DATA_RG* register prior DAC synchronous data acquisition process starts. Also, *DAC_FIFO_DATA_RG* register must be used by host DSP on-chip DMA controller in order to upload real-time DAC input data for all enabled DAC channels using DAC IRT controller (refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details). This mode must be used in case user application requires to extract DAC input data from DAC FIFO, perform D/A conversion for pre-defined number of sampling frequency periods, and then automatically terminate DAC data acquisition process. For more details about *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode refer to the corresponding subsection below.
- *DAC SYNCHRONOUS FIFO PASS-THRU data acquisition mode* (*DAC-SYNC-FIFO-PTM-DAQ*), which allows to continuously perform D/A conversion with accurate sampling timing and extract real-time DAC input data from DAC FIFO. As for all DAC synchronous data acquisition modes, this mode features accurate sampling timing, DAC output data load on sampling frequency event, and provides support for external hardware defined start synchronization for DAC synchronous data acquisition process. Once started, DAC synchronous data acquisition process in this mode continuously performs D/A conversion until terminated by host DSP software via *DAC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register (table 2-3). DAC input data during each sampling frequency period is automatically extracted from DAC FIFO for all enabled DAC channels as defined by *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-7). In this mode, real-time DAC input data write via host PIOX-16 interface via *DACx_DATA_RG* registers (*x*=0..3) is not supported.

Stored DAC real-time input data for enabled DAC channels must be written to *DAC_FIFO_DATA_RG* register prior DAC synchronous data acquisition process starts. Also, *DAC_FIFO_DATA_RG* register must be used by host DSP on-chip DMA controller in order to upload real-time DAC input data for all enabled DAC channels using DAC IRT controller (refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details). This mode must be used in case user application requires to continuously perform D/A conversion with temporary DAC input data buffering in DAC FIFO. For more details about *DAC-SYNC-FIFO-PTM-DAQ* data acquisition mode refer to the corresponding subsection below.

Generally, all above listed DAC data acquisition modes can be grouped as DAC asynchronous data acquisition mode (*DAC-ASYNC-DAQ*) and all DAC synchronous data acquisition modes (*DAC-SYNC-PX-DAQ*, *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ*). DAC asynchronous and synchronous data acquisition modes principally differ in DAC sampling timing accuracy, and a set of software defined parameters, which are used to configure DAC data acquisition process.

Also, DAC data acquisition modes can be grouped as either ADC data acquisition modes with direct access to DAC real-time input data from host PIOX-16 interface (*DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ*) and DAC synchronous FIFO data acquisition modes (*DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ*) with temporary store of DAC real-time output data in DAC FIFO. These DAC real-time input data I/O options allow to use *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM in different applications, which either do not require high sampling frequencies and/or complex signal processing algorithms, and therefore allow direct write of DAC real-time input data from host DSP environment, as well as in applications, which require high sampling frequencies and/or complex signal processing algorithm, and therefore cannot perform real-time DAC input data upload at full speed in each sampling frequency cycle and need to temporary store real-time DAC input data in high-density FIFO with further extract for real-time D/A conversion..

On-board DAC FIFO, which is used to temporary store DAC real-time input data in *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes, is 256Kx18 static RAM array with the corresponding read/write pointer logic and programmable FIFO flag logic. DAC FIFO flag logic generates four output flags, which are used to indicate current FIFO status and to control operation of DAC data acquisition controller and DAC IRT controller. For more details about FIFO flag logic refer to the corresponding subsection below.

For more details and timing diagrams for each DAC data acquisition mode refer to the corresponding subsections below.

FIFO flag logic

ADC/DAC FIFO flag logic is used during corresponding ADC/DAC synchronous FIFO data acquisition modes (*ADC-SYNC-FIFO-OPM-DAQ*, *ADC-SYNC-FIFO-PTM-DAQ*, *DAC-SYNC-FIFO-OPM-DAQ*, and *DAC-SYNC-FIFO-PTM-DAQ*) in order to terminate the corresponding ADC/DAC synchronous FIFO one-pass data acquisition process, monitor FIFO fill-in conditions, and to control corresponding ADC/DAC IRT controller. FIFO flag logic for each of the ADC/DAC FIFO comprises of the following flags:

- FIFO empty flag (EF)
- Software configured FIFO partially empty flag (PAE)
- FIFO full flag (FF)
- Software configured FIFO partially full flag (PAF).

FIFO empty flag (EF) for each of ADC/DAC FIFO appears for software polling as the corresponding *xxxx_FIFO_EF* bit in read-only *FIFO_STAT_RG* registers (table 2-9). FIFO EF flag is non-programmable and indicates FIFO empty condition. ADC FIFO EF flag can be used to terminate transmission cycle for ADC

IRT controller, whereas DAC FIFO EF flags can be used for normal termination of DAC synchronous data acquisition process in *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode and to initialize transmission cycle for DAC IRT controller. All FIFO EF flags can be also used to generate host PIOX-16 interrupt requests IRQ-0 and IRQ-1 (tables 2-15).

FIFO partially empty flag (PAE) for each of ADC/DAC FIFO appears for software polling as *xxxx_FIFO_PAE* bit in read-only *FIFO_STAT_RG* register (table 2-9). FIFO PAE flag can be programmed during corresponding *ADC-ASYNC-DAQ* or *DAC-ASYNC-DAQ* data acquisition mode (also known as corresponding *ADC/DAC FIFO configuration modes*) via *ADC_FIFO_SIN_RG* and *DAC_FIFO_SIN_RG* registers (table 2-12). ADC FIFO PAE flag can be used to terminate transmission cycle for ADC IRT controller, whereas DAC FIFO PAE flags can be used for normal termination of DAC data acquisition process in *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode and to initialize transmission cycle for DAC IRT controller. All FIFO PAE flags can be also used to generate host PIOX-16 interrupt requests IRQ-0 and IRQ-1 (tables 2-15). In case FIFO PAE flag is not set, then there is more than N unread samples inside the corresponding FIFO (N is the programmed offset for FIFO PAE flag). In case FIFO PAE flag is set, then there is N or less number of unread samples inside the corresponding FIFO.

CAUTION

Offset value for FIFO PAE flag is set to default N=1,023 value on the host PIOX-16 reset condition, and can be reprogrammed to any value below 256K=262,144 in the corresponding *ADC-ASYNC-DAQ* or *DAC-ASYNC-DAQ* data acquisition mode (also known as corresponding *ADC/DAC FIFO configuration modes*).

FIFO full flag (FF) for each of the ADC/DAC FIFO appears for software polling as the corresponding *xxxx_FIFO_FF* bit in read-only *FIFO_STAT_RG* register (table 2-9). FIFO FF flag is non-programmable and indicates FIFO full condition. ADC FIFO FF flag can be used for normal termination of ADC synchronous data acquisition process in *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode and to initialize transmission cycle for ADC IRT controller, whereas DAC FIFO FF flags can be used to terminate transmission cycle for DAC IRT controller. All FIFO FF flags can be also used to generate host PIOX-16 interrupt requests IRQ-0 and IRQ-1 (tables 2-15).

FIFO partially full flag (PAF) for each of ADC/DAC FIFO appears for software polling as *xxxx_FIFO_PAF* bit in read-only *FIFO_STAT_RG* register (table 2-9). FIFO PAF flag can be programmed during corresponding *ADC-ASYNC-DAQ* or *DAC-ASYNC-DAQ* data acquisition mode (also known as corresponding *ADC/DAC FIFO configuration modes*) via *ADC_FIFO_SIN_RG* and *DAC_FIFO_SIN_RG* registers (table 2-12). ADC FIFO PAF flag can be used for normal termination of ADC synchronous data acquisition process in *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode and to initialize transmission cycle for ADC IRT controller, whereas DAC FIFO PAF flags can be used to terminate transmission cycle for DAC IRT controller. All FIFO PAF flags can be also used to generate host PIOX-16 interrupt requests IRQ-0 and IRQ-1 (tables 2-15). In case FIFO PAF flag is not set, then there is (2^{18}-M-1) or less number of unread samples inside the corresponding FIFO (M is the programmed offset for FIFO PAF flag). In case FIFO PAF flag is set, then there is (2^{18}-M) or more number of unread samples inside the corresponding FIFO (M is the programmed offset for FIFO PAF flag).

CAUTION

Offset value for FIFO PAF flag is set to default M=1,023 value on the host PIOX-16 reset condition, and can be reprogrammed to any value below 256K=262,144 in the corresponding *ADC-ASYNC-DAQ* or *DAC-ASYNC-DAQ* data acquisition mode (also known as corresponding *ADC/DAC FIFO configuration modes*).

CAUTION

Offset values for FIFO PAE and PAF flags offsets will not change when performing reset of the corresponding data acquisition controllers and FIFO read/write pointer by writing to *ADC_DAQ_RESET_RG* , *DAC_DAQ_RESET_RG* and *ADDA_DAQ_RESET_RG* registers.

Table 2-20 contains description for ADC/DAC FIFO flag logic operation during corresponding ADC/DAC synchronous FIFO data acquisition modes.

Table 2-20. FIFO flag logic procurement.

Number of words in FIFO	**FIFO FF**	**FIFO PAF**	**FIFO PAE**	**FIFO EF**
0	0	0	1	1
1 to N	0	0	1	0
(N+1) to (262,144-(M+1))	0	0	0	0
(262,144-M) to 262,143	0	1	0	0
262,144	1	1	0	0

Notes:
1. '*N* ' denotes offset value for FIFO PAE flag.
2. '*M* ' denotes offset value for FIFO PAF flag.

Programming ADC/DAC FIFO flags offset

ADC/DAC FIFO partially full (PAF) and partially empty (PAE) flags offsets shall be programmed during corresponding *ADC-ASYNC-DAQ* or *DAC-ASYNC-DAQ* data acquisition mode (also known as corresponding *ADC/DAC FIFO configuration modes*) in case default offset values of ADC/DAC FIFO PAE/PAF flags, which are set on the PIOX-16 reset condition, are not sufficient for user application.

CAUTION

ADC-ASYNC-DAQ ADC data acquisition mode (also known as *ADC FIFO configuration mode*) is selected in case {*ADC_DAQ_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register (table 2-2) are set to the {0,0} state. This mode is set as default on PIOX-16 interface reset condition.

DAC-ASYNC-DAQ DAC data acquisition mode (also known as *DAC FIFO configuration mode*) is selected in case {*DAC_DAQ_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register (table 2-2) are set to the {0,0} state. This mode is set as default on PIOX-16 interface reset condition.

PAE and PAF flags for ADC/DAC FIFO can be programmed by host DSP software of host *TORNADO* DSP system/controller by means of serial write to *ADC_FIFO_SIN_RG* and *DAC_FIFO_SIN_RG* registers correspondingly (table 2-12).

CAUTION

Writing to *ADC_FIFO_SIN_RG* register will take effect only in case ADC data acquisition controller has been configured in *ADC-ASYNC-DAQ* ADC data acquisition mode (also known as *ADC FIFO configuration mode*). PAE/PAF flags for ADC FIFO will be programmed simultaneously when writing to the *ADC_FIFO_SIN_RG* register.

Writing to *DAC_FIFO_SIN_RG* register will take effect only in case DAC data acquisition controller has been configured in *DAC-ASYNC-DAQ* DAC data acquisition mode (also known as *DAC FIFO configuration mode*). PAE/PAF flags for DAC FIFO will be programmed simultaneously when writing to the *DAC_FIFO_SIN_RG* register.

Figures 2-7a and 2-7b provides timing diagrams for programming PAE/PAF flags for ADC and DAC FIFO during *ADC-ASYNC-DAQ* ADC data acquisition mode and *DAC-ASYNC-DAQ* DAC data acquisition mode correspondingly.

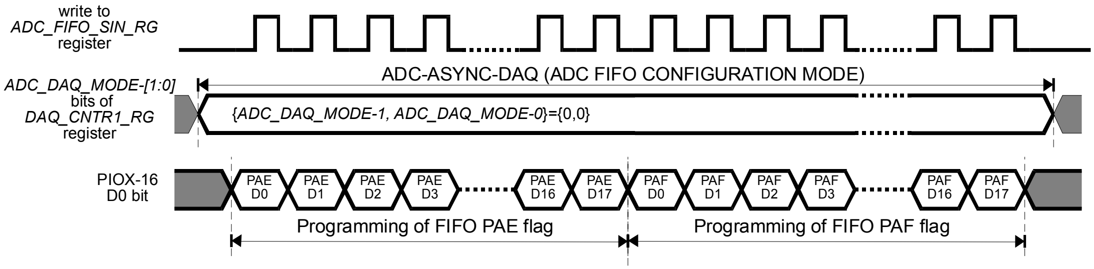

Fig.2-7a. Timing diagram for programming of PAE/PAF flags for ADC FIFO.

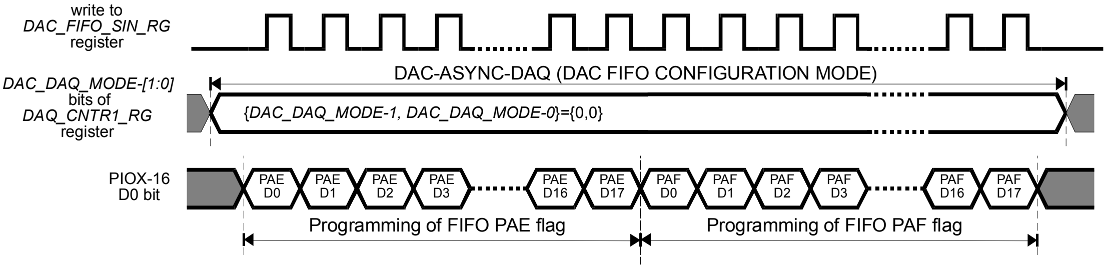

Fig.2-7b. Timing diagram for programming of PAE/PAF flags for DAC FIFO.

After either *ADC-ASYNC-DAQ* ADC data acquisition mode and/or *DAC-ASYNC-DAQ* DAC data acquisition mode has been set by host DSP software, then host DSP software must perform a series of 36 writes to the corresponding either *ADC_FIFO_SIN_RG* and/or *DAC_FIFO_SIN_RG* register in order to program new 18-bit offset values for PAE and PAF flags starting from the least significant bit of offset value for PAE flag and ending with most significant bit of offset value for PAF flag.

CAUTION

When writing to either *ADC_FIFO_SIN_RG* or *DAC_FIFO_SIN_RG* registers, only D0 data bit of PIOX-16 data bus is valid and is used for serial programming of offset values for PAE and PAF flags of the corresponding FIFO.

CAUTION

New offset values for PAE and PAF flags of ADC FIFO will be set only after the 36-th write to *ADC_FIFO_SIN_RG* register.

New offset values for PAE and PAF flags of DAC FIFO will be set only after the 36-th write to *DAC_FIFO_SIN_RG* register.

It is not possible for DSP software to read back programmed offset values for PAE and PAF flags of ADC FIFO and DAC FIFO.

It is recommended to reset of FIFO logic of ADC FIFO and DAC FIFO prior programming PAE and PAF flags for the corresponding FIFO. This will ensure that internal logic of ADC FIFO and DAC FIFO will correctly interpret external serial programming procedure for PAE/PAF flags as starting from the least significant bit for PAE flag in order to match serial input data, which will be loaded by host DSP software via *ADC_FIFO_SIN_RG* and *DAC_FIFO_SIN_RG* registers.

CAUTION

Writing to *ADC_DAQ_RESET_RG* register will reset internal logic of ADC FIFO only along with reset of ADC data acquisition controller. Written data is ignored.

Writing to *DAC_DAQ_RESET_RG* register will reset internal logic of DAC FIFO only along with reset of ADC data acquisition controller. Written data is ignored.

Writing to *ADDA_DAQ_RESET_RG* register will reset logic for both ADC and DAC FIFO along with reset of both ADC and DAC data acquisition controllers. Written data is ignored.

ADC data acquisition timing for ADC asynchronous data acquisition mode (ADC-ASYNC-DAQ)

ADC asynchronous data acquisition mode (*ADC-ASYNC-DAQ*) features most simple timing diagram and is a recommended selection in case A/D conversion cycles (ADC sampling cycles) shall be initialized on either software defined or sampling frequency events, and/or in case automatic A/D conversion of more than 4 analog inputs in one sampling frequency period is not required.

ADC asynchronous data acquisition mode is the most simple to configure and to use ADC data acquisition mode, which delivers flexibility in A/D conversion control via host DSP software.

CAUTION

ADC-ASYNC-DAQ data acquisition mode is selected in case {*ADC_DAQ_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register are set to the {0,0} state (table 2-2).

Operation of ADC data acquisition controller in *ADC asynchronous* data acquisition mode comprises of a series of *ADC sampling cycles*, which may appear as either *standard ADC sampling cycles*, or *extended ADC sampling cycles*.

CAUTION

Along in this manual, if other is not specified, common *ADC sampling cycle* convention is used to denote either of *standard ADC sampling cycle* and *extended ADC sampling cycle* of ADC data acquisition controller in ADC asynchronous data acquisition mode (*ADC-ASYNC-DAQ*).

Each *ADC sampling cycle* performs synchronous sampling and A/D conversion of analog outputs of analog input multiplexers for all ADC-0..3 channels.

CAUTION

ADC0_EN..ADC3_EN bits of *ADC_CNF_RG* register (table 2-6) are ignored by ADC data acquisition controller in *ADC asynchronous* data acquisition mode, and are used by ADC IRT controller only in order to automatically download ADC output data into host DSP environment (refer to section "ADC IRT and DAC IRT Controllers" later in this chapter for more details).

Standard ADC sampling cycle includes A/D conversion procedure only, and is selected in case *ADC_ASYNC_DAQ_ACQ_EN* bit of *DAQ_SYNC_RG* register (table 2-5) is set to the '1' state. *Standard ADC sampling cycle* is terminated on ADC output data ready condition, which is indicated via bit *ADC_RDY* of *DAQ_CNTR2_RG* register (table 2-3). Although *standard ADC sampling cycle* can provide maximum of ADC sampling frequency, care must be taken by host DSP software in order to ensure that next *standard ADC sampling cycle* will start at least 0.25 µS after last ADC data ready event and at least 0.8 µS time after last update of analog input multiplexer, whichever comes latest.

Extended ADC sampling cycle includes A/D conversion procedure preceded by extra 0.8 µS time delay that is required to properly acquire signal at analog inputs prior A/D conversion starts, and is selected in case *ADC_ASYNC_DAQ_ACQ_EN* bit of *DAQ_SYNC_RG* register (table 2-5) is set to the '1' state. *Extended ADC sampling cycle* is terminated on ADC output data ready condition, which is indicated via bit *ADC_RDY* of *DAQ_CNTR2_RG* register (table 2-3). In case *Extended ADC sampling cycles* are selected, then next *extended ADC sampling cycle* can be initialized immediately after the end of the previous *extended ADC sampling cycle*.

A/D conversion time, i.e. time position of ADC output data ready condition, and maximum ADC sampling frequency are defined by the setting of *ADC_HIGH_SPEED_EN* bit of *DAQ_SYNC_RG* register (table 2-5), which is used to select between *ADC normal-speed* and *ADC high-speed* modes.

In case *ADC high-speed mode* is selected (*ADC_HIGH_SPEED_EN* bit of *DAQ_SYNC_RG* register is in the '1' state), then A/D conversion time is 0.75 µS. Maximum ADC sampling frequency in this case is 1 MHz (1 µS of summary 'sampling+conversion' time) in case *ADC_ASYNC_DAQ_ACQ_EN* bit of *DAQ_SYNC_RG* register is in the '0' state, and 0.645 MHz (1.55 µS of summary 'sampling+conversion' time) in case *ADC_ASYNC_DAQ_ACQ_EN* bit of *DAQ_SYNC_RG* register is in the '1' state.

CAUTION

For *ADC high-speed mode*, maximum time between sequential A/D conversion starts is 1 mS, and in case it exceeds, then ADC output data for the first ADC sampling cycle must be ignored and A/D conversion must be restarted.

In case *ADC normal-speed mode* is selected (*ADC_HIGH_SPEED_EN* bit of *DAQ_SYNC_RG* register is in the '0' state), then A/D conversion time is 1 µS. In this case, maximum ADC sampling frequency is 0.8 MHz (1.25 µS of summary 'sampling+conversion' time) in case *ADC_ASYNC_DAQ_ACQ_EN* bit of *DAQ_SYNC_RG* register is in the '0' state, and 0.555 MHz (1.8 µS of summary 'sampling+conversion' time) in case *ADC_ASYNC_DAQ_ACQ_EN* bit of *DAQ_SYNC_RG* register is in the '1' state.

CAUTION

There are no minimum sampling frequency restrictions for *ADC normal mode*, and it can be as low as 0 Hz.

ADC output data for *ADC normal mode* is valid for all ADC sampling cycles starting from the first ADC sampling cycle.

ADC sampling cycles in ADC asynchronous data acquisition mode can be initialized by host DSP software, and, if enabled, on hardware defined sampling frequency event.

CAUTION

In case there is active *ADC sampling cycle* in progress, then all software and hardware defined initialization events for *ADC sampling cycle* will be ignored until termination of currently active *ADC sampling cycle*.

Software initialized *ADC sampling cycles* in ADC asynchronous data acquisition mode are generated anytime when host DSP writes to either *ADC_START_RG* or *ADC_START_DAC_OUTPUT_LD_RG* register.

CAUTION

Software initialization of *ADC sampling cycle* is performed independently from the state of *ADC_ASYNC_DAQ_FS_EN* bit of *DAQ_CNTR1_RG* register (table 2-5).

Software initialized *ADC sampling cycles* may feature large time jitter value for ADC analog input sampling events, which time positions are defined exclusively by host DSP software.

Figure 2-8 shows ADC sampling timing diagram for *standard* and *extended ADC sampling cycles*, which are initialized by host DSP software.

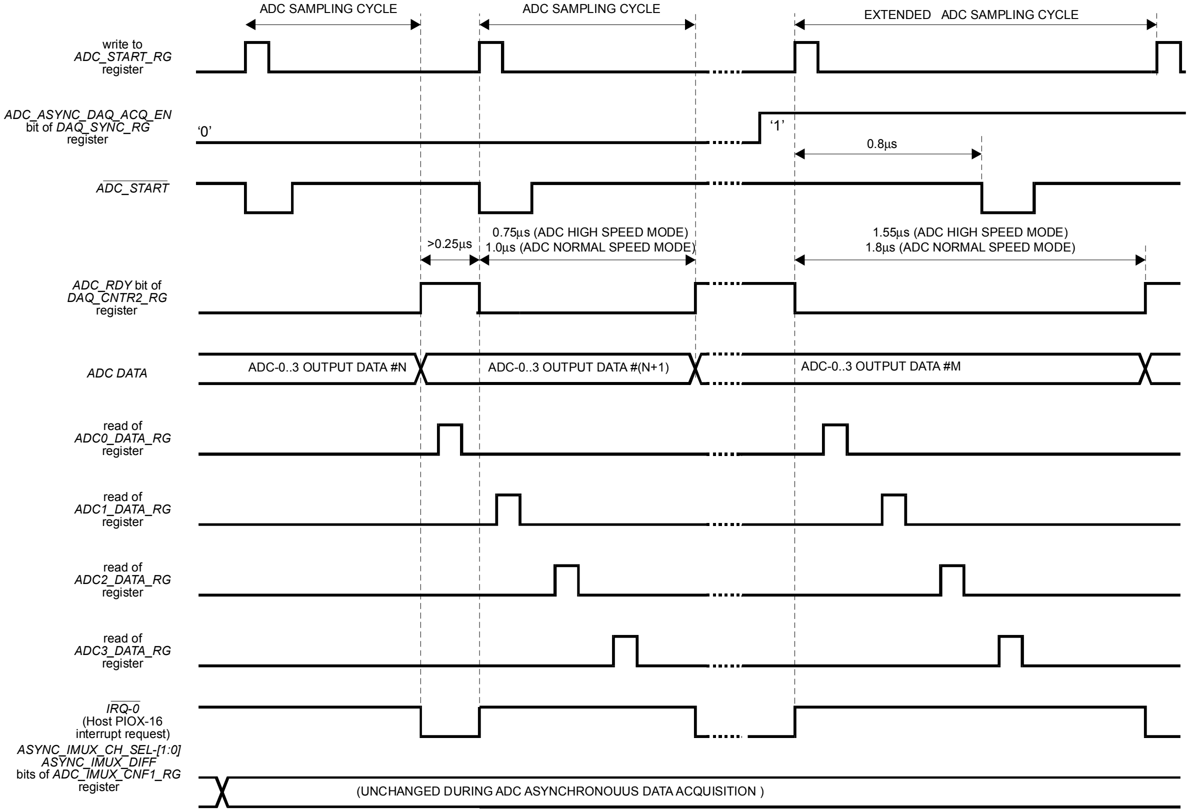
ADC SAMPLING CYCLE
ADC SAMPLING CYCLE
EXTENDED ADC SAMPLING CYCLE
write to ADC_START_RG register
ADC_ASYNC_DAQ_ACQ_EN bit of DAQ_SYNC_RG register
'0'
'1'
0.8µs
ADC_START
>0.25µs
0.75µs (ADC HIGH SPEED MODE)
1.0µs (ADC NORMAL SPEED MODE)
1.55µs (ADC HIGH SPEED MODE)
1.8µs (ADC NORMAL SPEED MODE)
ADC_RDY bit of DAQ_CNTR2_RG register
ADC DATA
ADC-0..3 OUTPUT DATA #N
ADC-0..3 OUTPUT DATA #(N+1)
ADC-0..3 OUTPUT DATA #M
read of ADC0_DATA_RG register
read of ADC1_DATA_RG register
read of ADC2_DATA_RG register
read of ADC3_DATA_RG register
IRQ-0 (Host PIOX-16 interrupt request)
ASYNC_IMUX_CH_SEL-[1:0] ASYNC_IMUX_DIFF bits of ADC_IMUX_CNF1_RG register
(UNCHANGED DURING ADC ASYNCHRONOUUS DATA ACQUISITION)

Fig.2-8. Timing diagram for software initialized ADC sampling cycles for *ADC-ASYNC-DAQ* ADC data acquisition mode.

For application, which require high ADC timing accuracy while using simple ADC asynchronous data acquisition mode, *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* PIOX-16 AD/DA DCM also allows to initialize *ADC sampling cycles* on hardware defined sampling frequency events in case *ADC_ASYNC_DAQ_FS_EN* bit of *DAQ_CNTR1_RG* register (table 2-2) is set to the '1' state. In this case, hardware defined sampling frequency source is selected via bits {*FS_SEL-1, FS_SEL-0*} of *DAQ_SYNC_RG* register (table 2-5, refer to the corresponding subsection earlier).

CAUTION

ADC sampling cycles, which are initialized on hardware defined sampling frequency events, feature extremely small or even no time jitter for ADC analog input sampling events.

Note, that combination of hardware and software initialized ADC sampling cycle can be effectively used for A/D conversion of 8 and more analog inputs in one sampling frequency period, which can be generated by any of available hardware sampling frequency sources.

CAUTION

In case analog input multiplexer is updated by host DSP software during sampling frequency period, then either *ADC_ASYNC_DAQ_ACQ_EN* bit of *DAQ_SYNC_RG* register (table 2-5) must be set to the ‘1’ state in order to meet requirements of proper analog signal acquisition, or care must be taken by host DSP software in order to ensure that next A/D conversion does will start at least 0.25 µS after last ADC data ready event and at least 0.8 µS after last update of analog input multiplexer, whichever comes latest.

Figure 2-9 presents ADC sampling timing diagram for periodical ADC asynchronous data acquisition process, with each sampling frequency period comprising of two *ADC sampling cycles*. First ADC sampling cycle is initialized on hardware defined sampling frequency event (on-board PFG sampling frequency generator is assumed to be used as sampling frequency source), whereas second software initialized ADC sampling cycles inside the same sampling frequency period is used to perform A/D conversion of 8 analog inputs with ADC after analog input multiplexer update. Note, that in this example, large time jitter for ADC analog inputs events may apply to all software initialized ADC sampling frequency cycles, whereas small time jitter is applicable for ADC sampling events, which correspond to the first ADC sampling cycles inside each sampling frequency period, since these ADC sampling cycles are initialized on hardware defined sampling frequency source.

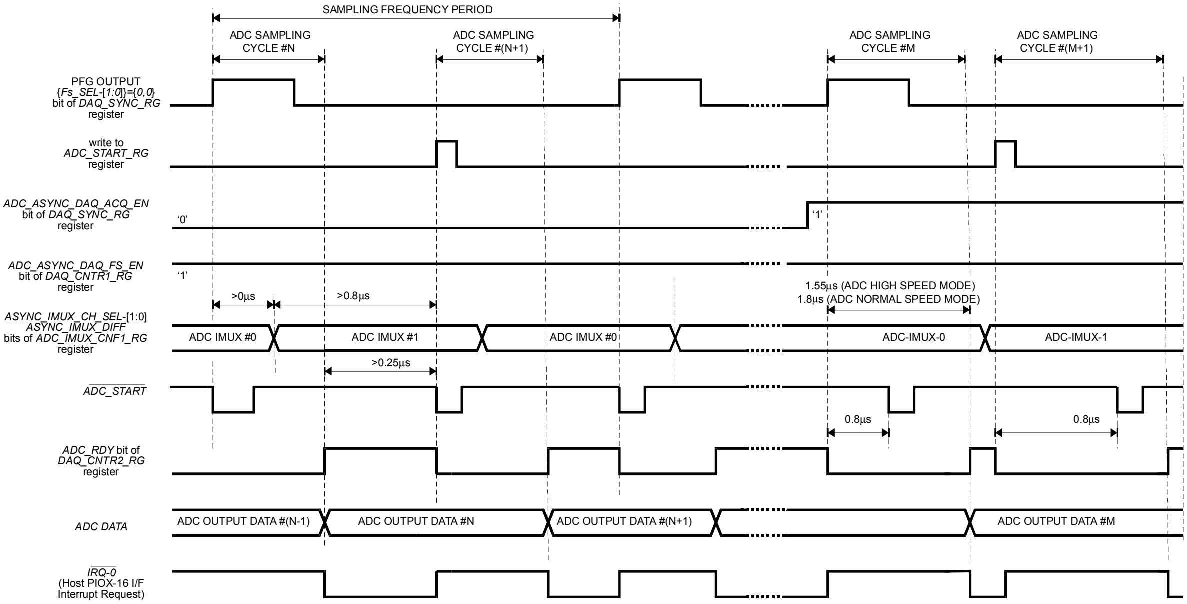
SAMPLING FREQUENCY PERIOD
ADC SAMPLING CYCLE #N
ADC SAMPLING CYCLE #(N+1)
ADC SAMPLING CYCLE #M
ADC SAMPLING CYCLE #(M+1)
PFG OUTPUT {Fs_SEL-[1:0]}={0,0} bit of DAQ_SYNC_RG register
write to ADC_START_RG register
ADC_ASYNC_DAQ_ACQ_EN bit of DAQ_SYNC_RG register
‘0’
‘1’
ADC_ASYNC_DAQ_FS_EN bit of DAQ_CNTR1_RG register
‘1’
>0µs
>0.8µs
1.55µs (ADC HIGH SPEED MODE)
1.8µs (ADC NORMAL SPEED MODE)
ASYNC_IMUX_CH_SEL-[1:0] ASYNC_IMUX_DIFF bits of ADC_IMUX_CNF1_RG register
ADC IMUX #0
ADC IMUX #1
ADC IMUX #0
ADC-IMUX-0
ADC-IMUX-1
>0.25µs
ADC_START
0.8µs
0.8µs
ADC_RDY bit of DAQ_CNTR2_RG register
ADC DATA
ADC OUTPUT DATA #(N-1)
ADC OUTPUT DATA #N
ADC OUTPUT DATA #(N+1)
ADC OUTPUT DATA #M
IRQ-0 (Host PIOX-16 I/F Interrupt Request)

Fig.2-9. Timing diagram for hardware defined sampling frequency periods with several ADC sampling cycles for *ADC-ASYNC-DAQ* ADC data acquisition mode.

ADC output data for each ADC sampling cycle for *ADC-ASYNC-DAQ* data acquisition mode is updated on ADC output data ready condition, which is indicated via bit *ADC_RDY* of *DAQ_CNTR2_RG* register (table 2-3), and can be accessed directly via host PIOX-16 interface either by host DSP software or by host DSP on-chip DMA controller synchronized by ADC IRT controller. ADC output data ready condition can be used to generate host PIOX-16 interrupt request (table 2-16).

For more details about how to download real-time ADC output data into host DSP environment for ADC asynchronous data acquisition mode, refer to subsection “Download of real-time ADC output data into host DSP environment for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes” below and to section “ADC IRT and DAC IRT Controllers” later in this chapter.

ADC data acquisition timing for ADC synchronous data acquisition modes (ADC-SYNC-PX-DAQ, ADC-SYNC-FIFO-OPM-DAQ, ADC-SYNC-FIFO-PTM-DAQ)

All *ADC synchronous* data acquisition modes (*ADC-SYNC-PX-DAQ*, *ADC-SYNC-FIFO-OPM-DAQ*, and *ADC-SYNC-FIFO-PTM-DAQ*) feature common synchronous pipelined timing diagram with high ADC sampling time accuracy, allow to perform automatic A/D conversion of up to 16 analog inputs during one sampling frequency period, and support external hardware start synchronization for ADC synchronous data acquisition process. High ADC sampling time accuracy, which is available with all *ADC synchronous* data acquisition modes, delivers ultra-low signal spectrum distortion for converted analog input signal, and is an ultimate requirement for most DSP algorithms.

ADC synchronous data acquisition modes differ in real-time ADC output data access only (either direct access via host PIOX-16 interface, or via ADC FIFO). Also, *ADC-SYNC-FIFO-OPM-DAQ* ADC data acquisition mode also allows to automatically terminate ADC synchronous data acquisition process when software selected ADC FIFO output flag comes true.

ADC synchronous data acquisition modes are a recommended selection in case ultra-low signal spectrum distortions, high sampling frequency, and A/D conversion of 8 and more analog inputs per one sampling frequency period are a must. Selection of particular *ADC synchronous* data acquisition mode is defined by particular maximum required ADC sampling frequency value and what particular method is preferred to download real-time ADC output data into host DSP environment (either directly via host PIOX-16 interface, or via ADC FIFO).

T/PDAS-AD16/16Q/1M-DA4/16Q/1M PIOX-16 AD/DA DCM provides flexible configuration and real-time control of ADC data acquisition controller for ADC synchronous data acquisition modes via host DSP software.

CAUTION

ADC synchronous data acquisition modes are selected in case {*ADC_DAQ_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register (table 2-2) are set either to the {0,1} state (*ADC-SYNC-PX-DAQ* data acquisition mode), or to the {1,0} state (*ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode), or to the {1,1} state (*ADC-SYNC-FIFO-PTM-DAQ* data acquisition mode).

In *ADC synchronous* data acquisition modes, ADC data acquisition controller continuously generates *ADC sampling packets*, which are initialized on hardware defined sampling frequency event and can include from one to four *ADC scan cycles*, which are used to perform A/D conversion with different settings of ADC analog input multiplexers. Four *ADC scan cycles* inside one *ADC sampling packet* allow to perform A/D conversion of all 16 analog inputs inside one sampling frequency period.

CAUTION

Sampling frequency for all ADC synchronous data acquisition modes is sourced from hardware sampling frequency sources as defined by {*FS_SEL-1, FS_SEL-0*} bits of *DAQ_SYNC_RG* register (table 2-5, refer to the corresponding subsection earlier).

Each sampling frequency event performs initialization of new *ADC sampling frequency packet* in case there is no active *ADC sampling frequency packet* currently in progress.

Each *ADC scan cycle* performs synchronous sampling and A/D conversion of analog outputs of analog input multiplexers for all ADC-0..3 channels independently of the state of *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6), except for exceptional condition when *ADC0_EN..ADC3_EN* bits are all set to the '0' state. *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register are used by ADC data acquisition controller in *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* ADC synchronous data acquisition modes only in order to select particular ADC channels, which will be read and pushed into ADC FIFO in each *ADC scan cycle*.

CAUTION

In case all *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6) are set to the '0' state, then ADC synchronous data acquisition process will not initialize.

Each *ADC scan cycle* controller, which runs inside each *ADC sampling packets* for all ADC synchronous data acquisition modes, is controlled via {*SCCLx_IMUX_CH_SEL*-[1:0]} bits (*x*=0..3), *SCCLx_IMUX_DIFF* bits (*x*=0..3), and *SCCLx_IMUX_DIFF* bits (*x*=0..2) of *ADC_IMUX_CNF1_RG* and *ADC_IMUX_CNF2_RG* registers (table 2-8b).

CAUTION

Bits {*SCCLx_IMUX_CH_SEL*-[1:0]} and *SCCLx_IMUX_DIFF* (*x*=0..3) are used to configure analog input multiplexers for all ADC channels in the corresponding *ADC scan cycles* #0..#3 of each *ADC sampling packet* of ADC synchronous data acquisition process.

Bits *SCCLx_EOS* (*x*=0..2) are used to define last *ADC scan cycle* (#0..#2, *ADC scan cycle* #3 is default last *ADC scan cycle*) for each *ADC sampling packet* of ADC synchronous data acquisition process.

Last *ADC scan cycle* inside each *ADC sampling packet* for ADC synchronous data acquisition process corresponds to the first of the *SCCL0_EOS*, *SCCL1_EOS* and *SCCL2_EOS* bits of *ADC_IMUX_CNF1_RG* and *ADC_IMUX_CNF2_RG* registers (table 2-8b), which is set to the '1' state. In case all *SCCL0_EOS*, *SCCL1_EOS* and *SCCL2_EOS* bits are set to the '0' state, then four *ADC scan cycles* will be generated.

Figure 2-10 presents timing diagram for a part of ADC synchronous data acquisition process comprising of a series of *ADC sampling packets*, which are initialized on selected hardware defined sampling frequency event (Fs) and each containing three *ADC scan cycles*.

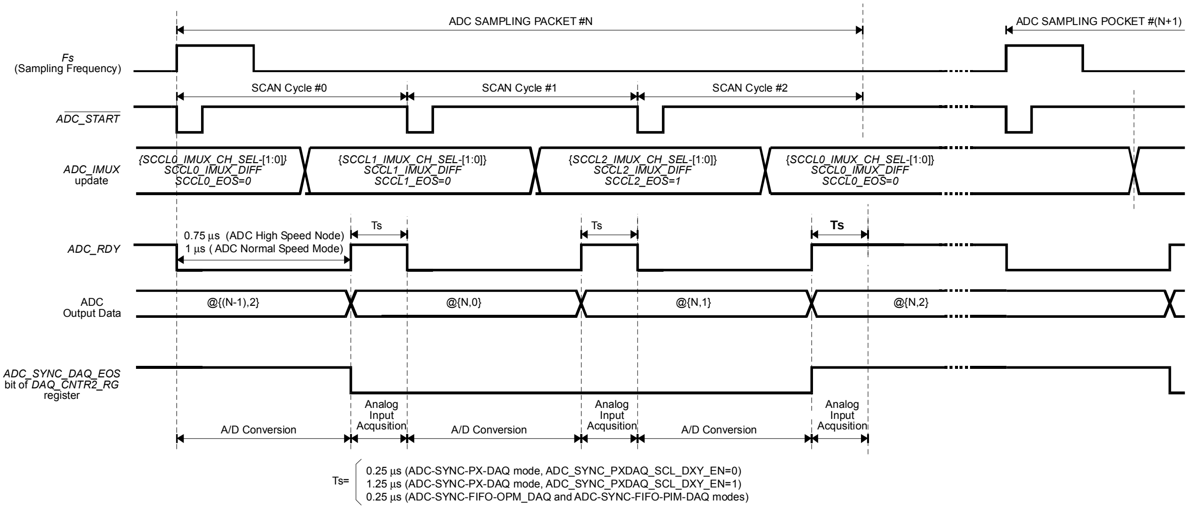

Fig.2-10. Timing diagram for ADC synchronous data process with ADC sampling packets comprising of three ADC scan cycles.

Once *ADC sampling packet* has been initialized, then ADC scan cycle #0 is automatically generated and A/D conversion is started. Analog input multiplexers have been already configured in the previous scan cycle (last scan cycle of previous ADC sampling packet) using {*SCCL0_IMUX_CH_SEL*-[1:0]} and *SCCL0_IMUX_DIFF* bits of *ADC_IMUX_CNF1_RG* register (table 2-8b), so analog signals at ADC inputs have been already set to sufficient accuracy prior A/D conversion starts. Once A/D conversion has been started and ADC input signals have been acquired (sampled), then analog input multiplexers are updated with the contents of {*SCCL1_IMUX_CH_SEL*-[1:0]} and *SCCL1_IMUX_DIFF* bits of *ADC_IMUX_CNF1_RG* register (table 2-8b) in order to set ADC analog inputs properly to sufficient accuracy prior next A/D conversion will start in next ADC scan cycle #1.

A/D conversion time for each ADC scan cycle is defined upon the state of *ADC_HIGH_SPEED_EN* bit of *DAQ_SYNC_RG* register (table 2-5). In case *ADC_HIGH_SPEED_EN* bit is set to default '0' state, then *ADC normal speed mode* is selected, and A/D conversion time is 1 µS. In case *ADC_HIGH_SPEED_EN* bit is set to the '1' state, then *ADC high-speed mode* is selected, and A/D conversion time is 0.75 µS.

CAUTION

Minimum sampling frequency for *ADC high-speed mode*
is 1 kHz.

In case *ADC high-speed mode* is used and ADC data acquisition controller was not running for more than 1 mS prior start of ADC synchronous data acquisition process, then ADC output data for the first ADC scan cycle of the first ADC sampling packet must be ignored.

CAUTION

There are no minimum sampling frequency restrictions for *ADC normal speed mode*, and it can be as low as 0 Hz.

ADC output data for *ADC normal speed mode* is valid for all ADC sampling packets starting from the first ADC scan cycle.

CAUTION

ADC high-speed mode is a recommended selection with ADC synchronous data acquisition modes with the sampling frequency 10 kHz and above.

ADC normal-speed mode is a recommended selection with ADC synchronous data acquisition modes with the sampling frequency below 10 kHz.

ADC output data ready condition is indicated by *ADC_RDY* read-only bit of *DAQ_CNTR2_RG* register (table 2-3). ADC output data is updated on ADC output data ready condition, and shall be read prior next ADC output data will get ready, otherwise ADC output data will be overwritten. *ADC_RDY* read-only bit can be also used to generate host PIOX-16 interrupt request (table 2-16).

For *ADC-SYNC-PX-DAQ* data acquisition mode, real-time ADC output data can be accessed directly via host PIOX-16 interface either by host DSP software or by host DSP on-chip DMA controller synchronized by ADC IRT controller. For more details about how to download real-time ADC output data into host DSP environment for *ADC-SYNC-PX-DAQ* data acquisition mode, refer to subsection “Download of real-time ADC output data into host DSP environment for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes” below and to section “ADC IRT and DAC IRT Controllers” later in this chapter.

For *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes, real-time ADC output data cannot be directly read via host PIOX-16 interface. Instead, ADC output data is read by ADC data acquisition controller in each ADC scan cycle and pushed into high-density ADC FIFO for all enabled

ADC channels in accordance with the state of *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6). Buffering of real-time ADC output data in high-density ADC FIFO allows to significantly offload host DSP data bus traffic and perform read of ADC FIFO data later as soon as it will be required by data processing algorithm. ADC FIFO output data can be read either by host DSP software or host DSP on-chip DMA controller synchronized by ADC IRT controller. For more details about how to download ADC FIFO output data into host DSP environment for ADC synchronous FIFO data acquisition modes, refer to subsection "Download of ADC FIFO output data into host DSP environment for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes" below and to section "ADC IRT and DAC IRT Controllers" later in this chapter.

ADC scan cycle does not terminate immediately after ADC output data ready condition occurs. Instead, *ADC scan cycle* prolongs extra *Ts* time pause after ADC output data ready condition in order to allow ADC analog inputs toe acquire input signal properly to sufficient accuracy prior next A/D conversion will start in next ADC scan cycle. Duration of *Ts* time pause is always 0.25 µS for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes, and is selected by *ADC_SYNC_PX_DAQ_SCCL_DLY_EN* bit of *DAQ_SYNC_RG* register (table 2-5) from either 0.25 µS or 1.25 µS values for *ADC-SYNC-PX-DAQ* data acquisition mode only.

CAUTION

Duration of *ADC scan cycle* for ADC high-speed mode for *ADC-SYNC-PX-DAQ* data acquisition mode is either 1 µS or 2 µS depending upon the state of *ADC_SYNC_PX_DAQ_SCCL_DLY_EN* bit of *DAQ_SYNC_RG* register (table 2-5).

Duration of *ADC scan cycle* for ADC normal speed mode for *ADC-SYNC-PX-DAQ* data acquisition mode is either 1.25 µS or 2.25 µS depending upon the state of *ADC_SYNC_PX_DAQ_SCCL_DLY_EN* bit of *DAQ_SYNC_RG* register (table 2-5).

Duration of *ADC scan cycle* for ADC high-speed mode for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes is always 1 µS.

Duration of *ADC scan cycle* for ADC normal speed mode for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes is always 1.25 µS.

Default *Ts* = 0.25 µS setting for *ADC-SYNC-PX-DAQ* data acquisition mode corresponds to maximum performance and is selected in case *ADC_SYNC_PX_DAQ_SCCL_DLY_EN* bit of *DAQ_SYNC_RG* register (table 2-5) is set to the '0' state. Alternative selection is *Ts* = 1.25 µS setting for *ADC-SYNC-PX-DAQ* data acquisition mode corresponds to maximum A/D conversion performance and is selected in case *ADC_SYNC_PX_DAQ_SCCL_DLY_EN* bit of *DAQ_SYNC_RG* register (table 2-5) is set to the '0' state. However, in case *Ts* is set to the 0.25 µS for *ADC-SYNC-PX-DAQ* data acquisition mode, then care must be taken to ensure that host DSP software (or host DSP on-chip DMA controller synchronized by ADC IRT controller) provides sufficient traffic over PIOX-16 interface data bus in order to read ADC output data for all enabled ADC channels within either 1 µS or 1.25 µS time interval (depending upon selected ADC speed mode, which is defined by bit *ADC_HIGH_SPEED_EN* bit of *DAQ_SYNC_RG* register) prior next ADC output data ready condition will occur and ADC output data for current ADC scan cycle will be overwritten with new data. This setting works well, for example, in case only ADC synchronous data acquisition process is in progress with all ADC channels enabled and DAC synchronous data acquisition process is not running.

However, in case both ADC and DAC synchronous data acquisition processes are running with all ADC/DAC channels enabled, then this my provide hard traffic over PIOX-16 interface data bus and host DSP data bus correspondingly. Instead, relaxed *Ts* = 1.25 μS setting provides extra time room for host DSP software (or host DSP on-chip DMA controller synchronized by ADC IRT controller) to download real-time ADC output data via host PIOX-16 interface.

Next *ADC scan cycle*, if requested, will start immediately after *Ts* time pause inside current ADC scan cycle will expire. At figure 2-10, three ADC scan cycles are requested as it is defined via bits *SCCL0_EOS* and *SCCL1_EOS* of *ADC_IMUX_CNF1_RG* register, which are both set to the '0' state. Since bit *SCCL2_EOS* of *ADC_IMUX_CNF2_RG* register is set to the '1' state, then *ADC scan cycle* #3 will terminate *ADC sampling packet*, and ADC data acquisition controller will wait for the next hardware sampling frequency event in order to initialize next *ADC sampling packet,* which will feature absolutely identical timing diagram.

In order to indicate valid last ADC scan cycle output data for host DSP software for *ADC-SYNC-PX-DAQ* data acquisition mode, *ADC_SYNC_DAQ_EOS* bit of *DAQ_CNTR2_RG* register (table 2-3) is provided, which is updated on ADC output data ready condition and is set to the '1' state in the last *ADC scan cycle* of each *ADC sampling packet*. For *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes, last ADC scan cycle output data indicator is pushed into ADC FIFO by ADC data acquisition controller along with ADC output data during last *ADC scan cycle* of each *ADC sampling packet*, and is available for read from ADC FIFO on host PIOX-16 interface side via *ADC_SCCL_EOS* bit of *ADC_FIFO_DATA_MSW_RG* register (table 2-11).

High ADC sampling time accuracy for all ADC synchronous data acquisition modes is guaranteed by design of ADC data acquisition controller, which provides accurate predefined timing for all ADC synchronous data acquisition modes. Thus, absolute maximum time jitter for the first A/D sampling event inside each *ADC sampling packet* is only ± 10 ns, whereas all succeeding A/D sampling events inside the same *ADC sampling packet* are internally synchronized to the first A/D sampling event without time jitter. This provides extremely low spectrum distortion for digitized signal for the sampling frequencies up to 1 MHz.

DAC data acquisition timing for DAC asynchronous data acquisition mode (DAC-ASYNC-DAQ)

DAC asynchronous data acquisition mode (*DAC-ASYNC-DAQ*) features simple timing and is a recommended selection in case DAC outputs shall be updated on either software defined or hardware defined sampling frequency events. *DAC asynchronous* data acquisition mode is the most simple to configure and to use DAC data acquisition mode, which delivers flexibility in D/A conversion control via host DSP software and offers direct access to DAC input data registers via host PIOX-16 interface.

CAUTION

DAC-ASYNC-DAQ data acquisition mode is selected in case {*DAC_DAQ_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register are set to the {0,0} state (table 2-2).

Operation of DAC data acquisition controller in *DAC asynchronous* data acquisition mode comprises of a series of *DAC sampling cycles*. Each *DAC sampling cycle* provides synchronous D/A conversion and sampling of analog outputs for all DAC-0..3 channels, i.e. actually generates common DAC output data load signal only.

CAUTION

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM uses double-buffered D/A converter chips architecture (refer to fig2-4) with independent DAC input data register and DAC output data register, so each *DAC sampling cycle* performs synchronous D/A conversion for all DAC chips by generation of common DAC output data load signal, which is 100..120 ns long.

CAUTION

DAC0_EN..DAC3_EN bits of *DAC_CNF_RG* register (table 2-6) are ignored by DAC data acquisition controller in *DAC asynchronous* data acquisition mode, and are used by DAC IRT controller only in order to automatically download DAC output data into host DSP environment (refer to section "DAC IRT and DAC IRT Controllers" later in this chapter for more details).

CAUTION

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM provides DAC analog output settling time 1 µS after DAC output data load signal is generated.

In *DAC asynchronous* data acquisition mode, DAC data acquisition controllers can be configured to generate *DAC sampling cycles* (actually generated is common DAC output data load signal) in different modes (on different source events) as it is defined by {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register (table 2-2).

Transparent DAC output data load mode for DAC asynchronous data acquisition mode is selected in case {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE*-[1:0]} bits are set to the {0,0} state. In this mode (fig.2-11a), common DAC output data load signal is generated immediately by DAC data acquisition controller after host DSP performs write of DAC input data to any of *DACx_DATA_RG* registers (x=0..3) via host PIOX-16 interface, and analog output of each DAC chip is updated immediately after DAC data is being written. Although all DAC outputs are sampled synchronously, this mode really does not allow simultaneous update of all DAC outputs, since each DAC output is updated as soon as this DAC is being written by host DSP, whereas other DAC outputs remain in their previous state. This means, that in one software defined 'virtual' sampling frequency period, all DAC outputs are updated at different times. In this mode, time jitter for DAC output data load event is defined by DSP software. This mode is useful when there is no sampling frequency, and DAC data is loaded on software defined events. *Transparent DAC output data load mode* is a default setting on host PIOX-16 interface reset condition.

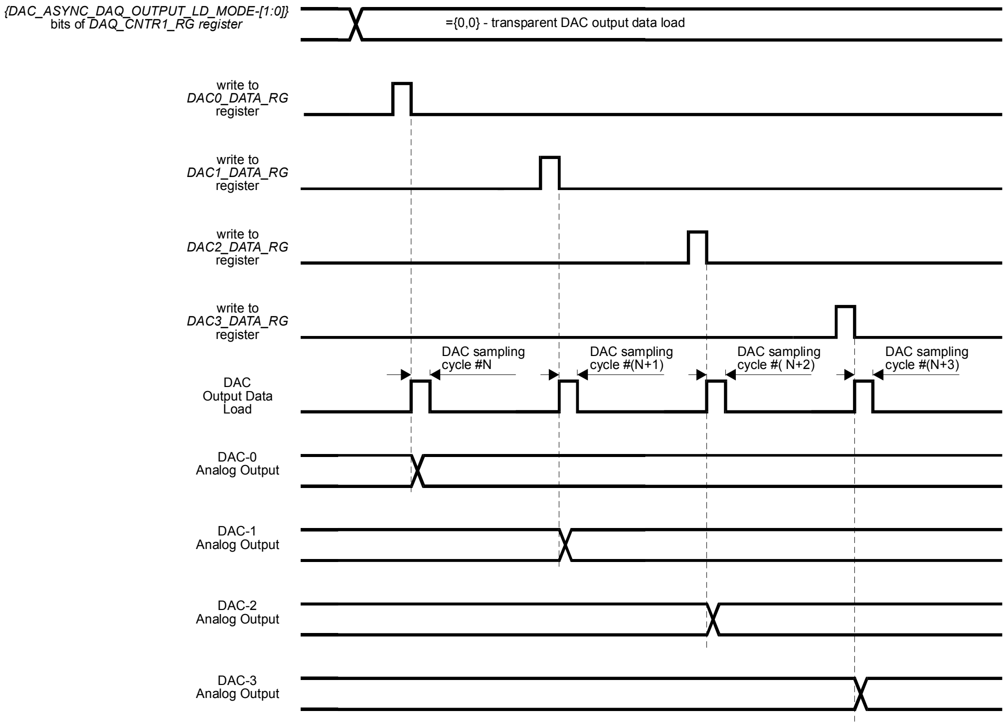
{DAC_ASYNC_DAQ_OUTPUT_LD_MODE-[1:0]} bits of DAQ_CNTR1_RG register
={0,0} - transparent DAC output data load
write to DAC0_DATA_RG register
write to DAC1_DATA_RG register
write to DAC2_DATA_RG register
write to DAC3_DATA_RG register
DAC Output Data Load
DAC sampling cycle #N
DAC sampling cycle #(N+1)
DAC sampling cycle #(N+2)
DAC sampling cycle #(N+3)
DAC-0 Analog Output
DAC-1 Analog Output
DAC-2 Analog Output
DAC-3 Analog Output

Fig.2-11a. Timing diagram for transparent DAC output data load mode for *DAC-ASYNC-DAQ* data acquisition mode.

DAC output data load on software defined event mode for DAC asynchronous data acquisition mode is selected in case {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE*-[1:0]} bits are set to the {0,1} state. In this mode (fig.2-11b), common DAC output data load signal is generated on host DSP write to either *DAC_OUTPUT_LD_RG* or *ADC_START_DAC_OUTPUT_LD_RG* registers. This mode provides synchronous sampling of all DAC outputs on software defined event. This means, that in one software defined 'virtual' sampling frequency period, all DAC outputs are updated synchronously after individual DAC input data have been written to each DAC input register. In this mode, time jitter for DAC output data load event is defined by DSP software, and there is no guaranteed maximum for its value. This mode is useful when there is no sampling frequency, and DAC data is loaded on software defined events, although it is required that all DAC outputs shall be updated synchronously.

CAUTION

Once host DSP writes to either DAC_OUTPUT_LD_RG or ADC_START_DAC_OUTPUT_LD_RG registers, then this will result in generation of common DAC output data load signal for each of the DAC output data load modes independent of the state of {DAC_ASYNC_DAQ_OUTPUT_LD_MODE-[1:0]} bits of DAQ_CNTR1_RG register during DAC asynchronous data acquisition mode.

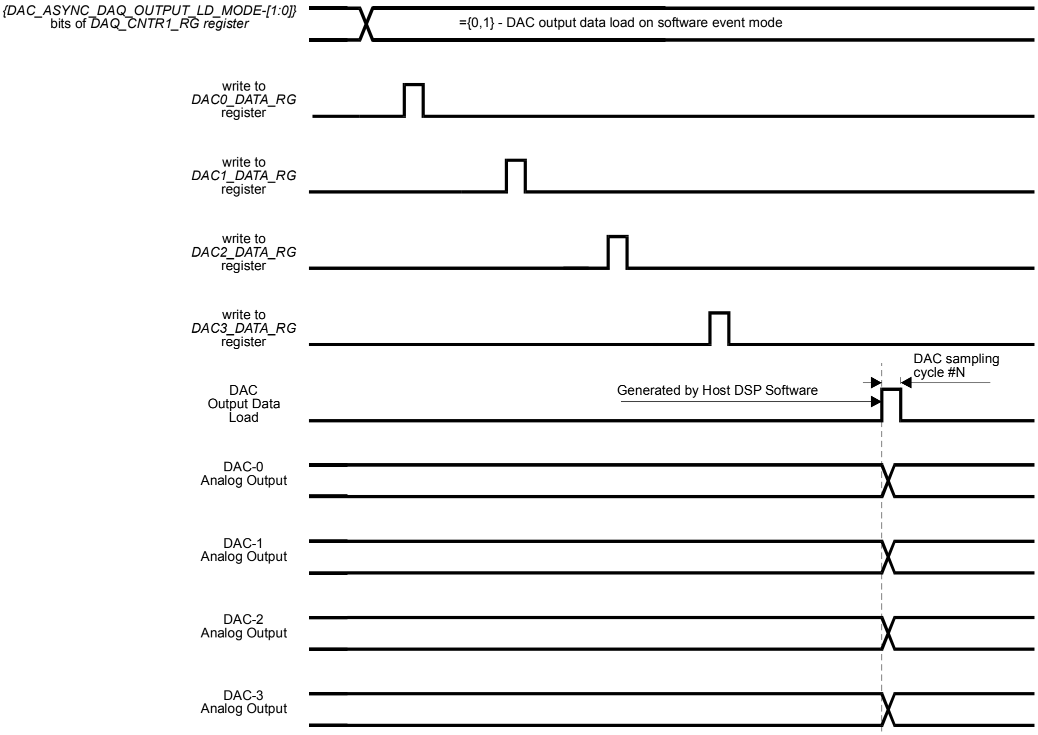
{DAC_ASYNC_DAQ_OUTPUT_LD_MODE-[1:0]} bits of DAQ_CNTR1_RG register
={0,1} - DAC output data load on software event mode
write to DAC0_DATA_RG register
write to DAC1_DATA_RG register
write to DAC2_DATA_RG register
write to DAC3_DATA_RG register
DAC sampling cycle #N
DAC Output Data Load
Generated by Host DSP Software
DAC-0 Analog Output
DAC-1 Analog Output
DAC-2 Analog Output
DAC-3 Analog Output

Fig.2-11b. Timing diagram for DAC output data load on software defined event mode for *DAC-ASYNC-DAQ* data acquisition mode.

DAC output data load on hardware defined sampling frequency event mode for DAC asynchronous data acquisition mode is selected in case {DAC_ASYNC_DAQ_OUTPUT_LD_MODE-[1:0]} bits are set to the {1,0} state. In this mode (fig.2-11c), common DAC output data load signal is generated on hardware defined sampling frequency events with the sampling frequency source being selected via bits {*Fs_SEL-[1:0]*} of DAQ_SYNC_RG register (table 2-5). This mode features most accurate DAC output sampling timing without added time jitter for DAC output data load event, and provides true synchronous sampling of all DAC outputs. That guarantees minimum spectrum distortions for DAC analog output signal. This mode is very similar to DAC synchronous data acquisition with DAC write from PIOX-16 interface mode (DAC-SYNC-PX-DAQ) except for it does not provide start synchronization for DAC acquisition process. This mode useful when there is hardware defined sampling frequency, and all DAC outputs shall be sampled synchronously without time jitter.

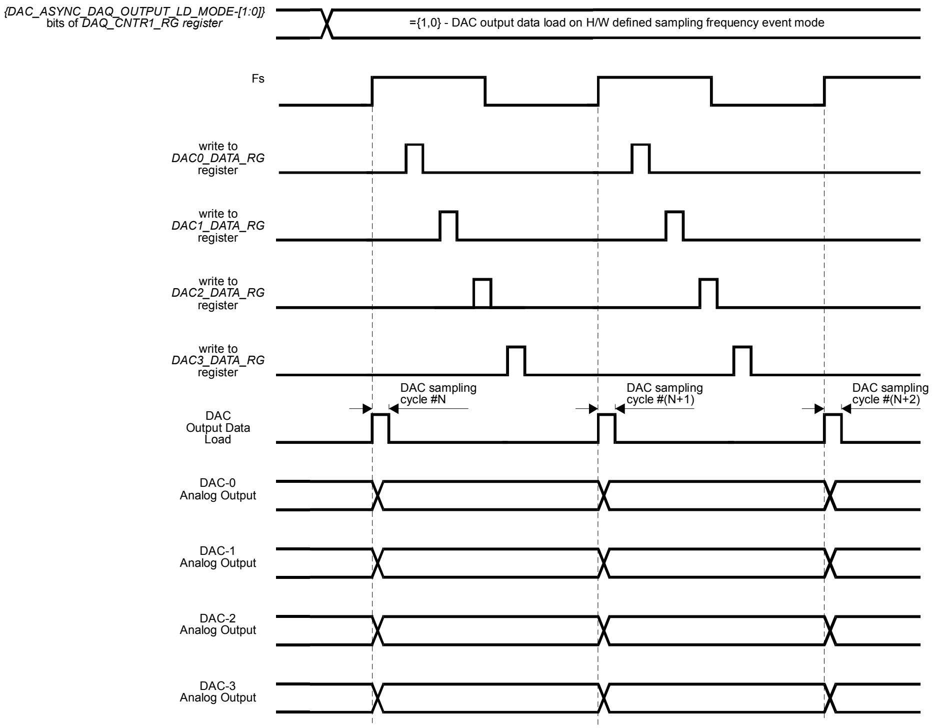
{DAC_ASYNC_DAQ_OUTPUT_LD_MODE-[1:0]}
bits of DAQ_CNTR1_RG register
={1,0} - DAC output data load on H/W defined sampling frequency event mode
Fs
write to DAC0_DATA_RG register
write to DAC1_DATA_RG register
write to DAC2_DATA_RG register
write to DAC3_DATA_RG register
DAC sampling cycle #N
DAC sampling cycle #(N+1)
DAC sampling cycle #(N+2)
DAC Output Data Load
DAC-0 Analog Output
DAC-1 Analog Output
DAC-2 Analog Output
DAC-3 Analog Output

Fig.2-11c. Timing diagram for DAC output data load on hardware defined sampling frequency event mode for *DAC-ASYNC-DAQ* data acquisition mode.

DAC output data load on the end of DAC IRT data transfer mode for DAC asynchronous data acquisition mode is selected in case {*DAC_ASYNC_DAQ_OUTPUT_LD_MODE*-[1:0]} bits are set to the {1,1} state. In this mode (fig.2-11d), common DAC output data load signal is generated on the end of each DAC IRT data transfer cycle. Each DAC IRT data transfer cycle is initialized on the hardware defined sampling frequency event with the sampling frequency source being selected via bits {*Fs_SEL-[1:0]*} of *DAQ_SYNC_RG* register (table 2-5). Each DAC IRT data transfer cycle performs DAC input data write for all enabled DAC channels in accordance with the state of *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-6). Although this mode provides synchronous sampling of all DAC outputs, the maximum time jitter for DAC output data load event cannot be guaranteed, since the end of DAC IRT data transfer cycle depends upon the DSP data bus traffic. This mode is useful when DSP on-chip DMA controller synchronized by DAC IRT controller is being used for real-time DAC data transfer, whereas DAC output data load is required immediately after all DAC input data have been written in the same hardware defined sampling frequency period. For more details about operation of DAC IRT controller, refer to section “ADC IRT and DAC IRT Controllers” later in this chapter

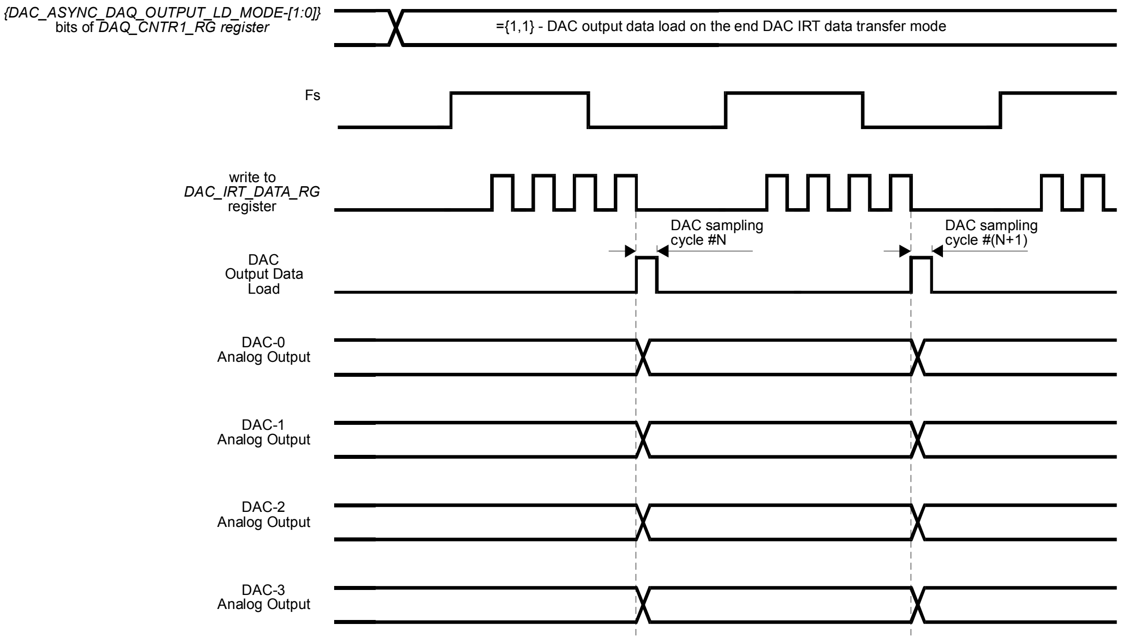
{DAC_ASYNC_DAQ_OUTPUT_LD_MODE-[1:0]} bits of DAQ_CNTR1_RG register
={1,1} - DAC output data load on the end DAC IRT data transfer mode
Fs
write to DAC_IRT_DATA_RG register
DAC sampling cycle #N
DAC sampling cycle #(N+1)
DAC Output Data Load
DAC-0 Analog Output
DAC-1 Analog Output
DAC-2 Analog Output
DAC-3 Analog Output

Fig.2-11d. Timing diagram for DAC output data load on the end of DAC IRT data transfer mode for *DAC-ASYNC-DAQ* data acquisition mode.

For details about how to upload real-time DAC input data from host DSP environment for DAC asynchronous data acquisition mode, refer to subsection "Upload of real-time DAC input data from host DSP environment for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes" below and to section "ADC IRT and DAC IRT Controllers" later in this chapter.

DAC data acquisition timing for DAC synchronous data acquisition modes (DAC-SYNC-PX-DAQ, DAC-SYNC-FIFO-OPM-DAQ, DAC-SYNC-FIFO-PTM-DAQ)

All *DAC synchronous* data acquisition modes (*DAC-SYNC-PX-DAQ*, *DAC-SYNC-FIFO-OPM-DAQ*, and *DAC-SYNC-FIFO-PTM-DAQ*) feature common synchronous pipelined timing diagram with high DAC sampling time accuracy, and support external hardware start synchronization for DAC synchronous data acquisition process. High DAC sampling time accuracy, which is available with all *DAC synchronous* data acquisition modes, delivers ultra-low signal spectrum distortion for converted analog output signal, and is an ultimate requirement for most DSP algorithms.

DAC synchronous data acquisition modes differ in real-time DAC input data access only (either direct access via host PIOX-16 interface, or via DAC FIFO). Also, *DAC-SYNC-FIFO-OPM-DAQ* DAC data acquisition mode allows to automatically terminate DAC synchronous data acquisition process when software selected DAC FIFO output flag comes true.

DAC synchronous data acquisition modes are a recommended selection in case ultra-low signal spectrum distortions, high sampling frequency, and external start synchronization are a must. Selection of particular *DAC synchronous* data acquisition mode is defined by particular maximum required DAC sampling frequency value and what particular method is preferred to upload real-time DAC input data from host DSP environment (either directly via host PIOX-16 interface, or via ADC FIFO).

CAUTION

DAC synchronous data acquisition modes are selected in case {*DAC_DAQ_MODE*-[1:0]} bits of *DAQ_CNTR1_RG* register (table 2-2) are set either to the {0,1} state (*DAC-SYNC-PX-DAQ* data acquisition mode), or to the {1,0} state (*DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode), or to the {1,1} state (*DAC-SYNC-FIFO-PTM-DAQ* data acquisition mode).

All *DAC synchronous* data acquisition modes feature simple pipelined data acquisition timing (fig.2-12), which is similar to *DAC output data load on hardware defined sampling frequency event mode* for DAC asynchronous data acquisition mode (fig.2-11c). The difference is the added start synchronization procedure for DAC synchronous data acquisition mode, which is described in the corresponding subsection below.

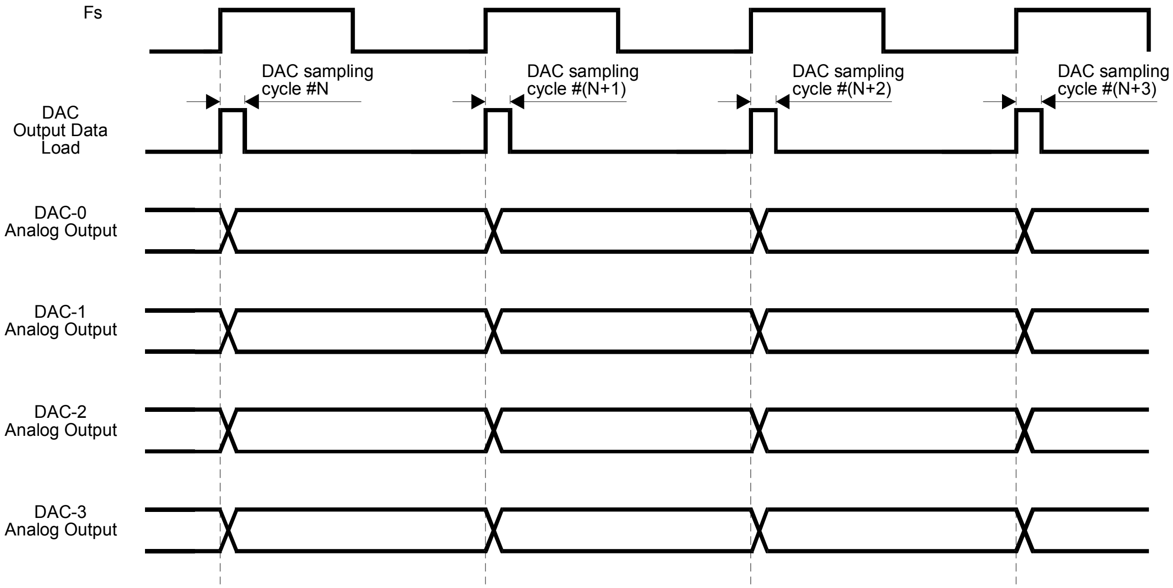

Fig.2-12. Timing diagram for DAC synchronous data acquisition process.

In all *DAC synchronous* data acquisition modes, DAC data acquisition controller continuously generates *DAC sampling cycles* (DAC output data load signal) on hardware defined sampling frequency events with the sampling frequency source being selected via bits {*Fs_SEL-[1:0]*} of *DAQ_SYNC_RG* register (table 2-5). *DAC synchronous* data acquisition modes feature accurate DAC output sampling timing without added time jitter for DAC output data load event, and provide true synchronous sampling of all DAC outputs. That guarantees minimum spectrum distortions for DAC analog output signals.

CAUTION

In all DAC synchronous data acquisition modes, only hardware defined sampling frequency events are used to initialize *DAC sampling cycles*.

DAC input data, which is copied from DAC input register to DAC output data register (refer to fig.2-4) on hardware defined sampling frequency event in each DAC sampling cycle, shall be written to DAC input register one DAC sampling cycle in advance in order to provide pipelined operation.

For *DAC-SYNC-PX-DAQ* data acquisition mode, real-time DAC input data can be written directly via host PIOX-16 interface either by host DSP software or by host DSP on-chip DMA controller synchronized by DAC IRT controller. For more details about how to upload real-time DAC input data from host DSP environment for *DAC-SYNC-PX-DAQ* data acquisition mode, refer to subsection "Upload of real-time DAC input data from host DSP environment for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes" below and to section "ADC IRT and DAC IRT Controllers" later in this chapter.

For *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes, real-time DAC output data cannot be directly written via host PIOX-16 interface. Instead, DAC input data is written by DAC data acquisition controller in each DAC scan cycle from high-density DAC FIFO for all enabled DAC channels in accordance with the state of *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-6). Temporary buffering of real-time DAC input data in high-density DAC FIFO allows to significantly offload host DSP data bus traffic and perform fill-in of DAC FIFO data in advance as it will be required by data processing algorithm. DAC FIFO input data can be filled-in either by host DSP software or host DSP on-chip DMA controller synchronized by DAC IRT controller. For more details about how to upload DAC FIFO input data from host DSP environment for DAC synchronous FIFO data acquisition modes, refer to subsection "Upload of DAC FIFO input data from host DSP environment for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes" below and to section "ADC IRT and DAC IRT Controllers" later in this chapter.

general control for ADC/DAC synchronous data acquisition processes in ADC/DAC synchronous data acquisition modes

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM offers a set of DSP software controlled features, which allow to provide general control of ADC and DAC synchronous data acquisition processes procurement for ADC and DAC synchronous data acquisition modes correspondingly. These features include:

- ADC/DAC synchronous data acquisition start and stop commands (for all ADC/DAC synchronous data acquisition modes)
- selection of start synchronization source for ADC/DAC synchronous data acquisition process (for all ADC/DAC synchronous data acquisition modes)
- automatic termination of ADC and DAC synchronous data acquisition process after acquisition of desired number of ADC sampling packets for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode and after output of desired number of DAC sampling cycles for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode correspondingly
- detection of real-time ADC/DAC FIFO error conditions (for *ADC-SYNC-FIFO-PTM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition mode only).

This subsection will provide details and timing diagrams for general control options for all ADC/DAC synchronous data acquisition modes.

Figures 2-13a, 2-13b and 2-13c provide timing diagrams for procurement of ADC synchronous data acquisition process for *ADC-SYNC-PX-DAQ*, *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* ADC synchronous data acquisition modes correspondingly.

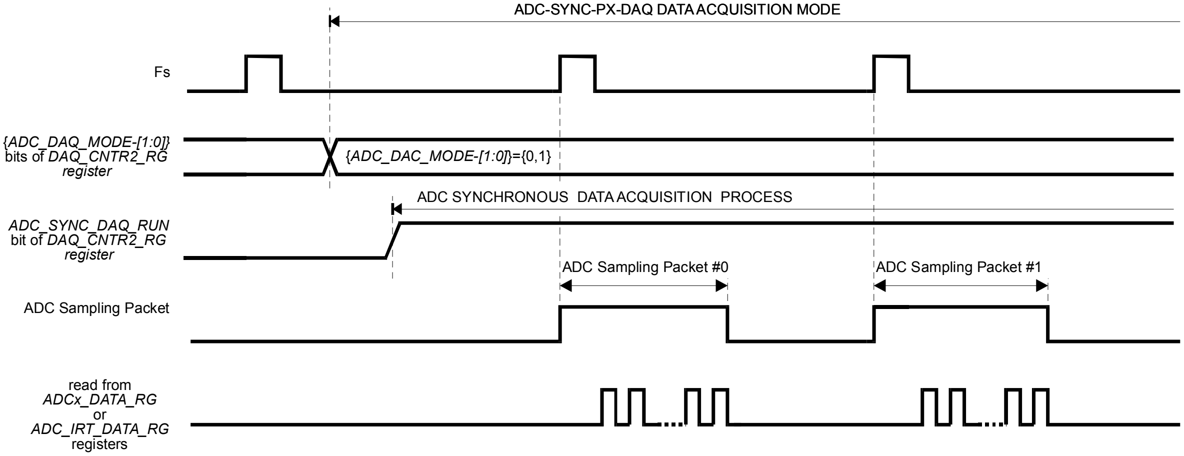
ADC-SYNC-PX-DAQ DATA ACQUISITION MODE
Fs
{ADC_DAQ_MODE-[1:0]} bits of DAQ_CNTR2_RG register
{ADC_DAC_MODE-[1:0]}={0,1}
ADC SYNCHRONOUS DATA ACQUISITION PROCESS
ADC_SYNC_DAQ_RUN bit of DAQ_CNTR2_RG register
ADC Sampling Packet #0
ADC Sampling Packet #1
ADC Sampling Packet
read from ADCx_DATA_RG or ADC_IRT_DATA_RG registers

Fig.2-13a. Timing diagram for procurement of ADC synchronous data acquisition process for *ADC-SYNC-PX-DAQ* data acquisition mode.

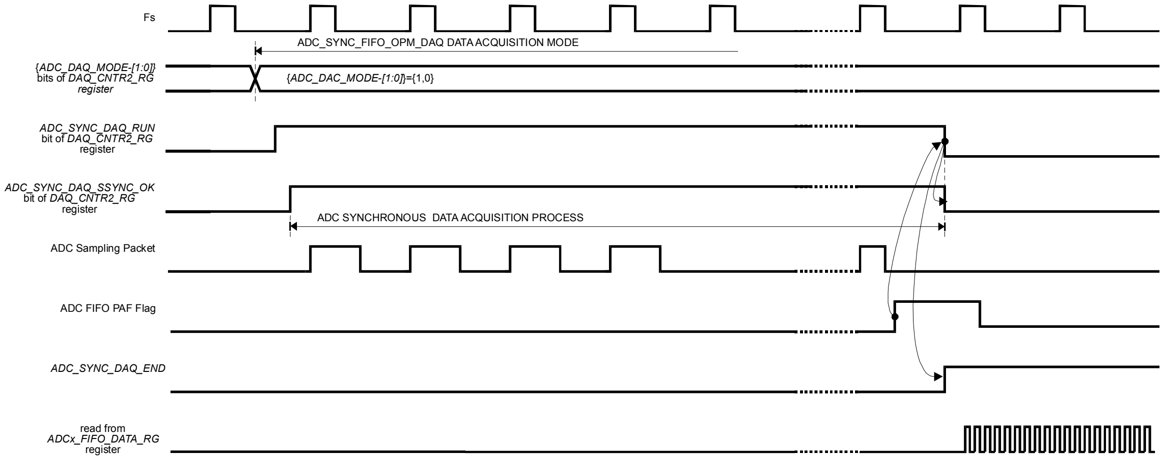
Fs
ADC_SYNC_FIFO_OPM_DAQ DATA ACQUISITION MODE
{ADC_DAQ_MODE-[1:0]} bits of DAQ_CNTR2_RG register
{ADC_DAC_MODE-[1:0]}={1,0}
ADC_SYNC_DAQ_RUN bit of DAQ_CNTR2_RG register
ADC_SYNC_DAQ_SSYNC_OK bit of DAQ_CNTR2_RG register
ADC SYNCHRONOUS DATA ACQUISITION PROCESS
ADC Sampling Packet
ADC FIFO PAF Flag
ADC_SYNC_DAQ_END
read from ADCx_FIFO_DATA_RG register

Fig.2-13b. Timing diagram for procurement of ADC synchronous data acquisition process for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode.

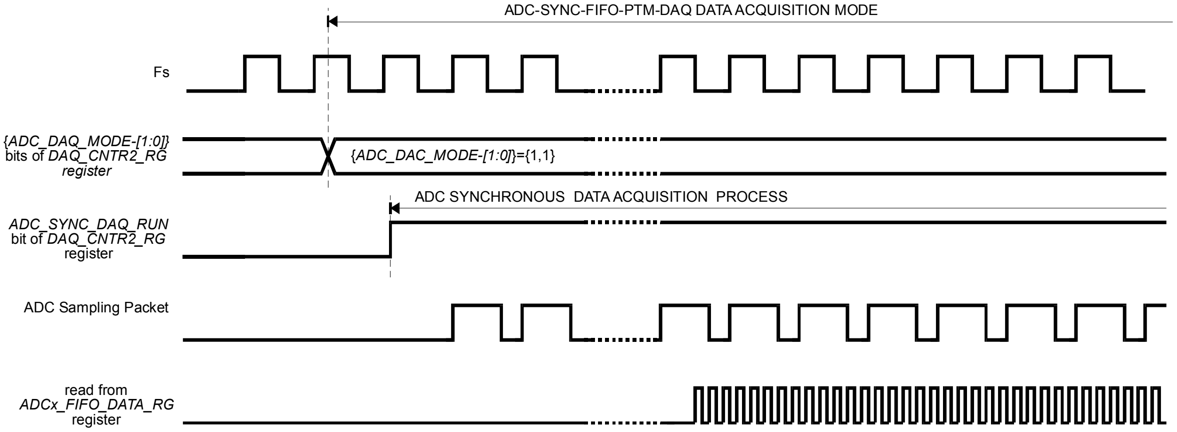
ADC-SYNC-FIFO-PTM-DAQ DATA ACQUISITION MODE
Fs
{ADC_DAQ_MODE-[1:0]} bits of DAQ_CNTR2_RG register
{ADC_DAC_MODE-[1:0]}={1,1}
ADC SYNCHRONOUS DATA ACQUISITION PROCESS
ADC_SYNC_DAQ_RUN bit of DAQ_CNTR2_RG register
ADC Sampling Packet
read from ADCx_FIFO_DATA_RG register

Fig.2-13c. Timing diagram for procurement of ADC synchronous data acquisition process for *ADC-SYNC-FIFO-PTM-DAQ* data acquisition mode.

CAUTION

Timing diagrams for procurement of ADC synchronous data acquisition processes for different ADC synchronous data acquisition modes at figures 2-13a, 2-13b and 2-13c do not provide details about start synchronization of ADC synchronous data acquisition process, generation of ADC sampling packets, and real-time ADC and ADC FIFO data read.

For more details about start synchronization of ADC synchronous data acquisition process, ADC sampling packets, and download of real-time ADC output data and ADC FIFO data read refer to the corresponding subsections in this section.

Figures 2-14a, 2-14b and 2-14c provide timing diagrams for procurement of DAC synchronous data acquisition process for *DAC-SYNC-PX-DAQ*, *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* DAC synchronous data acquisition modes correspondingly.

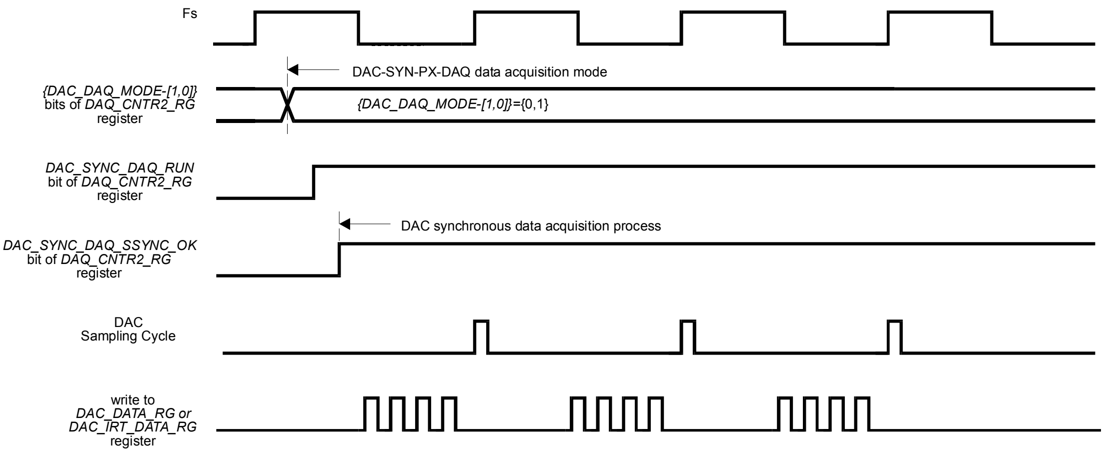
Fs
DAC-SYN-PX-DAQ data acquisition mode
{DAC_DAQ_MODE-[1,0]} bits of DAQ_CNTR2_RG register
{DAC_DAQ_MODE-[1,0]}={0,1}
DAC_SYNC_DAQ_RUN bit of DAQ_CNTR2_RG register
DAC synchronous data acquisition process
DAC_SYNC_DAQ_SSYNC_OK bit of DAQ_CNTR2_RG register
DAC Sampling Cycle
write to DAC_DATA_RG or DAC_IRT_DATA_RG register

Fig.2-14a. Timing diagram for procurement of DAC synchronous data acquisition process for *DAC-SYNC-PX-DAQ* data acquisition mode.

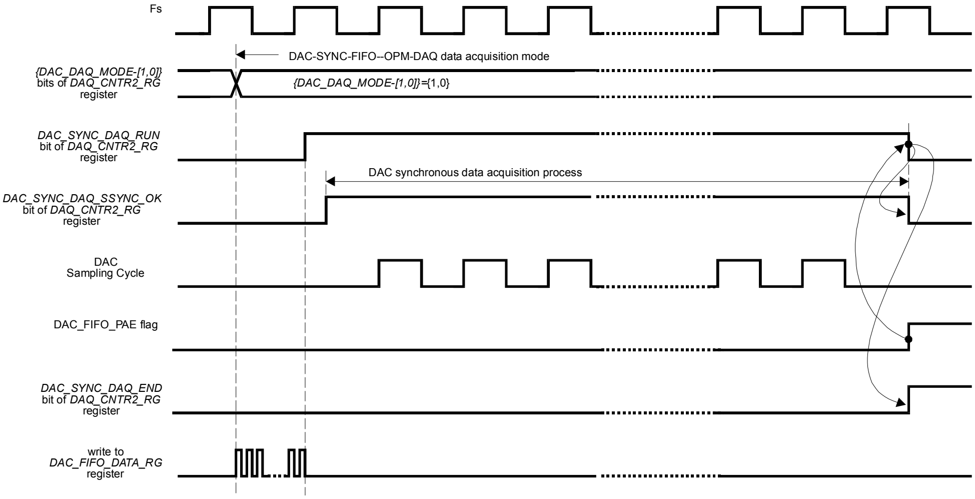
Fs
DAC-SYNC-FIFO--OPM-DAQ data acquisition mode
{DAC_DAQ_MODE-[1,0]} bits of DAQ_CNTR2_RG register
{DAC_DAQ_MODE-[1,0]}={1,0}
DAC_SYNC_DAQ_RUN bit of DAQ_CNTR2_RG register
DAC synchronous data acquisition process
DAC_SYNC_DAQ_SSYNC_OK bit of DAQ_CNTR2_RG register
DAC Sampling Cycle
DAC_FIFO_PAE flag
DAC_SYNC_DAQ_END bit of DAQ_CNTR2_RG register
write to DAC_FIFO_DATA_RG register

Fig.2-14b. Timing diagram for procurement of DAC synchronous data acquisition process for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode.

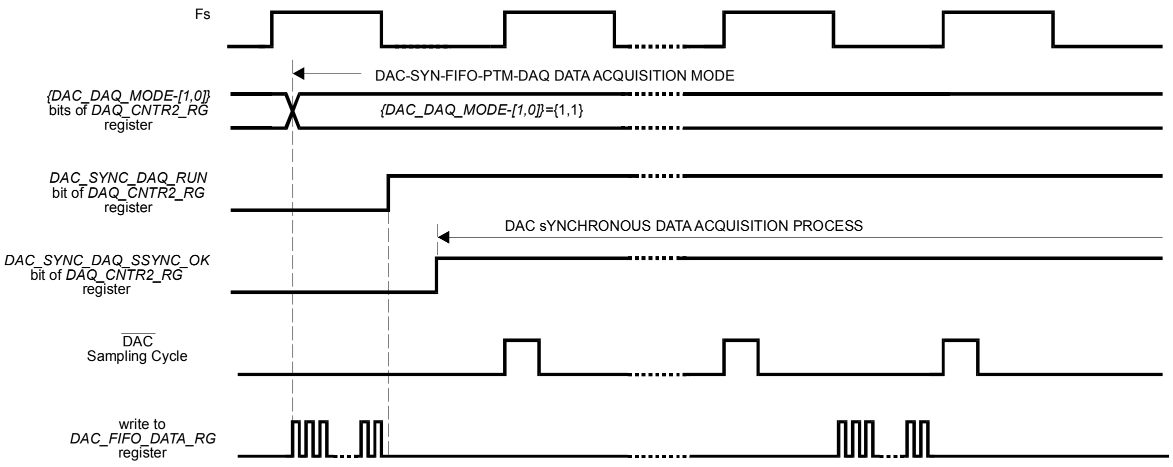
Fs
DAC-SYN-FIFO-PTM-DAQ DATA ACQUISITION MODE
{DAC_DAQ_MODE-[1,0]} bits of DAQ_CNTR2_RG register
{DAC_DAQ_MODE-[1,0]}={1,1}
DAC_SYNC_DAQ_RUN bit of DAQ_CNTR2_RG register
DAC sYNCHRONOUS DATA ACQUISITION PROCESS
DAC_SYNC_DAQ_SSYNC_OK bit of DAQ_CNTR2_RG register
DAC Sampling Cycle
write to DAC_FIFO_DATA_RG register

Fig.2-14c. Timing diagram for procurement of DAC synchronous data acquisition process for *DAC-SYNC-FIFO-PTM-DAQ* data acquisition mode.

CAUTION

Timing diagrams for procurement of DAC synchronous data acquisition process for different DAC synchronous data acquisition modes at figures 2-14a, 2-14b and 2-14c do not provide details about start synchronization of DAC synchronous data acquisition process, and real-time DAC input data and DAC FIFO data write.

For more details about start synchronization of DAC synchronous data acquisition process, and upload of real-time DAC input data and DAC FIFO input data write refer to the corresponding subsections in this section.

ADC/DAC synchronous data acquisition process for all ADC/DAC synchronous data acquisition modes is activated by setting *ADC_SYNC_DAQ_RUN* and *DAC_SYNC_DAQ_RUN* bits correspondingly of *DAQ_CNTR2_RG* register (table 2-3) to the '1' state and after detection of start synchronization event, which is indicated via *ADC_SYNC_DAQ_SSYNC_OK* and *DAC_SYNC_DAQ_SSYNC_OK* bit of *DAQ_CNTR2_RG* register (refer to table 2-3 and to the corresponding subsection below in this section).

Once started, ADC/DAC synchronous data acquisition process will begin continuously generate *ADC sampling packets* and *DAC sampling cycles* correspondingly as it has been described in subsections "ADC data acquisition timing for ADC synchronous data acquisition modes" and "DAC data acquisition timing for DAC synchronous data acquisition modes" earlier in this section.

ADC synchronous data acquisition process never terminates itself for *ADC-SYNC-PX-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes (fig.2-13a and 2-13c), and DAC synchronous data acquisition process never terminates itself for *DAC-SYNC-PX-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes (fig.2-14a and 2-14c).

For *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode (fig.2-13b), ADC synchronous data acquisition process can be automatically terminated on software selected either ADC FIFO FF or ADC FIFO PAF flag condition in accordance with the state of *ADC_SYNC_FIFO_DAQ_TF_SEL* bit of *DAQ_CNTR3_RG* register (table 2-4). In case *ADC_SYNC_FIFO_DAQ_TF_SEL* bit of *DAQ_CNTR3_RG* register is set to default ‘0’ state, then ADC FIFO FF (FIFO full) flag is selected to normally terminate ADC synchronous data acquisition process for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode, otherwise ADC FIFO PAF (FIFO partially-full) flag is selected.

Correspondingly, for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode (fig.2-14b), DAC synchronous data acquisition process can be automatically terminated on software selected either DAC FIFO EF or DAC FIFO PAE flag condition in accordance with the state of *DAC_SYNC_FIFO_DAQ_TF_SEL* bit of *DAQ_CNTR3_RG* register (table 2-4). In case *DAC_SYNC_FIFO_DAQ_TF_SEL* bit of *DAQ_CNTR3_RG* register is set to default ‘0’ state, then ADC FIFO EF (FIFO empty) flag is selected to normally terminate DAC synchronous data acquisition process for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode, otherwise ADC FIFO PAE (FIFO partially-empty) flag is selected.

CAUTION

ADC FIFO FF flag must be used as normal data acquisition termination event for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode in case user application needs to use full FIFO depth (2^{18}=262,144 samples) in order to store real-time ADC output data.

Programmable ADC FIFO PAF flag must be used as normal data acquisition termination event for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode in case user application needs to store short packet of real-time ADC output data in ADC FIFO and does not need to use full ADC FIFO depth (2^{18}=262,144 samples).

ADC FIFO PAF flag offset can be configured by host DSP software in ADC asynchronous data acquisition mode (also known as ADC FIFO configuration mode) to any predefined number of stored ADC FIFO samples (refer to subsection “Programming ADC/DAC FIFO flags offset” earlier in this section).

CAUTION

DAC FIFO EF flag must be used as normal data acquisition termination event for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode in case user application needs to use full FIFO depth (2^{18}=262,144 samples) in order to store real-time DAC input data. Host DSP software must fill-in full DAC FIFO prior activating *DAC-SYNC-FIFO-OPM-DAQ* data acquisition process.

Programmable DAC FIFO PAE flag must be used as normal data acquisition termination event for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode in case user application needs to use a fraction of DAC FIFO in order to store real-time DAC input data, and does not need to use full DAC FIFO depth (2^{18}=262,144 samples). Host DSP software must write desired number of DAC samples to DAC FIFO prior activating *DAC-SYNC-FIFO-OPM-DAQ* data acquisition process.

DAC FIFO PAE flag offset can be configured by host DSP software in DAC asynchronous data acquisition mode (also known as DAC FIFO configuration mode) to any predefined number of stored DAC FIFO samples (refer to subsection "Programming ADC/DAC FIFO flags offset" earlier in this section).

Figure 2-13b shows normal termination of ADC synchronous data acquisition process on ADC FIFO PAF flag condition for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode. After ADC FIFO write pointer will become equal to pre-programmed offset value for ADC FIFO PAF flag (in case ADC FIFO read pointer was not updated, otherwise the difference between ADC FIFO write and read pointers values must be used), then ADC FIFO PAF flag output will bet set to the '1' state (table 2-20), and ADC synchronous data acquisition process for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode will terminate normally as shown at figure 2-13b. Normal termination of ADC synchronous data acquisition process for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode on ADC FIFO flag condition is indicated via *ADC_SYNC_DAQ_END* bit of *DAQ_CNTR2_RG* register (table 2-3). *ADC_SYNC_DAQ_END* bit will remain in the '1' state until it will be either reset automatically by initializing new ADC synchronous data acquisition process, or can be reset by host DSP software along with abort of ADC synchronous data acquisition process. Active '1' state of *ADC_SYNC_DAQ_END* bit of *DAQ_CNTR2_RG* register for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode can be also used to generate host PIOX-16 interrupts (table 2-16).

Correspondingly, figure 2-14b shows normal termination of DAC synchronous data acquisition process on DAC FIFO PAE flag condition for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode. After DAC FIFO read pointer will become equal to pre-programmed offset value for DAC FIFO PAE flag (in case DAC FIFO write pointer was not updated, otherwise the difference between DAC FIFO write and read pointers values must be used), then DAC FIFO PAE flag output will bet set to the '1' state (table 2-20), and DAC synchronous data acquisition process for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode will terminate normally as shown at figure 2-14b. Normal termination of DAC synchronous data acquisition process for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode on DAC FIFO flag condition is indicated via *DAC_SYNC_DAQ_END* bit of *DAQ_CNTR2_RG* register (table 2-3). *DAC_SYNC_DAQ_END* bit will remain in the '1' state until it will be either reset automatically by initializing new DAC synchronous data acquisition process, or can be reset by host DSP software along with abort of DAC synchronous data acquisition process. Active '1' state of *DAC_SYNC_DAQ_END* bit of *DAQ_CNTR2_RG* register for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode can be also used to generate host PIOX-16 interrupts (table 2-16).

For all ADC synchronous data acquisition modes, ADC synchronous data acquisition process can be aborted anytime by host DSP software by either setting *ADC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register to the ‘1’ state, or reset of ADC data acquisition controller on write to either of *ADC_DAQ_RESET_RG* and *ADDA_DAQ_RESET_RG* registers (written data is ignored for writes to *ADC_DAQ_RESET_RG* and *ADDA_DAQ_RESET_RG* registers).

Correspondingly, for all DAC synchronous data acquisition modes, DAC synchronous data acquisition process can be aborted anytime by host DSP software by either setting *DAC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register to the ‘1’ state, or reset of DAC data acquisition controller on write to either of *DAC_DAQ_RESET_RG* and *ADDA_DAQ_RESET_RG* registers (written data is ignored for writes to *DAC_DAQ_RESET_RG* and *ADDA_DAQ_RESET_RG* registers).

CAUTION

Reset of ADC data acquisition controller during *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* ADC synchronous data acquisition modes by write to either *ADC_DAQ_RESET_RG* or *ADDA_DAQ_RESET_RG* registers also performs reset of ADC FIFO read and write pointers, thus making already stored ADC FIFO data unavailable for read via host PIOX-16 interface.

Instead, normal abort procedure for ADC synchronous data acquisition process for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes via setting *ADC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register will just stop ADC FIFO data acquisition process and freeze last value of ADC FIFO write pointer. Stored ADC FIFO data will remain available for normal read via host PIOX-16 interface. After aborting, ADC synchronous data acquisition process can be restarted by setting *ADC_SYNC_DAQ_RUN* bit of *DAQ_CNTR2_RG* register to the ‘1’ state and detection of new start synchronization event.

CAUTION

Reset of DAC data acquisition controller during *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* DAC synchronous data acquisition modes by write to either *DAC_DAQ_RESET_RG* or *ADDA_DAQ_RESET_RG* registers also performs reset of DAC FIFO read and write pointers, thus making already stored DAC FIFO data unavailable for further output to DAC input registers.

Instead, normal abort procedure for DAC synchronous data acquisition process for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes via setting *DAC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register will just stop DAC FIFO data acquisition process and freeze last value of DAC FIFO read pointer. Unread DAC FIFO data will still remain available for further output to DAC input registers. After aborting, DAC synchronous data acquisition process can be restarted by setting *DAC_SYNC_DAQ_RUN* bit of *DAQ_CNTR2_RG* register to the '1' state and detection of new start synchronization event.

For *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes, real-time ADC output data cannot be directly read via host PIOX-16 interface. Instead, ADC output data is read by ADC data acquisition controller in each ADC scan cycle and pushed into high-density ADC FIFO for all enabled ADC channels for further read of ADC FIFO output data either via host DSP software or host DSP on-chip DMA controller synchronized via ADC IRT controller.

CAUTION

For *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode, it is not recommended to read ADC FIFO during running ADC synchronous data acquisition process, since this will result in modification of ADC FIFO read pointer on each read from ADC FIFO and termination of ADC synchronous data acquisition process on incorrect number of acquired and stored ADC sampling packets.

For *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes, real-time DAC input data cannot be directly written via host PIOX-16 interface. Instead, DAC input data is read by DAC data acquisition controller in each DAC scan cycle from high-density DAC FIFO for all enabled DAC channels, however either host DSP software or host DSP on-chip DMA controller synchronized via DAC IRT controller must upload DAC FIFO input data prior starting DAC synchronous data acquisition process.

CAUTION

For *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode, it is not recommended to write DAC FIFO during running DAC synchronous data acquisition process, since this will result in modification of DAC FIFO write pointer on each write to DAC FIFO and termination of DAC synchronous data acquisition process on incorrect number of DAC sampling cycles.

start synchronization for ADC/DAC synchronous data acquisition processes for ADC/DAC synchronous data acquisition modes

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM provides flexible software configured selection of start synchronization option for ADC/DAC synchronous data acquisition process for all ADC/DAC synchronous data acquisition modes (*ADC-SYNC-PX-DAQ*, *ADC-SYNC-FIFO-OPM-DAQ*, *ADC-SYNC-FIFO-PTM-DAQ* and *DAC-SYNC-PX-DAQ*, *DAC-SYNC-FIFO-OPM-DAQ*, *DAC-SYNC-FIFO-PTM-DAQ* correspondingly).

Start synchronization option allows to trigger start synchronization event for ADC/DAC synchronous data acquisition process and allows to synchronize start of acquisition by external hardware after ADC/DAC synchronous data acquisition process has been started by setting *ADC_SYNC_DAQ_RUN* and *DAC_SYNC_DAQ_RUN* bits correspondingly of *DAQ_CNTR2_RG* register (table 2-3) to the ‘1’ state (refer to subsection “General control for ADC/DAC synchronous data acquisition process in ADC synchronous data acquisition modes” earlier in this section).

CAUTION

Start synchronization option for ADC/DAC data acquisition controllers of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM is not supported in ADC/DAC asynchronous data acquisition mode (*ADC-ASYNC-DAQ* and *ADC-ASYNC-DAQ* correspondingly).

Selection of start synchronization mode for ADC/DAC synchronous data acquisition process from host DSP software for ADC data acquisition controller is performed via *ADC_SYNC_DAQ_SSYNC_SEL* and *DAC_SYNC_DAQ_SSYNC_SEL* bits correspondingly of *DAQ_SYNC_RG* register (table 2-5) as the following:

- in case *ADC_SYNC_DAQ_SSYNC_SEL* bit is set to default ‘0’ state, then *software defined start synchronization* is used to trigger start synchronization event for ADC synchronous data acquisition process. Correspondingly, in case *DAC_SYNC_DAQ_SSYNC_SEL* bit is set to default ‘0’ state, then *software defined start synchronization* is used to trigger start synchronization event for DAC synchronous data acquisition process. These are default settings on host PIOX-16 interface reset condition.
- in case *ADC_SYNC_DAQ_SSYNC_SEL* bit is set to the ‘1’ state, then *external hardware defined start synchronization* via *ADC_XSSYNC_IN* input pin of on-board JP3 external synchronization and I/O connector is used to trigger start synchronization event for ADC synchronous data acquisition process. Correspondingly, in case *DAC_SYNC_DAQ_SSYNC_SEL* bit is set to the ‘1’ state, then *external hardware defined start synchronization* via *DAC_XSSYNC_IN* input pin of on-board JP3 external synchronization and I/O connector is used to trigger start synchronization event for DAC synchronous data acquisition process. In case *external hardware defined start synchronization* via

ADC_XSSYNC_IN and/or *DAC_XSSYNC_IN* input pins of on-board JP3 connector has been selected, then active event at *ADC_XSSYNC_IN* and *DAC_XSSYNC_IN* input pins correspondingly is selected via bit *XSSYNC_MODE* of *DAQ_SYNC_RG* register (table 2-5) as the following:

- in case *XSSYNC_MODE* bit is set to ‘0’ state, then *active low* condition at *ADC_XSSYNC_IN* and *DAC_XSSYNC_IN* input pins of on-board JP3 connector is used to detect external hardware defined start synchronization events for ADC and DAC synchronous data acquisition process correspondingly. This setting is default for PIOX-16 interface reset condition.
- in case *XSSYNC_MODE* bit is set to the ‘1’ state, then the *falling edge* condition at *ADC_XSSYNC_IN* and *DAC_XSSYNC_IN* input pins of on-board JP3 connector is used to detect external hardware defined start synchronization events for ADC and DAC synchronous data acquisition processes correspondingly.

CAUTION

XSSYNC_MODE bit of *DAQ_SYNC_RG* register (table 2-5) provides identical common selection of active event for external hardware defined start synchronization for both ADC and DAC data acquisition processes.

Figures 2-15a, 2-15b and 2-15c present timing diagrams for *software defined start synchronization*, *active low external hardware defined start synchronization* and *falling edge triggered external hardware defined start synchronization* correspondingly for ADC synchronous data acquisition process.

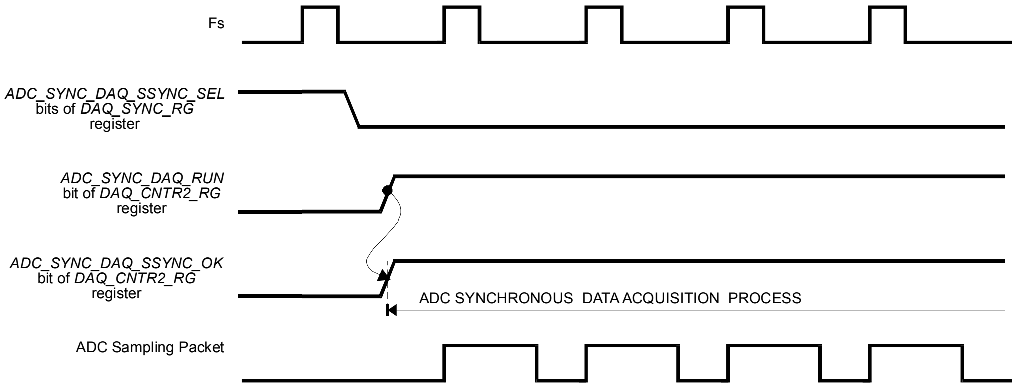
Fs
ADC_SYNC_DAQ_SSYNC_SEL bits of DAQ_SYNC_RG register
ADC_SYNC_DAQ_RUN bit of DAQ_CNTR2_RG register
ADC_SYNC_DAQ_SSYNC_OK bit of DAQ_CNTR2_RG register
ADC SYNCHRONOUS DATA ACQUISITION PROCESS
ADC Sampling Packet

Fig.2-15a. Timing diagram for software defined start synchronization for ADC synchronous data acquisition process.

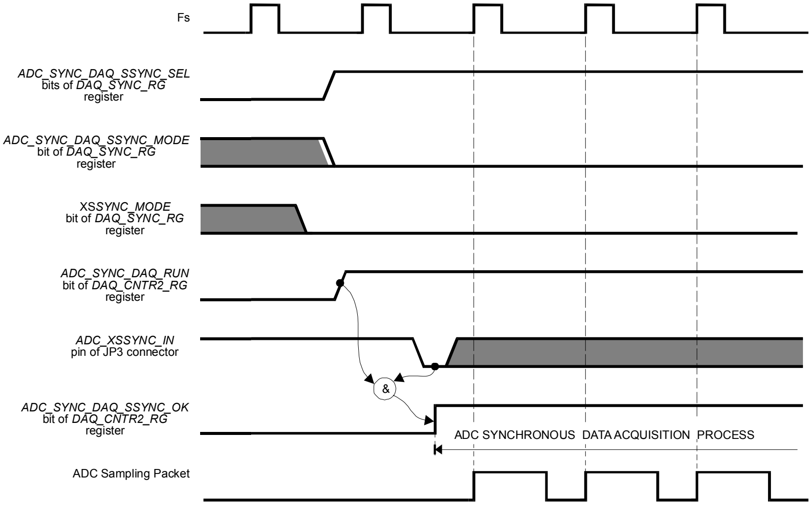

Fig.2-15b. Timing diagram for active low external hardware defined start synchronization for ADC synchronous data acquisition process.

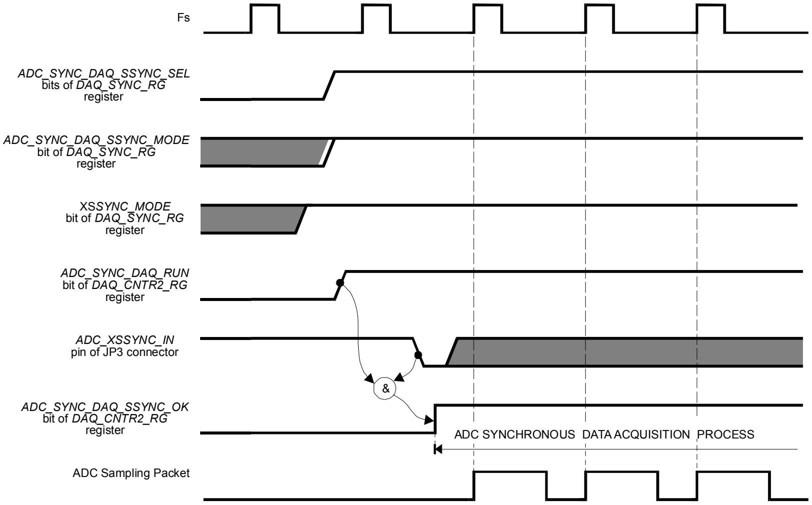

Fig.2-15c. Timing diagram for falling edge triggered external hardware defined start synchronization for ADC synchronous data acquisition process.

Figures 2-16a, 2-16b and 2-16c present timing diagrams for *software defined start synchronization*, *active low external hardware defined start synchronization* and *falling edge triggered external hardware defined start synchronization* correspondingly for ADC synchronous data acquisition process.

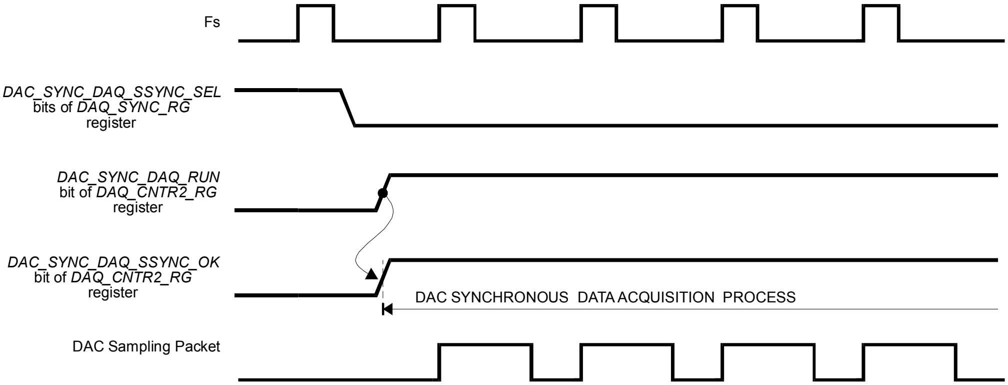

Fig.2-16a. Timing diagram for software defined start synchronization for DAC synchronous data acquisition process.

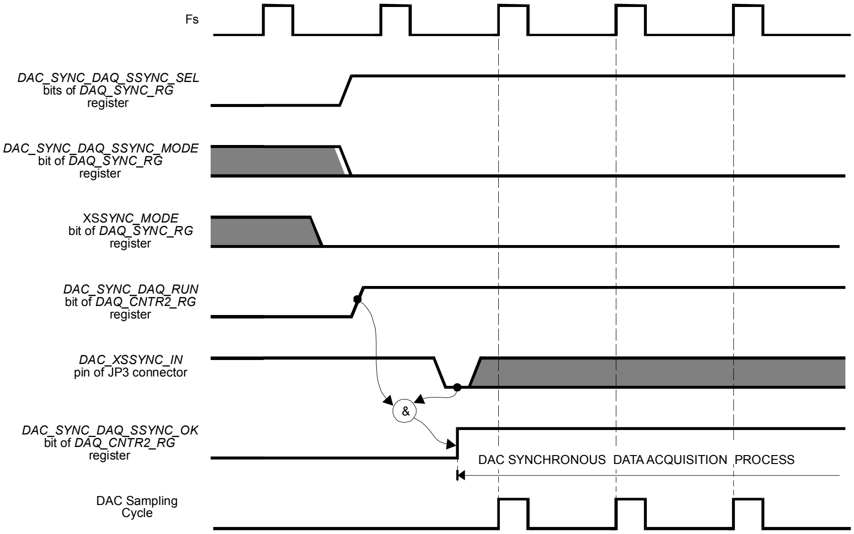

Fig.2-16b. Timing diagram for active low external hardware defined start synchronization for DAC synchronous data acquisition process.

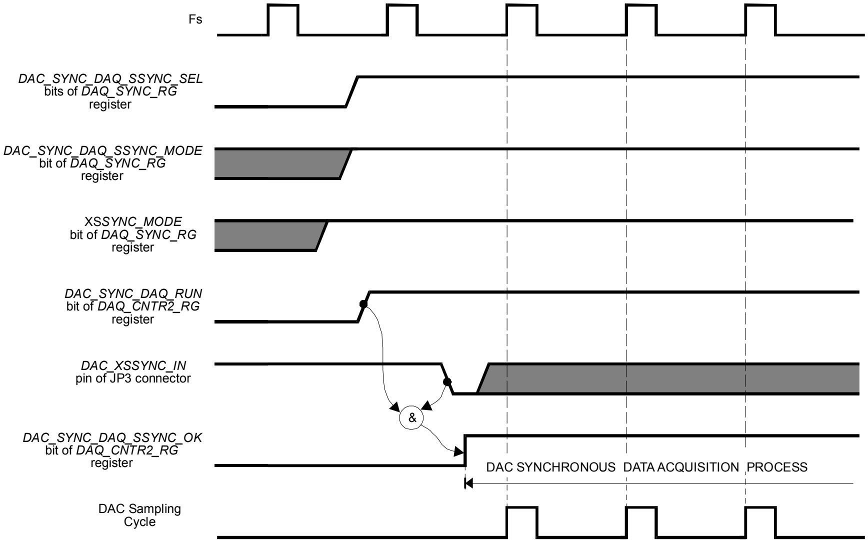
Fs
DAC_SYNC_DAQ_SSYNC_SEL bits of DAQ_SYNC_RG register
DAC_SYNC_DAQ_SSYNC_MODE bit of DAQ_SYNC_RG register
XSSYNC_MODE bit of DAQ_SYNC_RG register
DAC_SYNC_DAQ_RUN bit of DAQ_CNTR2_RG register
DAC_XSSYNC_IN pin of JP3 connector
&
DAC_SYNC_DAQ_SSYNC_OK bit of DAQ_CNTR2_RG register
DAC SYNCHRONOUS DATA ACQUISITION PROCESS
DAC Sampling Cycle

Fig.2-16c. Timing diagram for falling edge triggered external hardware defined start synchronization for DAC synchronous data acquisition process.

Software defined start synchronization (fig.2-15a and 2-16a) must be selected in case ADC synchronous data acquisition processes shall be started immediately after host DSP software will set *ADC_SYNC_DAQ_RUN* and *DAC_SYNC_DAQ_RUN* bits correspondingly of *DAQ_CNTR2_RG* register (table 2-3) without any need to be synchronized by external hardware. In this case, the read-only *ADC_SYNC_DAQ_SSYNC_OK* and *DAC_SYNC_DAQ_SSYNC_OK* bits correspondingly of *DAQ_CNTR2_RG* register (table 2-3), which can be used by host DSP software to indicate detected start synchronization event, will be set to the ‘1’ state immediately after *ADC_SYNC_DAQ_RUN* and *DAC_SYNC_DAQ_RUN* bits correspondingly of *DAQ_CNTR2_RG* register (table 2-3) is set to the ‘1’ state by host DSP software. ADC/DAC synchronous data acquisition process will be initialized by ADC data acquisition controller as soon as *ADC_SYNC_DAQ_SSYNC_OK* and *DAC_SYNC_DAQ_SSYNC_OK* bits correspondingly are set to the ‘1’ state. *Software defined start synchronization* is a typical selection for most applications and is selected as default on host PIOX-16 interface reset condition.

Active low external hardware defined start synchronization (fig.2-15b and fig.2-16b) must be selected in case ADC/DAC synchronous data acquisition processes shall be started synchronously with any external hardware defined event after host DSP software will set *ADC_SYNC_DAQ_RUN* and *DAC_SYNC_DAQ_RUN* bits correspondingly of *DAQ_CNTR2_RG* register (table 2-3). In this case, external hardware defined start synchronization event is detected as active low condition at *ADC_XSSYNC_IN* and *DAC_XSSYNC_IN* input pins correspondingly of on-board JP3 connector (refer to Appendix A and C for more details), and the read-only *ADC_SYNC_DAQ_SSYNC_OK* and *DAC_SYNC_DAQ_SSYNC_OK* bits correspondingly of *DAQ_CNTR2_RG* register (table 2-3), which can be used by host DSP software to indicate detected start synchronization event, will be set to the ‘1’ state immediately when external hardware defined start synchronization event will be detected at *ADC_XSSYNC_IN* and *DAC_XSSYNC_IN* input pins correspondingly of on-board JP3 connector after *ADC_SYNC_DAQ_RUN* and *DAC_SYNC_DAQ_RUN* bits correspondingly of *DAQ_CNTR2_RG* register (table 2-3) are set to the ‘1’ state by host DSP software. ADC/DAC synchronous data acquisition processes will be initialized by ADC/DAC data acquisition controller as soon as *ADC_SYNC_DAQ_SSYNC_OK* and *DAC_SYNC_DAQ_SSYNC_OK* bits correspondingly are set to the '1' state.

Falling edge triggered external hardware defined start synchronization (fig.2-15c and 2-16c) shall be selected in case ADC/DAC synchronous data acquisition process must be started synchronously with any external hardware defined event after host DSP software will set *ADC_SYNC_DAQ_RUN* and *DAC_SYNC_DAQ_RUN* bits correspondingly of *DAQ_CNTR2_RG* register (table 2-3). In this case, external hardware defined start synchronization event is detected as the falling edge condition at *ADC_XSSYNC_IN* and *DAC_XSSYNC_IN* input pins correspondingly of on-board JP3 connector (refer to Appendix A and C for more details), and the read-only *ADC_SYNC_DAQ_SSYNC_OK* and *DAC_SYNC_DAQ_SSYNC_OK* bits correspondingly of *DAQ_CNTR2_RG* register (table 2-3), which can be used by host DSP software to indicate detected start synchronization event, will be set to the '1' state immediately when external hardware defined start synchronization event will be detected at *ADC_XSSYNC_IN* and *DAC_XSSYNC_IN* input pins correspondingly of on-board JP3 connector after *ADC_SYNC_DAQ_RUN* and *DAC_SYNC_DAQ_RUN* bits correspondingly of *DAQ_CNTR2_RG* register (table 2-3) are set to the '1' state by host DSP software. ADC synchronous data acquisition process will be initialized by ADC data acquisition controller as soon as *ADC_SYNC_DAQ_SSYNC_OK* and *DAC_SYNC_DAQ_SSYNC_OK* bits correspondingly are set to the '1' state.

Note, that *ADC_SYNC_DAQ_SSYNC_OK* and *DAC_SYNC_DAQ_SSYNC_OK* bits of *DAQ_CNTR2_RG* register can be also used to generate host PIOX-16 interrupt requests (table 2-16).

Download of real-time ADC output data into host DSP environment for ADC-ASYNC-DAQ and ADC-SYNC-PX-DAQ data acquisition modes

Real-time ADC output data for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes, which both provide direct access to ADC output data via host PIOX-16 interface, can be downloaded to host DSP environment either by host DSP software or by host DSP on-chip DMA controller synchronized by ADC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

For both *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes, real-time ADC output data is updated on ADC output data ready condition inside either ADC sampling cycle for *ADC-ASYNC-DAQ* data acquisition mode or ADC scan cycle of ADC sampling packet for *ADC-SYNC-PX-DAQ* data acquisition mode. ADC output data ready condition is indicated via *ADC_RDY* bit of *DAQ_CNTR2_RG* register (table 2-3) and can be also used to generate host PIOX-16 interrupt request (table 2-16). Once updated, current ADC output data shall be read prior next ADC output data ready condition will occur, otherwise current ADC output data will be lost and overwritten by next ADC output data.

Note, that although both *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes provide convenient direct access to real-time ADC output data from host PIOX-16 interface, care must be taken to ensure that host DSP software and host DSP on-chip DMA controller (synchronized by ADC IRT controller) both shall provide enough performance and reserve sufficient DSP data bus traffic in order to download real-time ADC output data for all enabled ADC channels into host DSP environment without ADC data losses.

Note, that for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes, all four ADC channels perform synchronous A/D conversion regardless of the state of *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6), except for exceptional condition for *ADC-SYNC-PX-DAQ* data acquisition mode only when *ADC0_EN..ADC3_EN* bits are all set to the '0' state.

CAUTION

In case all *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6) are set to the '0' state, then ADC synchronous data acquisition process will not initialize.

Host DSP software can download ADC output data for any ADC channel by reading the corresponding *ADCx_DATA_RG* read-only register (*x*=0..3). In case host DSP on-chip DMA controller synchronized by ADC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM is used to download real-time ADC output data into host DSP environment, then *ADC_IRT_DATA_RG* register shall be specified as the source address without address increment/decrement feature for DMA controller.

CAUTION

In case host DSP on-chip controller synchronized by ADC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM is used to download real-time ADC output data into host DSP environment, then *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6) are used to define ADC channels, which real-time output data will be downloaded to host DSP environment.

Figure 2-17 provides timing diagram for real-time ADC output data read by host DSP software, which is applicable for both *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes.

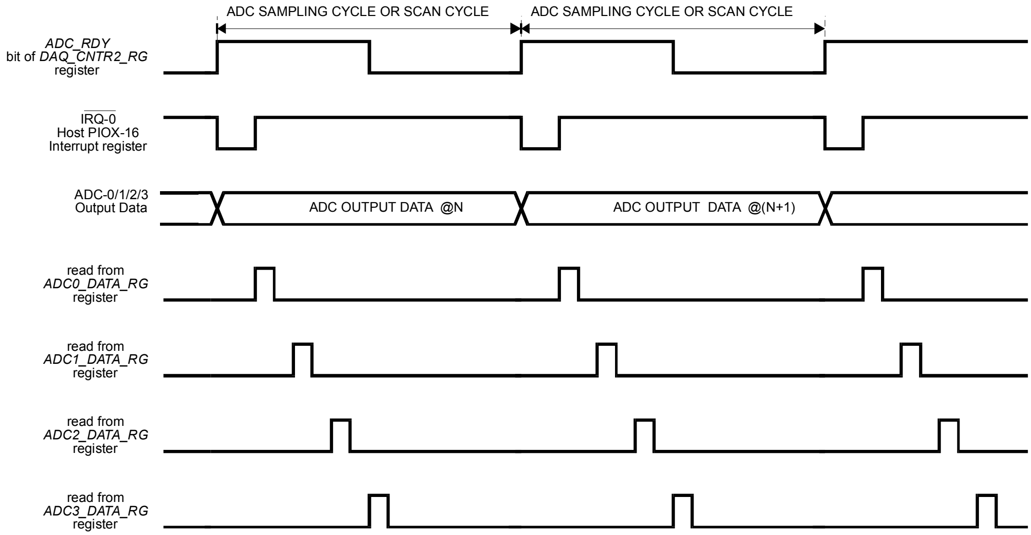
ADC SAMPLING CYCLE OR SCAN CYCLE
ADC SAMPLING CYCLE OR SCAN CYCLE
ADC_RDY bit of DAQ_CNTR2_RG register
IRQ-0 Host PIOX-16 Interrupt register
ADC-0/1/2/3 Output Data
ADC OUTPUT DATA @N
ADC OUTPUT DATA @(N+1)
read from ADC0_DATA_RG register
read from ADC1_DATA_RG register
read from ADC2_DATA_RG register
read from ADC3_DATA_RG register

Fig.2-17. Timing diagram for real-time ADC output data download by host DSP software for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes.

Read of real-time ADC output data via host DSP software is the most simple solution, however it may consume significant host DSP performance to execute. In case host DSP software is used to read real-time ADC output data for both *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes, then ADC output data ready condition can be detected by host DSP software either by polling *ADC_RDY* read-only bit of *DAQ_CNTR2_RG* register (table 2-3), or as a host PIOX-16 interrupt event in case any of host PIOX-16 interrupt request selectors is be configured to generate host PIOX-16 interrupt request on ADC output data ready condition (refer to table 2-12).

An alternative high-performance solution, which can be used to download real-time ADC output data into host DSP environment without consuming virtually any of DSP performance, is to utilize host DSP on-chip DMA controllers, which must be all synchronized by common ADC output data ready event via any of host PIOX-16 interrupt requests of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. The number of involved host DMA on-chip DMA controllers must correspond to the number of enabled ADC channels, i.e. each DMA controller must download real-time ADC output data from only one ADC channel. Thus, in case all ADC channels are enabled, then host DSP must provide at least four DSP on-chip DMA controllers in order to implement this ADC data transfer method.

CAUTION

Check with the DSP type of your host *TORNADO* DSP system/controller in order to ensure that it provides required number of on-chip DMA controllers. Recommended selection are *TORNADO* DSP systems/controllers with TMS320C6201/C6202/C6203/C6701 DSP, which provide four DSP on-chip DMA controllers, and *TORNADO* DSP systems/controllers with TMS320C6414/C6415/C6416 DSP, which provide up to 64 DSP on-chip EDMA controllers.

All involved host DSP on-chip DMA/EDMA controllers shall be configured with the source synchronization event set to any of IRQ-0..3 host PIOX-16 interrupt requests. The corresponding *HIRQx_SEL_RG* register (*x*=0..3, table 2-15) must be configured with the {*HIRQx_SEL*-[3:0]} bits to be set in accordance with table 2-16 to select ADC output data ready event as interrupt request source, host interrupt request enabled (bit *HIRQx_EN* must be set to the '1' state), and interrupt request polarity (bit *HIRQx_POL*) set in accordance with the type of host *TORNADO* DSP system/controller.

CAUTION

Check with your host *TORNADO* DSP system/controller user's guide in order to ensure that correct interrupt request polarity is set for involved host interrupt request input (either of IRQ-0..3) of host PIOX-16 interface in order to transfer source synchronization event for DMA controller of host DSP.

TORNADO-3x/P3x/E3x/P62/P6202/P6203/P67/P64xx/E6202/E6203/E64xx DSP systems and controllers require that host interrupt request(s) with positive polarity are used as the synchronization event(s) for DSP on-chip DMA controller(s).

TORNADO-62/67/E62/E67 DSP systems and controllers require that host interrupt request(s) with negative polarity are used as the synchronization event(s) for DSP on-chip DMA controller(s).

Also, involved host DSP on-chip DMA/EDMA controllers shall be configured with the source transfer address equal to the address of corresponding *ADCx_DATA_RG* register (table 2-1) without address increment/decrement feature in order to read real-time ADC output data of particular ADC channel. The destination transfer addresses of DMA controllers must meet requirements of host DSP application and point to any memory allocated arrays.

When using several host DSP on-chip DMA controllers in order to download real-time ADC output data for enabled ADC channels, all DMA controllers will receive common source synchronization event via selected host interrupt request on ADC ready condition, and will synchronously initialize data transfer cycle from particular *ADCx_DATA_RG* register to host DSP environment. Note, that with this ADC data transfer method, real-time ADC output data for each ADC channel will be downloaded to the dedicated memory array.

However, the most universal and recommended high-performance solution, which can be used to download real-time ADC output data into host DSP environment without consuming virtually any of DSP performance, is to utilize only one host DSP on-chip DMA controller, which must be synchronized by ADC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. For more details about how to download of real-time ADC output data using ADC IRT controller refer to section "ADC IRT and DAC IRT Controllers" later in this chapter.

Download of ADC FIFO output data into host DSP environment for ADC-SYNC-FIFO-OPM-DAQ and ADC-SYNC-FIFO-PTM-DAQ synchronous data acquisition modes

For *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes, real-time ADC output data is not directly available via host PIOX-16 interface. Instead, real-time ADC output data is read by ADC data acquisition controller in each ADC scan cycle and is transferred (pushed) into common high-density ADC FIFO. ADC FIFO contains packed real-time ADC output data for all enables ADC channels and can be read from host PIOX-16 interface.

ADC data acquisition controller transfers real-time ADC output data to ADC FIFO in each ADC scan cycle on ADC output data ready condition in accordance with the state of *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6). ADC data acquisition controller scans *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register starting from the *ADC0_EN* bit and detects all '1' bits until the last *ADC3_EN* bit will be processed:

ADC0_EN → *ADC1_EN* → *ADC2_EN* → *ADC3_EN*

Once *ADCx_EN* bit (*x*=0..3) is in the ‘1’ state, then the ADC output data for the corresponding ADC channels will be transferred into ADC FIFO.

CAUTION

In case all *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6) are set to the ‘0’ state, then ADC synchronous data acquisition process will not initialize.

Figure 2-18 provides timing diagram for operation of ADC data acquisition controller, which performs real-time ADC output data transfer into ADC FIFO for each ADC scan cycle for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes. Provided example corresponds to ADC-0, ADC-1, and ADC-3 channels enabled (*ADC0_EN*=1, *ADC1_EN*=1, *ADC2_EN*=0, *ADC3_EN*=1) and to the ADC sampling packet comprised of two ADC scan cycles.

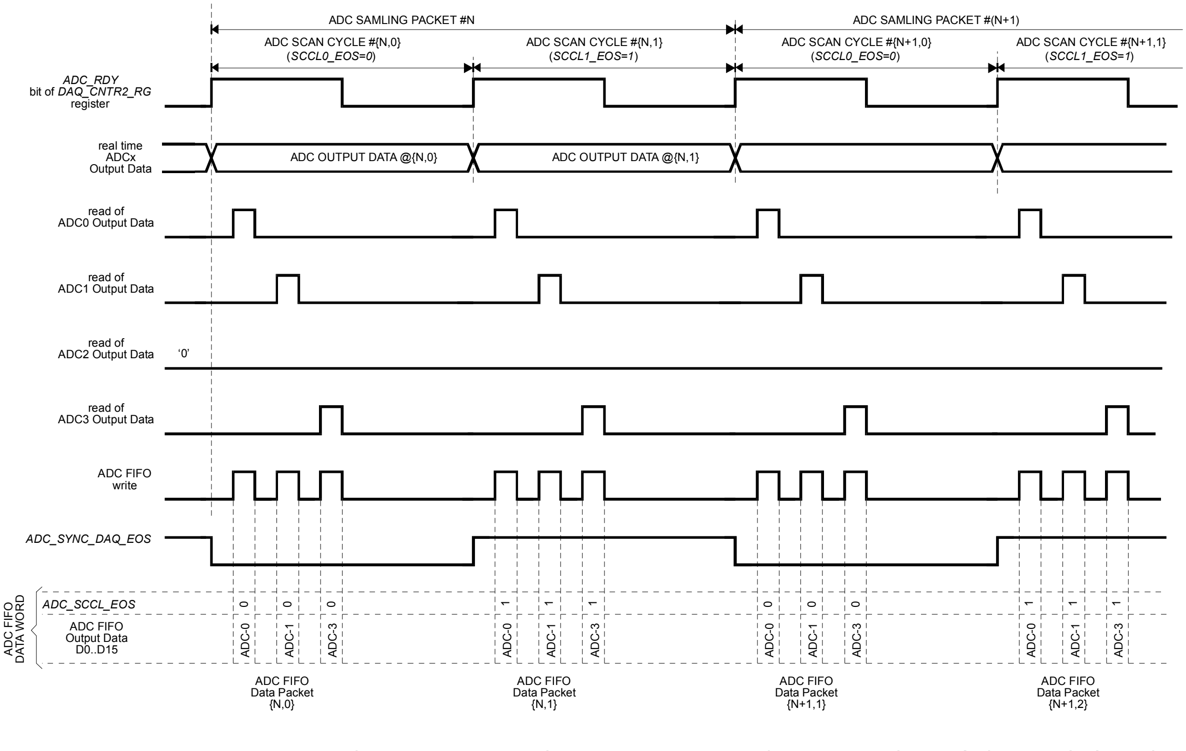
ADC SAMLING PACKET #N
ADC SAMLING PACKET #(N+1)
ADC SCAN CYCLE #{N,0} (SCCL0_EOS=0)
ADC SCAN CYCLE #{N,1} (SCCL1_EOS=1)
ADC SCAN CYCLE #{N+1,0} (SCCL0_EOS=0)
ADC SCAN CYCLE #{N+1,1} (SCCL1_EOS=1)
ADC_RDY bit of DAQ_CNTR2_RG register
real time ADCx Output Data
ADC OUTPUT DATA @{N,0}
ADC OUTPUT DATA @{N,1}
read of ADC0 Output Data
read of ADC1 Output Data
read of ADC2 Output Data
‘0’
read of ADC3 Output Data
ADC FIFO write
ADC_SYNC_DAQ_EOS
ADC FIFO DATA WORD
ADC_SCCL_EOS
ADC FIFO Output Data D0..D15
0 0 0 1 1 1 0 0 0 1 1 1
ADC-0 ADC-1 ADC-3
ADC FIFO Data Packet {N,0}
ADC FIFO Data Packet {N,1}
ADC FIFO Data Packet {N+1,1}
ADC FIFO Data Packet {N+1,2}

Fig.2-18. Timing diagram for real-time ADC output data transfer into ADC FIFO for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes.

Internal data of on-board 256Kx17 ADC FIFO is organized as a series of *ADC FIFO data packets*. Each *ADC FIFO data packet* corresponds to one ADC scan cycle and comprises of a set of 17-bit data words (fig.2-19), each corresponding to one enabled ADC channel. There is as many data words in one *ADC FIFO data packet*, as many ADC channels are enabled via *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register.

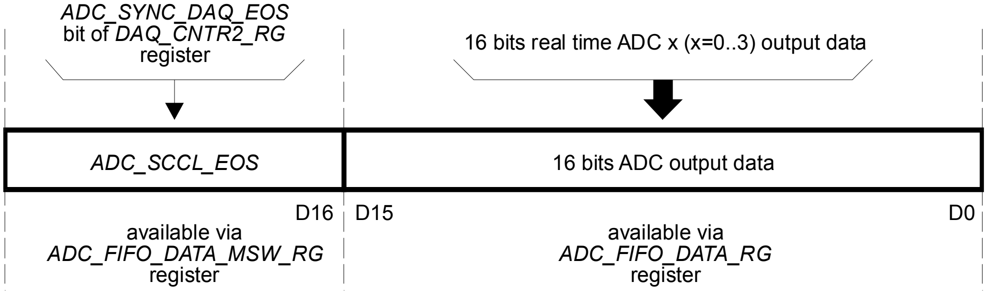

Fig.2-19. ADC FIFO data word format.

Each 17-bit ADC FIFO data word contains 16-bit ADC output data (ADC FIFO data bits #0..#15) for the corresponding ADC channel and associated *ADC_SCCL_EOS* end-of-scan indicator flag (ADC FIFO data bit #16), which appears as the copy of *ADC_SYNC_DAQ_EOS* bit of *DAC_SYNC_RG* register (table 2-5). Real-time ADC output data for all enabled ADC channels, which are acquired in the last ADC scan cycle of each ADC sampling packet will have the associated *ADC_SCCL_EOS* end-of-scan indicator flag set to the ‘1’ state.

CAUTION

ADC FIFO data words do not contain information about ADC channel number, which has to be reconstructed by host DSP software when unpacking *ADC FIFO data packets*.

On host PIOX-16 interface side of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, ADC FIFO output data words can be read via two 16-bit read-only data registers: *ADC_FIFO_DATA_RG* read-only register and *ADC_FIFO_DATA_MSW_RG* read-only register.

ADC_FIFO_DATA_RG read-only register (refer to table 2-1) contains 16-bit ADC output data (most significant bit is the sign bit) and can be read by either host DSP software or host DSP on-chip DMA controller in order to extract *ADC FIFO data packets* from ADC FIFO.

CAUTION

When reading from *ADC_FIFO_DATA_RG* read-only register, full 17-bit ADC FIFO data word is being read with the *ADC_SCCL_EOS* end-of-scan flag (bit #16) being latched into optional *ADC_FIFO_DATA_MSW_RG* read-only register (table 2-11), which can be further read if required by host DSP software.

> **CAUTION**
>
> Each read from *ADC_FIFO_DATA_RG* read-only register advances ADC FIFO read pointer, so ADC FIFO output data cannot be re-read.

Optional *ADC_FIFO_DATA_MSW_RG* read-only register must be used by host DSP software in order to correctly unpack *ADC FIFO data packets* and reconstruct real-time ADC output data arrays inside host DSP environment for each enabled ADC channel during *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes. *ADC_FIFO_DATA_MSW_RG* read-only register must be read only after *ADC_FIFO_DATA_RG* register has been read, and contains *ADC_SCCL_EOS* flag indicator (table 2-11), which is associated with the real-time ADC output data sample, which has been previously read from *ADC_FIFO_DATA_RG* read-only register. In case *ADC_SCCL_EOS* of *ADC_FIFO_DATA_MSW_RG* read-only register appears as logical '1', then previously read real-time ADC output data sample from *ADC_FIFO_DATA_RG* read-only register has been acquired in the last ADC scan cycle of ADC sampling packet.

Temporary buffering of real-time ADC output data in high-density ADC FIFO allows host DSP software and/or host DSP on-chip DMA controller not to concern about download of real-time ADC output data in each ADC scan cycle, and, therefore, it allows to significantly offload host DSP data bus traffic structure and perform read of ADC FIFO data later as soon as it will be required by data processing algorithm. Nevertheless, for *ADC-SYNC-FIFO-PTM-DAQ* data acquisition mode, which provides continuous analog input signal acquisition and never terminates itself normally, either host DSP software or host DSP on-chip DMA controller synchronized by ADC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, both shall provide enough performance and reserve enough DSP data bus traffic in order to download real-time ADC FIFO output data for all enabled ADC channels into host DSP environment without ADC FIFO overflow condition.

> **CAUTION**
>
> For *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode, it is not recommended to read ADC FIFO until ADC synchronous data acquisition process will terminate itself normally on ADC FIFO termination flag condition, since otherwise it may result in modification of ADC FIFO read pointer on each read from ADC FIFO and termination of ADC synchronous data acquisition process on incorrect number of acquired and stored ADC sampling packets.

Figures 2-20a and 2-20b provides timing diagrams for typical ADC FIFO data read procurement during *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes. Note, that ADC FIFO output data reads for *ADC-SYNC-FIFO-PTM-DAQ* data acquisition mode typically appear as periodical ADC FIFO data read packet bursts during continuous ADC synchronous data acquisition process typically initialized on software configured ADC FIFO PAF (FIFO partially full) flag condition, whereas for *ADC-SYNC-FIFO-OTM-DAQ* data acquisition mode, ADC FIFO output data reads shall occur after ADC synchronous data acquisition process will normally terminate.

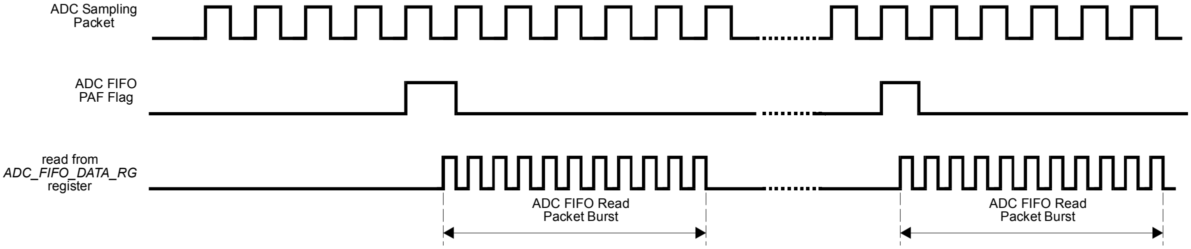
ADC Sampling Packet
ADC FIFO PAF Flag
read from *ADC_FIFO_DATA_RG* register
ADC FIFO Read Packet Burst
ADC FIFO Read Packet Burst

Fig.2-20a. ADC FIFO output data read procurement for *ADC-SYNC-FIFO-PTM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition mode.

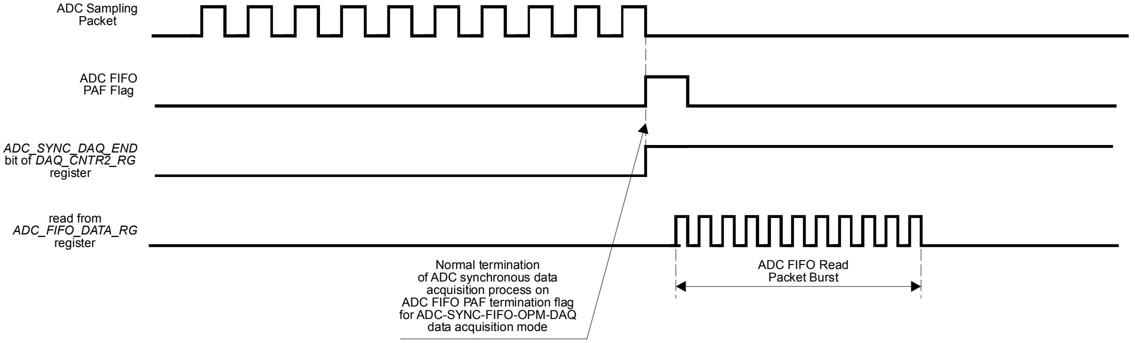
ADC Sampling Packet
ADC FIFO PAF Flag
ADC_SYNC_DAQ_END bit of *DAQ_CNTR2_RG* register
read from *ADC_FIFO_DATA_RG* register
Normal termination of ADC synchronous data acquisition process on ADC FIFO PAF termination flag for ADC-SYNC-FIFO-OPM-DAQ data acquisition mode
ADC FIFO Read Packet Burst

Fig.2-20b. ADC FIFO output data read procurement for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition mode.

It is important to note, that ADC FIFO output data is also available for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes after normal abort procedure for ADC synchronous data acquisition process via setting *ADC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register. Normal abort procedure for ADC synchronous data acquisition process will just stop ADC FIFO data acquisition process and freeze last value of ADC FIFO write pointer. Stored ADC FIFO data will remain available for normal read via host PIOX-16 interface. After aborting, ADC synchronous data acquisition process can be restarted by setting *ADC_SYNC_DAQ_RUN* bit of *DAQ_CNTR2_RG* register to the ‘1’ state and detection of new start synchronization event.

CAUTION

Reset of ADC data acquisition controller during *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* ADC synchronous data acquisition modes by write to either *ADC_DAQ_RESET_RG* or *ADDA_DAQ_RESET_RG* registers also performs reset of ADC FIFO read and write pointers, thus making already stored ADC FIFO data unavailable for read via host PIOX-16 interface.

While ADC synchronous data acquisition process is running in *ADC-SYNC-FIFO-OTM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition mode, ADC data acquisition controller always checks for *ADC underflow condition*, while host PIOX-16 interface controller always checks for *ADC FIFO overflow condition*. For more details, refer to subsection "Errors processing" later in this section.

CAUTION

Neither of *ADC FIFO overflow and underflow error conditions* do not terminate ADC synchronous data acquisition process for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes.

The most high-performance solution to download real-time ADC output data into host DSP environment for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes is to use host DSP on-chip DMA controller, which must be synchronized by ADC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. For more details about how to download real-time ADC output data using ADC IRT controller refer to section "ADC IRT and DAC IRT Controllers" later in this chapter.

Upload of real-time DAC input data from host DSP environment for DAC-ASYNC-DAQ and DAC-SYNC-PX-DAQ data acquisition modes

Real-time DAC input data for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes, which both offer direct access to DAC input data via host PIOX-16 interface, can be uploaded from host DSP environment either by host DSP software or by host DSP on-chip DMA controller synchronized by DAC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

For both *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes, host DSP software can upload real-time DAC input data for any DAC channel by writing to the corresponding *DACx_DATA_RG* write-only register (*x*=0..3). In case host DSP on-chip DMA controller synchronized by DAC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM is used to upload real-time DAC input data from host DSP environment, then *DAC_IRT_DATA_RG* register shall be specified as the destination address without address increment/decrement feature for DMA controller.

CAUTION

In case host DSP on-chip controller synchronized by DAC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM is used to upload real-time DAC input data from host DSP environment, then *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-6) are used to define DAC channels, which real-time input data will be uploaded from host DSP environment.

Note, that for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes, all four DAC channels perform synchronous update of DAC output signals regardless of the state of *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-6), except for exceptional condition for *DAC-SYNC-PX-DAQ* data acquisition mode only when *DAC0_EN..DAC3_EN* bits are all set to the ‘0’ state.

CAUTION

In case all *DAC0_EN..ADC3_EN* bits of *DAC_CNF_RG* register (table 2-6) are set to the ‘0’ state, then DAC synchronous data acquisition process for *DAC-SYNC-PX-DAQ* data acquisition mode will not initialize.

For *DAC-ASYNC-DAQ* data acquisition mode with *transparent DAC output data load mode* (fig.2-11a), DAC output signal is updated immediately after DAC input data has been written, so there is no any time restrictions and time frame for upload of real-time DAC input from host DSP environment.

For *DAC-ASYNC-DAQ* data acquisition mode with *DAC output data load on software defined event mode* (fig.2-11b), DAC output signals are updated on DAC sampling cycle, which is generated when host DSP write to either *DAC_OUTPUT_LD_RG* or *ADC_START_DAC_OUTPUT_LD_RG* registers. Since host DSP software also performs write of DAC input data, then there is no any time restrictions and time frame for upload of real-time DAC input from host DSP environment.

For *DAC-ASYNC-DAQ* data acquisition mode with *DAC output data load on the end of DAC IRT data transfer mode* (fig.2-11c), DAC output signals are updated immediately on the end of DAC IRT data transfer cycle, so there is no any time restrictions and time frame for upload of real-time DAC input from host DSP environment.

Instead, for *DAC-ASYNC-DAQ* data acquisition mode with selected *DAC output data load on hardware defined sampling frequency event mode* (fig.2-11c) and for *DAC-SYNC-PX-DAQ* data acquisition mode, DAC output signal is updated during DAC sampling cycle, which is generated on hardware defined sampling frequency event, so real-time DAC input data shall be uploaded prior next DAC sampling cycle will occur, otherwise DAC output signals will be not updated and will correspond to the last written DAC input data.

CAUTION

In case host DSP will write DAC input data during DAC sampling cycle is active, then DAC output signal value is not guaranteed.

Figure 2-21 provides timing diagram for real-time DAC input data write by host DSP software, which is applicable for *DAC-ASYNC-DAQ* data acquisition mode with selected *DAC output data load on hardware defined sampling frequency event mode* and for *DAC-SYNC-PX-DAQ* data acquisition mode.

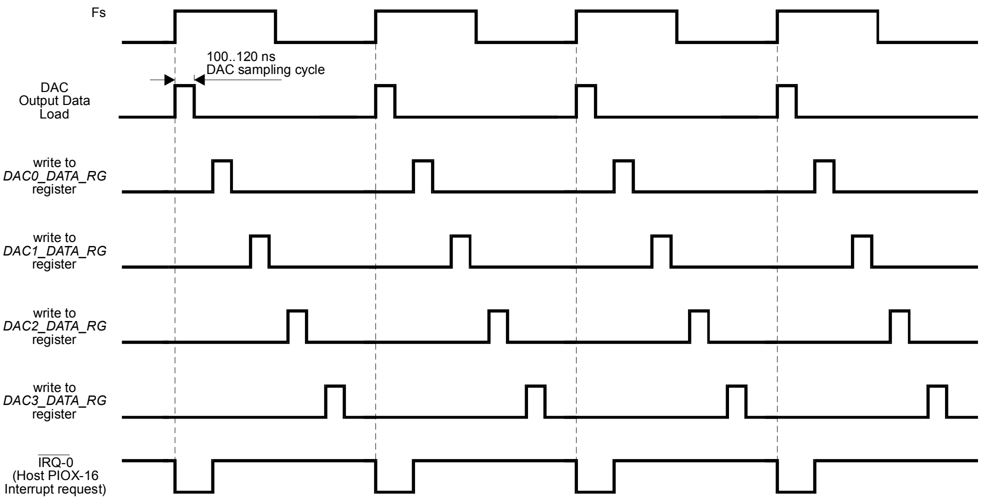

Fig.2-21. Timing diagram for real-time DAC input data upload by host DSP software for *DAC-ASYNC-DAQ* data acquisition mode with DAC output data load on hardware defined sampling frequency event mode and for *DAC-SYNC-PX-DAQ* data acquisition mode.

Note, that although both *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes provide convenient direct access to real-time DAC input data from host PIOX-16 interface, care must be taken to ensure that host DSP software and host DSP on-chip DMA controller (synchronized by DAC IRT controller) both shall provide enough performance and reserve sufficient DSP data bus traffic in order to upload real-time DAC input data for all enabled DAC channels from host DSP environment without DAC data miss.

Write of real-time DAC input data via host DSP software is the most simple solution, however it may consume significant host DSP performance to execute. In case host DSP software is used to write real-time DAC input data for both *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes, then time event for write of DAC input data can be detected by host DSP software, for example as a host PIOX-16 interrupt configured to generate on sampling frequency event (refer to table 2-12).

An alternative high-performance solution, which can be used to upload real-time DAC input data from host DSP environment without consuming virtually any of DSP performance, is to utilize host DSP on-chip DMA controllers, which must be all synchronized by common hardware defined sampling frequency event via any of host PIOX-16 interrupt requests of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. The number of involved host DMA on-chip DMA controllers must correspond to the number of enabled DAC channels, i.e. each DMA controller must upload real-time DAC output data for only one DAC channel. Thus, in case all DAC channels are enabled, then host DSP must provide at least four DSP on-chip DMA controllers in order to implement this DAC data transfer method.

> **CAUTION**
>
> Check with the DSP type of your host *TORNADO* DSP system/controller in order to ensure that it provides required number of on-chip DMA controllers. Recommended selection are *TORNADO* DSP systems/controllers with TMS320C6201/C6202/C6203/C6701 DSP, which provide four DSP on-chip DMA controllers, and *TORNADO* DSP systems/controllers with TMS320C6414/C6415/C6416 DSP, which provide up to 64 DSP on-chip EDMA controllers.

All involved host DSP on-chip DMA/EDMA controllers shall be configured with the destination synchronization event set to any of IRQ-0..3 host PIOX-16 interrupt requests. The corresponding *HIRQx_SEL_RG* register (*x*=0..3, table 2-15) must be configured with the {*HIRQx_SEL*-[3:0]} bits to be set in accordance with table 2-16 to select sampling frequency event as interrupt request source, host interrupt request enabled (bit *HIRQx_EN* must be set to the ‘1’ state), and interrupt request polarity (bit *HIRQx_POL*) set in accordance with the type of host *TORNADO* DSP system/controller.

> **CAUTION**
>
> Check with your host *TORNADO* DSP system/controller user’s guide in order to ensure that correct interrupt request polarity is set for involved host interrupt request input (either of IRQ-0..3) of host PIOX-16 interface in order to transfer source synchronization event for DMA controller of host DSP.
>
> *TORNADO-3x/P3x/E3x/P62/P6202/P6203/P67/P64xx/E6202/E6203/E64xx* DSP systems and controllers require that host interrupt request(s) with positive polarity are used as the synchronization event(s) for DSP on-chip DMA controller(s).
>
> *TORNADO-62/67/E62/E67* DSP systems and controllers require that host interrupt request(s) with negative polarity are used as the synchronization event(s) for DSP on-chip DMA controller(s).

Also, involved host DSP on-chip DMA/EDMA controllers shall be configured with the destination transfer address equal to the address of the corresponding *DACx_DATA_RG* register (table 2-1) without address increment/decrement feature in order to write real-time DAC input data for particular ADC channel. The source transfer addresses of DMA controllers must meet requirements of host DSP application and point to any memory allocated arrays.

When using several host DSP on-chip DMA controllers in order to upload real-time DAC input data for enabled DAC channels, all DMA controllers will receive common destination synchronization event via selected host interrupt request on hardware defined sampling frequency condition, and will synchronously initialize data transfer cycle from host DSP environment to particular *DACx_DATA_RG* register. Note, that with this DAC data transfer method, real-time DAC input data for each DAC channel will be uploaded from the dedicated memory array.

The most universal and recommended high-performance solution, which can be used to upload real-time DAC input data from host DSP environment without consuming virtually any of DSP performance, is to utilize only one host DSP on-chip DMA controller, which must be synchronized by DAC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. For more details about how to upload real-time DAC input data using DAC IRT controller refer to section "ADC IRT and DAC IRT Controllers" later in this chapter.

Upload of DAC FIFO input data from host DSP environment for DAC-SYNC-FIFO-OPM-DAQ and DAC-SYNC-FIFO-PTM-DAQ synchronous data acquisition modes

For *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes, real-time DAC input data is not directly available via host PIOX-16 interface. Instead, real-time DAC input data is extracted from common high-density DAC FIFO and is written to DAC input data registers by DAC data acquisition controller in each sampling period. DAC FIFO contains packed real-time DAC input data for all enables DAC channels and can be written to DAC FIFO by host PIOX-16 interface.

DAC data acquisition controller transfers real-time DAC input data from DAC FIFO in each sampling period immediately after DAC sampling cycle in accordance with the state of *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-6). DAC data acquisition controller scans *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register starting from the *DAC0_EN* bit and detects all '1' bits until the last *DAC3_EN* bit will be processed:

DAC0_EN → *DAC1_EN* → *DAC2_EN* → *DAC3_EN*

Once *DACx_EN* bit (*x*=0..3) is in the '1' state, then the DAC input data for the corresponding DAC channels will be transferred from DAC FIFO.

CAUTION

In case all *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-6) are set to the '0' state, then DAC synchronous data acquisition process will not initialize.

Figure 2-22 provides timing diagram for operation of DAC data acquisition controller, which performs real-time DAC input data transfer from DAC FIFO during each sampling period for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes. Provided example corresponds to DAC-0, DAC-1, and DAC-3 channels enabled (*DAC0_EN*=1, *DAC1_EN*=1, *DAC2_EN*=0, *DAC3_EN*=1).

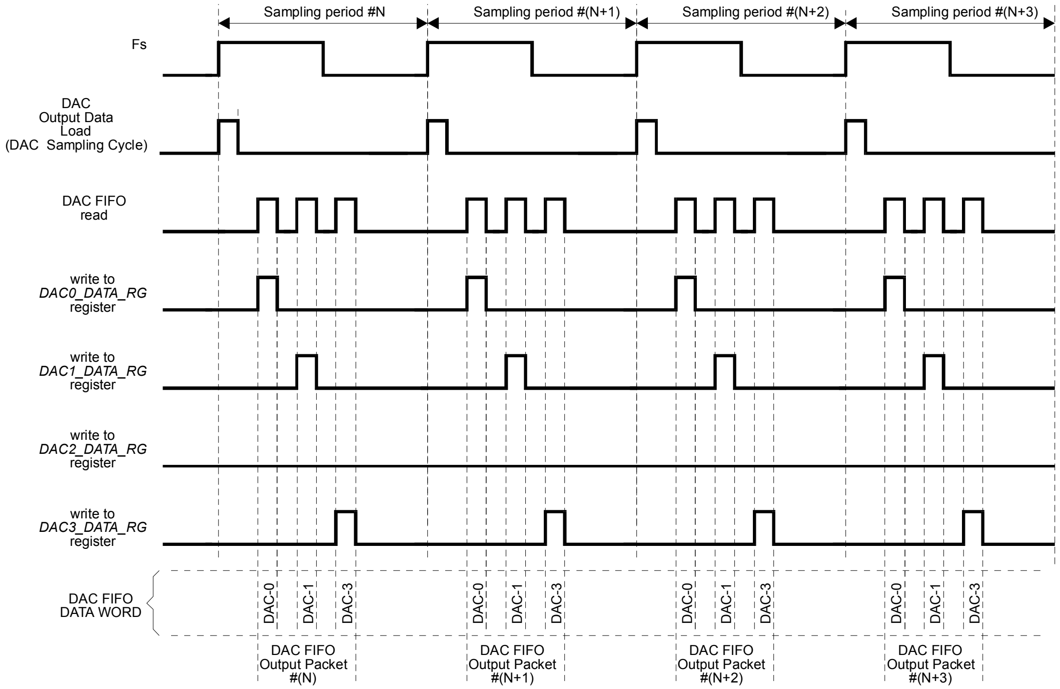
Sampling period #N
Sampling period #(N+1)
Sampling period #(N+2)
Sampling period #(N+3)
Fs
DAC Output Data Load (DAC Sampling Cycle)
DAC FIFO read
write to DAC0_DATA_RG register
write to DAC1_DATA_RG register
write to DAC2_DATA_RG register
write to DAC3_DATA_RG register
DAC FIFO DATA WORD
DAC-0
DAC-1
DAC-3
DAC FIFO Output Packet #(N)
DAC FIFO Output Packet #(N+1)
DAC FIFO Output Packet #(N+2)
DAC FIFO Output Packet #(N+3)

Fig.2-22. Timing diagram for real-time DAC input data transfer from DAC FIFO for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes.

Internal data of on-board 256Kx16 DAC FIFO is organized as a series of *DAC FIFO data packets*. Each *DAC FIFO data packet* corresponds to one sampling period and comprises of a set of 16-bit data words (most significant bit is the sign bit), each corresponding to the DAC input data for one enabled DAC channel. There is as many data words in one *DAC FIFO data packet*, as many DAC channels are enabled via *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register.

CAUTION

DAC FIFO data packets do not contain information about DAC channel numbers corresponding to DAC input data words.

Host DSP software must correctly pack DAC input data for all enabled DAC channels into *DAC FIFO data packets*.

Once started, synchronous DAC FIFO data acquisition process will extract DAC input data words from DAC FIFO starting from the DAC input data words for the first enabled DAC channel of the first *DAC FIFO data packet*.

On host PIOX-16 interface side of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, DAC FIFO input data words can be written via *DAC_FIFO_DATA_RG* read-only 16-bit write-only data register (most significant bit is the sign bit) (refer to table 2-1), which can be written by either host DSP software or host DSP on-chip DMA controller in order to fill-in *DAC FIFO data packets* to DAC FIFO.

CAUTION

Each write to *DAC_FIFO_DATA_RG* write-only register advances DAC FIFO write pointer, so DAC FIFO input data cannot be re-written.

Temporary buffering of real-time DAC input data in high-density DAC FIFO allows host DSP software and/or host DSP on-chip DMA controller not to concern about upload of real-time DAC output data in each sampling period, and, therefore, it allows to significantly offload host DSP data bus traffic structure and perform write of DAC FIFO input data as soon as it will be required by data processing algorithm. Nevertheless, for *DAC-SYNC-FIFO-PTM-DAQ* data acquisition mode, which provides continuous analog output signal acquisition and never terminates itself normally, either host DSP software or host DSP on-chip DMA controller synchronized by DAC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, both shall provide enough performance and reserve enough DSP data bus traffic in order to upload real-time DAC FIFO input data for all enabled DAC channels from host DSP environment without DAC FIFO underflow condition.

CAUTION

For *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode, it is not recommended to write DAC FIFO input data after DAC synchronous data acquisition process starts and until it terminates normally on DAC FIFO termination flag condition, since otherwise it may result in modification of DAC FIFO write pointer on each write to DAC FIFO and termination of DAC synchronous data acquisition process on incorrect number of DAC sampling packets.

Figures 2-23a and 2-23b provides timing diagrams for typical DAC FIFO data write procurement during *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes. Note, that DAC FIFO input data writes for *DAC-SYNC-FIFO-PTM-DAQ* data acquisition mode typically appear as periodical DAC

FIFO data write packet bursts during continuous DAC synchronous data acquisition process typically initialized on software configured DAC FIFO PAE (FIFO partially empty) flag condition, whereas for *DAC-SYNC-FIFO-OTM-DAQ* data acquisition mode, DAC FIFO input data writes shall occur before DAC synchronous data acquisition process will start. Note, that for both *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes, host DSP must fill-in at least a fraction of DAC FIFO prior initializing DAC synchronous FIFO data acquisition process, otherwise DAC FIFO underflow condition may occur and *DAC FIFO data packet* synchronization may be lost.

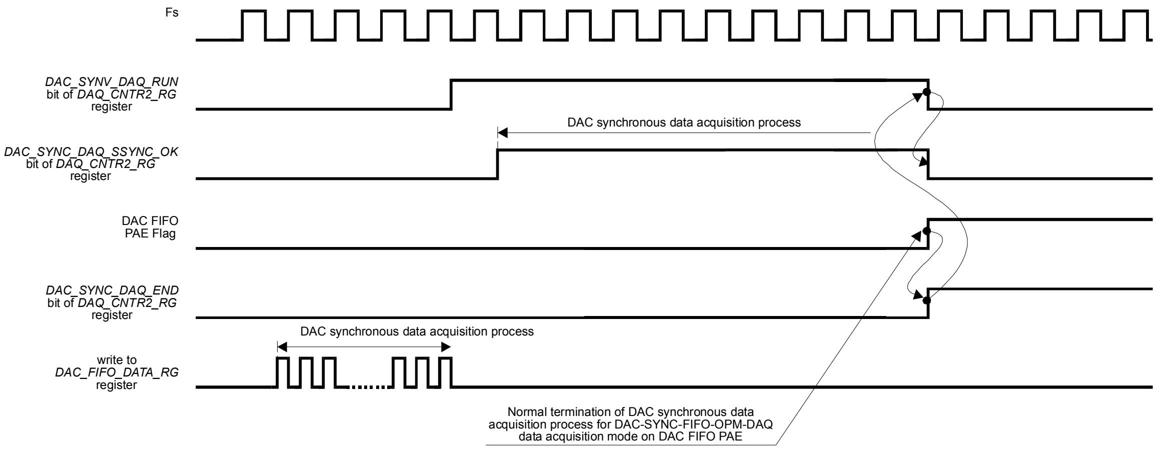
Fs
DAC_SYNV_DAQ_RUN bit of *DAQ_CNTR2_RG* register
DAC synchronous data acquisition process
DAC_SYNC_DAQ_SSYNC_OK bit of *DAQ_CNTR2_RG* register
DAC FIFO PAE Flag
DAC_SYNC_DAQ_END bit of *DAQ_CNTR2_RG* register
DAC synchronous data acquisition process
write to *DAC_FIFO_DATA_RG* register
Normal termination of DAC synchronous data acquisition process for DAC-SYNC-FIFO-OPM-DAQ data acquisition mode on DAC FIFO PAE

Fig.2-23a. DAC FIFO input data write procurement for *DAC-SYNC-FIFO-PTM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition mode.

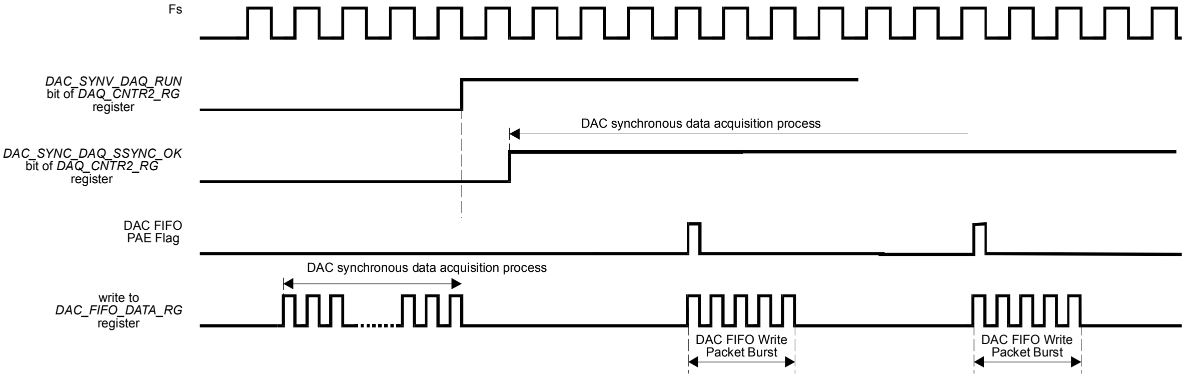
Fs
DAC_SYNV_DAQ_RUN bit of *DAQ_CNTR2_RG* register
DAC synchronous data acquisition process
DAC_SYNC_DAQ_SSYNC_OK bit of *DAQ_CNTR2_RG* register
DAC FIFO PAE Flag
DAC synchronous data acquisition process
write to *DAC_FIFO_DATA_RG* register
DAC FIFO Write Packet Burst
DAC FIFO Write Packet Burst

Fig.2-23b. DAC FIFO input data write procurement for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition mode.

It is important to note, that unpulled (unused) DAC FIFO data is still available for further output to DAC input data registers during *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes after normal abort procedure for DAC synchronous data acquisition process via setting *DAC_SYNC_DAQ_ABORT* bit of *DAQ_CNTR2_RG* register. Normal abort procedure for DAC synchronous data acquisition process will just stop DAC FIFO data acquisition process and freeze last value of

DAC FIFO read pointer. After aborting, DAC synchronous data acquisition process can be restarted by setting *DAC_SYNC_DAQ_RUN* bit of *DAQ_CNTR2_RG* register to the '1' state and detection of new start synchronization event.

> **CAUTION**
>
> Reset of DAC data acquisition controller during *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* DAC synchronous data acquisition modes by write to either *DAC_DAQ_RESET_RG* or *ADDA_DAQ_RESET_RG* registers also performs reset of DAC FIFO read and write pointers, thus making already stored DAC FIFO data unavailable for further read by DAC data acquisition controller.

While DAC synchronous data acquisition process is running in *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition mode, DAC data acquisition controller always checks for *DAC underflow condition*, while host PIOX-16 interface controller always checks for *DAC FIFO overflow condition*. For more details, refer to subsection "Errors processing" later in this section.

> **CAUTION**
>
> Neither of *DAC FIFO overflow and underflow error conditions* do not terminate DAC synchronous data acquisition process for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes.

The most high-performance solution to upload real-time DAC input data from host DSP environment for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes is to use host DSP on-chip DMA controller, which must be synchronized by DAC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. For more details about how to upload real-time DAC input data using DAC IRT controller refer to section "ADC IRT and DAC IRT Controllers" later in this chapter.

Errors processing

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM provides real-time data acquisition errors control in order host DSP software can monitor correct progress of ADC and DAC data acquisition processes as well as of correct operation of ADC and DAC IRT controllers. Data acquisition errors are available for polling via *ERR_STAT_RG* register (table 2-10) and can generate host PIOX-16 interrupt requests (tables 2-17 and 2-17).

The following data acquisition error conditions are being tracked by ADC and DAC data acquisition controllers during corresponding ADC and DAC data acquisition processes:

- ADC FIFO overflow, which appears as *ADC_FIFO_OVF* bit in *ERR_STAT_RG* register (*ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes only)
- ADC FIFO underflow, which appears as *ADC_FIFO_UVF* bit in *ERR_STAT_RG* register (*ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes only)
- DAC FIFO overflow, which appears as *DAC_FIFO_OVF* bit in *ERR_STAT_RG* register (*DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes only)

- DAC FIFO underflow, which appears as *DAC_FIFO_UVF* bit in *ERR_STAT_RG* register (*DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes only)
- ADC IRT error, which appears as *ADC_IRT_ERR* bit in *ERR_STAT_RG* register
- DAC IRT error, which appears as *DAC_IRT_ERR* bit in *ERR_STAT_RG* register.

While running in *ADC-SYNC-FIFO-PTM-DAQ* and *ADC-SYNC-FIFO-OPM-DAQ* synchronous data acquisition modes, ADC data acquisition controller always check for the *ADC overflow condition* in order to ensure that ADC FIFO has enough room to store real-time ADC outputs data. *ADC overflow condition* is indicated via *ADC_FIFO_OVF* bit of *ERR_STAT_RG* register (table 2-10) and is detected in case ADC data acquisition controller tries to write real-time ADC output data into ADC FIFO while ADC FIFO FF (FIFO full) flag has been already set to the ‘1’ state. ADC FIFO overflow error occurs in case input data stream into ADC FIFO, which is generated by ADC data acquisition controller, is not balanced with the ADC FIFO output data stream, which is controlled either by host DSP software or host DSP on-chip DMA controller synchronized by ADC IRT controller.

ADC overflow condition will never occur in *ADC-SYNC-FIFO-OPM-DAQ* synchronous data acquisition mode, since ADC data acquisition controller will terminate running ADC synchronous data acquisition process on either ADC FIFO PAF or ADC FIFO FF flag condition depending upon the state of *ADC_SYNC_FIFO_DAQ_TF_SEL* bit of *DAQ_SYNC_RG* register (table 2-4) as it has been described in the corresponding subsection above.

On the other side, host PIOX-16 interface controller always checks for *ADC FIFO underflow condition* when reading ADC FIFO output data from *ADC_FIFO_DATA_RG* read-only register during both *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes in order to ensure that ADC FIFO contains valid data to read. *ADC FIFO underflow condition* is indicated via *ADC_FIFO_UNF* bit of *ERR_STAT_RG* register (table 2-10) and is detected in case host DSP software reads from *ADC_FIFO_DATA_RG* read-only register while ADC FIFO EF (FIFO empty) flag has been already set to the ‘1’ state.

CAUTION

For *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes, host DSP software must check for the state of ADC FIFO EF flag via *ADC_FIFO_EF* bit of *FIFO_STAT_RG* register (table 2-9) in order to ensure that ADC FIFO contains valid data to read.

ADC underflow condition will never occur during *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes in case host DSP on-chip DMA controller synchronized by ADC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM is used to download ADC FIFO output data into host DSP environment, since ADC IRT controller will terminate data transfer on either ADC FIFO PAE or EF flag condition in accordance with the setting of *ADC_SYNC_FIFO_IRT_TF_SEL* bit of *DAQ_SYNC_RG* register (table 2-4).

CAUTION

Once set, ADC FIFO overflow error flag (*ADC_FIFO_OVF* bit of *ERR_STAT_RG* register) and ADC FIFO underflow error flag (*ADC_FIFO_UNF* bit of *ERR_STAT_RG* register) will stay in the '1' state until reset either via write of logical '1' to the corresponding bit of *ERR_CLR_RG* register (table 2-10), or by reset of ADC data acquisition controller by writing to either *ADC_DAQ_RESET_RG* or *ADDA_DAQ_RESET_RG* registers.

CAUTION

Neither of *ADC FIFO overflow and underflow error conditions* do not terminate ADC synchronous data acquisition process for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes.

While running in *DAC-SYNC-FIFO-PTM-DAQ* and *DAC-SYNC-FIFO-OPM-DAQ* synchronous data acquisition modes, DAC data acquisition controller always checks for *DAC underflow condition* in order to ensure that DAC FIFO still has real-time DAC input data. *DAC underflow condition* is indicated via *DAC_FIFO_UVF* bit of *ERR_STAT_RG* register (table 2-10) and is detected in case DAC data acquisition controller tries to read real-time DAC input data from DAC FIFO while DAC FIFO EF (FIFO empty) flag has been already set to the '1' state. DAC FIFO underflow error occurs in case input data stream into DAC FIFO, which is generated either by host DSP software or host DSP on-chip DMA controller synchronized by DAC IRT controller, is not balanced with the DAC FIFO output data stream, which is controlled by DAC data acquisition controller.

DAC underflow condition will never occur in *DAC-SYNC-FIFO-OPM-DAQ* synchronous data acquisition mode, since DAC data acquisition controller will terminate running DAC synchronous data acquisition process on either DAC FIFO PAE or DAC FIFO EF flag condition depending upon the state of *DAC_SYNC_FIFO_DAQ_TF_SEL* bit of *DAQ_SYNC_RG* register (table 2-4) as it has been described in the corresponding subsection above.

On the other side, host PIOX-16 interface controller always checks for *DAC FIFO overflow condition* when writing DAC FIFO input data to *DAC_FIFO_DATA_RG* write-only register during both *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes in order to ensure that DAC FIFO contains enough room for data to write. *DAC FIFO overflow condition* is indicated via *DAC_FIFO_OVF* bit of *ERR_STAT_RG* register (table 2-10) and is detected in case host DSP software writes to *DAC_FIFO_DATA_RG* write-only register while DAC FIFO FF (FIFO full) flag has been already set to the '1' state.

CAUTION

For *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes, host DSP software must check for the state of DAC FIFO FF flag via *DAC_FIFO_FF* bit of *FIFO_STAT_RG* register (table 2-9) in order to ensure that DAC FIFO has enough room for data to write.

DAC overflow condition will never occur during *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes in case host DSP on-chip DMA controller synchronized by DAC IRT controller of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM is used to upload DAC FIFO input data from host DSP environment, since DAC IRT controller will terminate data transfer on either DAC FIFO PAF or FF flag condition in accordance with the setting of *DAC_SYNC_FIFO_IRT_TF_SEL* bit of *DAQ_SYNC_RG* register (table 2-4).

CAUTION

Once set, DAC FIFO overflow error flag (*DAC_FIFO_OVF* bit of *ERR_STAT_RG* register) and DAC FIFO underflow error flag (*DAC_FIFO_UNF* bit of *ERR_STAT_RG* register) will stay in the ‘1’ state until reset either via write of logical ‘1’ to the corresponding bit of *ERR_CLR_RG* register (table 2-10), or by reset of DAC data acquisition controller by writing to either *DAC_DAQ_RESET_RG* or *ADDA_DAQ_RESET_RG* registers.

CAUTION

Neither of *DAC FIFO overflow and underflow error conditions* do not terminate DAC synchronous data acquisition process for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* synchronous data acquisition modes.

ADC IRT error (*ADC_IRT_ERR* bit in *ERR_STAT_RG* register) and DAC IRT error (*DAC_IRT_ERR* bit in *ERR_STAT_RG* register) are set in case of unbalanced traffic over host DSP data bus, which must be sufficient to match sampling frequency value. Thus, ADC IRT error is set in case host DSP on-chip DMA controller synchronized by ADC IRT controller was not able to read ADC output data for all enabled ADC channels prior next ADC data comes ready for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes. Correspondingly, DAC IRT error is set in case host DSP on-chip DMA controller synchronized by DAC IRT controller was not able to write DAC input data for all enabled DAC channels prior next DAC sampling cycle occurs for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes. For more details about ADC IRT error and DAC IRT error refer to section “ADC IRT and DAC IRT Controllers” later in this chapter.

CAUTION

Neither of ADC IRT and DAC IRT error conditions does not terminate operation of the corresponding IRT controller.

In order to clear data acquisition error(s), host DSP software must perform any of the following actions:

- write to the corresponding bit of *ERR_CLR_RG* register (table 2-10), which allows individual errors clear as well as to clear any combination of errors
- set *ADC-ASYNC-DAQ* data acquisition mode in order to clear *ADC_FIFO_OVF* and *ADC_FIFO_UNF* errors along with reset of synchronous section of ADC data acquisition controller
- set *DAC-ASYNC-DAQ* data acquisition mode in order to clear *DAC_FIFO_OVF* and *DAC_FIFO_UNF* errors along with reset of synchronous section of DAC data acquisition controller
- perform reset of host PIOX-16 interface, which will reset all ADC/DAC data acquisition controllers and ADC/DAC IRT controllers
- perform reset of ADC data acquisition controller and ADC IRT controller by writing to either *ADC_DAQ_RESET_RG* or *ADDA_DAQ_RESET_RG* registers for (written data will be ignored during write)
- perform reset of DAC data acquisition controller and DAC IRT controller by writing to either *DAC_DAQ_RESET_RG* or *ADDA_DAQ_RESET_RG* registers for (written data will be ignored during write).

Any combination of ADC/DAC data acquisition errors, host PIOX-16 interface errors, and ADC/DAC IRT errors can be used to generate host PIOX-16 interrupt requests (refer to tables 2-16 and 2-17). In this case the corresponding host PIOX-16 interrupt request selector (any of the *HIRQ0_SEL_RG..HIRQ3_SEL_RG* register, refer to table 2-16) must be configured to generate interrupt request on logical OR of ADC/DAC FIFO and ADC/DAC IRT error conditions, whereas *XIM_ERR_RG* register (table 2-17) must contain enable masks for the corresponding errors, which are allowed to generate active host PIOX-16 interrupt request. Selected host PIOX-16 interrupt request is generated as logical OR of all enabled error conditions.

2.4 ADC IRT and DAC IRT Controllers

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM provides on-board ADC IRT (interrupt retriggable transmission) controller (fig.2-3a) and DAC IRT controller (fig.2-3b), which shall be used to synchronize host DSP on-chip DMA controllers in order to download real-time ADC output data (or ADC FIFO output data) to host DSP environment and to upload real-time DAC input data (or DAC FIFO input data) from host DSP environment correspondingly during all ADC and DAC data acquisition modes.

ADC IRT and DAC IRT controllers have been designed to benefit from the high-speed data transfer, which is available via TMS320 DSP on-chip DMA controllers. Host TMS320 DSP on-chip DMA controllers, when synchronized by ADC IRT and DAC IRT controllers, provide the most universal and convenient high-performance solution to download/upload real-time ADC/DAC data to/from DSP environment and allow to significantly off-load host DSP core from time consuming data transfer between *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM and DSP environment via PIOX-16 interface thus leaving enough time room for DSP core to perform actual digital signal processing of acquired real-time data.

ADC IRT and DAC IRT controllers do not provide data transfer themselves. Instead, they only generate retriggable synchronization events for host DSP on-chip DMA controllers via any of host PIOX-16 interrupt request lines (refer to tables 2-15 and 2-16), whereas actual data transfer is maintained by DSP on-chip DMA controllers.

> **CAUTION**
>
> ADC IRT and DAC IRT controllers of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM have been designed to utilize only one DMA controller in order to download real-time ADC output data (or ADC FIFO output data) for any number of enabled ADC channels, and only one DMA controller in order to upload real-time DAC input data (or DAC FIFO input data) for any number of enabled DAC channels.

> **CAUTION**
>
> Check with the DSP type of your host *TORNADO* DSP system/controller in order to ensure that it provides required number of on-chip DMA controllers. Recommended selection are *TORNADO* DSP systems/controllers with TMS320C6201/C6202/C6203/C6701 DSP, which provide four DSP on-chip DMA controllers, and *TORNADO* DSP systems/controllers with TMS320C6414/C6415/C6416 DSP, which provide up to 64 DSP on-chip EDMA controllers.

Although ADC IRT controller supports data transfers for all ADC data acquisition modes, it features different timing and configuration for *ADC-ASYNC-DAQ/ADC-SYNC-PX-DAQ* and *ADC-SYNC-FIFO-OPM-DAQ/ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes due to different utilized source registers and source transfer synchronization. Correspondingly, DAC IRT controller supports data transfers for all ADC data acquisition modes, however it features different timing and configuration for *DAC-ASYNC-DAQ/DAC-SYNC-PX-DAQ* and *DAC-SYNC-FIFO-OPM-DAQ/DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes due to different utilized destination registers and destination transfer synchronization.

ADC IRT controller operation for ADC-ASYNC-DAQ and ADC-SYNC-PX-DAQ data acquisition modes

For *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes, only one host TMS320 DSP on-chip DMA controller is required in order to transfer real-time ADC output data to host DSP environment (memory) for any number of enabled ADC channels via *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6). This host DSP on-chip DMA controller must be synchronized by ADC IRT controller via any of IRQ-x (*x*=0..3) host PIOX-16 interrupt request outputs, which must be configured via the corresponding *HIRQx_SEL_RG* register (*x*=0..3) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-15) to generate host PIOX-16 interrupt requests on ADC IRT synchronization events.

In order to allow real-time ADC output data transfer to host DSP environment using host TMS320 DSP on-chip DMA controller synchronized by ADC IRT controller for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes, host DSP on-chip DMA controller, the corresponding *HIRQx_SEL_RG* register (*x*=0..3) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-15), and *ADC0_EN..ADC3_EN* bits of

ADC_CNF_RG register of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-6) shall be configured correctly as described below.

CAUTION

Host TMS320 DSP on-chip DMA controller, corresponding *HIRQx_SEL_RG* register (x=0..3, refer to table 2-15) and *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes shall be configured prior starting ADC data acquisition process, otherwise some of real-time ADC output data will be missed and *ADC_IRT_ERR* bit of *ERR_STAT_RG* register (table 2-10) will be set to the '1' state in order to indicate ADC output data download miss.

Host TMS320 DSP on-chip DMA controller must be configured as the following (refer to the corresponding TI TMS320 DSP user's guide for details how to configure DSP on-chip DMA controllers):

- source transfer address register of DMA controller must be set to the address of *ADC_IRT_DATA_RG* register (table 2-1)
- source address increment feature must be disabled
- source synchronization event for DMA controller must be configured to the corresponding host DSP external interrupt request input, which will be used to route that host PIOX-16 interrupt request input (IRQ-x, *x*=0..3), which is selected to provide synchronization from ADC IRT controller (for more details refer to subsection "Selection of host PIOX-16 interrupt request input for ADC/DAC IRT synchronization events" below)
- destination transfer address register of DMA controller must be assigned to the base address of DSP memory array in accordance with requirements of DSP application
- for *TORNADO* DSP systems/controllers with TMS320C6xxx DSP, the word element size must be set to either 16-bit (recommended) or 32-bit depending upon particular application requirements, however for *TORNADO-E64xx* DSP controllers, word element size must be always set to 16-bit only (refer to *TORNADO-E6x* User's Guide for more details)
- data transfer counter register of DMA controller must be set to the number of desired *ADC sampling cycles* for *ADC-ASYNC-DAQ* data acquisition mode, or to the number of desired *ADC scan cycles* for *ADC-SYNC-PX-DAQ* data acquisition mode, multiplied by the number of enabled ADC channels via *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6)
- for *TORNADO* DSP systems/controllers with TMS320C6xxx DSP, DMA auto-initialization feature must be enabled upon the application requirements.

CAUTION

Real-time ADC output data, which is transferred to host DSP environment memory by host DSP on-chip DMA controller synchronized by ADC IRT controller, is packed as a continuous series of ADC data packets, each corresponding to either one *ADC sampling cycle* (for *ADC-ASYNC-DAQ* data acquisition mode) or to one *ADC scan cycle* (for *ADC-SYNC-PX-DAQ* data acquisition mode), and contains ADC output data words for all enabled ADC channels in accordance with the *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (ADC channels are scanned starting from the ADC-0 channel with the ADC-3 channel being the last one).

The following is how *HIRQx_SEL_RG* register (*x*=0..3) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-15) must be configured in order to transfer source transfer synchronization events from ADC IRT controller to host DSP on-chip DMA controller:

- *HIRQx_EN* bit of *HIRQx_SEL_RG* register (*x*=0..3) (table 2-15) must be set to the ‘1’ state in order to enable IRQ-x host PIOX-16 interrupt request output
- {*HIRQx_SEL*-[3:0]} bits (or {*HIRQx_SEL*-[4:0]} bits) of *HIRQx_SEL_RG* register (*x*=0..3) (tables 2-15 and 2-16) shall be set to the {0,1,0,1} (or {0,0,1,0,1}) state in order to select interrupt from ADC IRT controller as the start synchronization event for host DSP on-chip DMA controller
- *HIRQx_POL* bit of *HIRQx_SEL_RG* register (*x*=0..3) (table 2-15) must be set upon the type of host TORNADO DSP system/controller. For more details refer to subsection “Configuration of host PIOX-16 interrupt request output for ADC/DAC IRT synchronization events” below.

Finally, *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM shall be set to enable those ADC channels, which will be involved into A/D data acquisition and real-time ADC output data transfer using host DSP on-chip DMA controller synchronized by ADC IRT controller.

Timing diagram for ADC IRT data transfer cycles for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes is presented at figure 2-24. Presented example corresponds to ADC-0, ADC-1 and ADC-3 channels enabled and IRQ-0 host PIOX-16 interrupt request being used to generate positive source synchronization events for host DSP on-chip DMA controller.

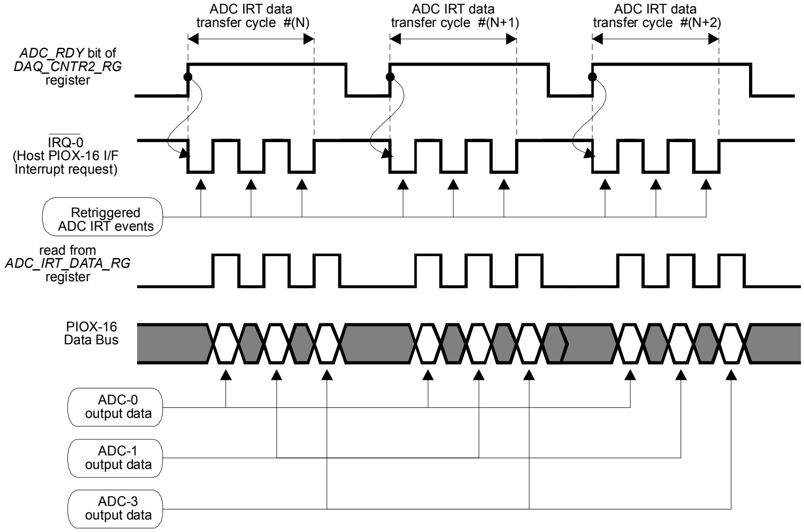
ADC IRT data transfer cycle #(N)
ADC IRT data transfer cycle #(N+1)
ADC IRT data transfer cycle #(N+2)
ADC_RDY bit of DAQ_CNTR2_RG register
IRQ-0 (Host PIOX-16 I/F Interrupt request)
Retriggered ADC IRT events
read from ADC_IRT_DATA_RG register
PIOX-16 Data Bus
ADC-0 output data
ADC-1 output data
ADC-3 output data

Fig.2-24. ADC IRT data transfer cycles for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes and positive ADC IRT data transfer synchronization events.

Each ADC IRT data transfer cycle for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes is initialized on ADC data ready condition and will generate as many source data transfer synchronization events via selected host PIOX-16 interrupt request output in one ADC IRT data transfer cycle, as many ADC channels have been enabled via *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register. ADC IRT controller automatically take care of which particular ADC channels are involved in ADC IRT data transfer cycle in accordance with *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register. Each generated start transfer synchronization event by ADC IRT controller is used by host DSP on-chip DMA controller to read ADC output data for particular ADC channel. Inside each ADC IRT data transfer cycle, real-time ADC output data for enabled ADC channels are read starting from the first enabled ADC channel with the ADC channels being scanned beginning from the ADC-0 channel and proceeds up to ADC-3 channel.

ADC IRT controller operation for ADC-SYNC-FIFO-OPM-DAQ and ADC-SYNC-FIFO-PTM-DAQ data acquisition modes

For *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes, only one host TMS320 DSP on-chip DMA controller is required in order to transfer ADC FIFO output data to host DSP environment (memory). This host DSP on-chip DMA controller must be synchronized by ADC IRT controller via any of IRQ-x (*x*=0..3) host PIOX-16 interrupt request outputs, which must be configured via the corresponding *HIRQx_SEL_RG* register (*x*=0..3) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-15) to generate host PIOX-16 interrupt requests on ADC IRT synchronization events.

In order to allow real-time ADC output data transfer to host DSP environment using host TMS320 DSP on-chip DMA controller synchronized by ADC IRT controller for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes, host DSP on-chip DMA controller, the corresponding *HIRQx_SEL_RG* register (*x*=0..3) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-15), and *DAQ_CNTR3_RG* register (table 2-4) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM shall be configured correctly as described below.

CAUTION

ADC0_EN..ADC3_EN bits of *ADC_CNF_RG* register of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-6) do not effect ADC IRT controller operation for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes, and are used instead by ADC data acquisition controller to define particular ADC channels, which real-time ADC output data will be pushed into ADC FIFO in each ADC scan cycle.

For *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes, ADC IRT controller does not concern what particular ADC channels are involved in ADC synchronous data acquisition process and are pushed by ADC data acquisition controller into ADC FIFO in each ADC scan cycle. Once any of ADC synchronous FIFO data acquisition modes has been selected, then ADC IRT controller is immediately enabled and starts ‘hunting’ for transmission start event. Active transmission start event for ADC IRT controller for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes is detected in case either ADC FIFO PAF or ADC FIFO FF flag comes active depending upon the state of *ADC_SYNC_FIFO_DAQ_TF_SEL* bit of *DAQ_CNTR3_RG* register (table 2-4) in accordance with table 2-21.

Table 2-21. ADC IRT transmission start event selector.

ADC_SYNC_FIFO_DAQ_TF_SEL **bit of** ***DAQ_CNTR3_RG*** **register**	**transmission start event for ADC IRT controller**
0	ADC FIFO FF flag is set to the ‘1’ state
1	ADC FIFO PAF flag is set to the ‘1’ state

Note: 1. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

Once transmission start event has been detected, then ADC IRT controller begin to continuously generate start transfer synchronization events for host DSP on-chip DMA controller over selected host PIOX-16 interrupt request line until either transmission termination event will be detected, or in case ADC data acquisition controller will be reset via host DSP write to either *ADC_DAQ_RESET_RG* or *ADDA_DAQ_RESET_RG* registers (table 2-1). Each generated start transfer synchronization event is used by host DSP on-chip DMA controller to read one data word from ADC FIFO. Active transmission termination event for ADC IRT controller is detected in case either ADC FIFO PAE or ADC FIFO EF flag comes active depending upon the state of the *ADC_SYNC_FIFO_IRT_TF_SEL* bit of *DAQ_CNTR3_RG* register (table 2-4) in accordance with table 2-22.

Table 2-22. ADC IRT transmission termination event selector.

ADC_SYNC_FIFO_IRT_TF_SEL **bit of** ***DAQ_CNTR3_RG*** **register**	**transmission termination event for ADC IRT controller**
0	ADC FIFO EF flag is set to the '1' state
1	ADC FIFO PAE flag is set to the '1' state

Note: 1. Highlighted configurations correspond to default settings on PIOX-16 reset

CAUTION

Host TMS320 DSP on-chip DMA controller, corresponding *HIRQx_SEL_RG* register (x=0..3, refer to table 2-15) and *DAQ_CNTR3_RG* register (table 2-4) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM for *ADC-SYNC-FIFO-OPM-DAQ* and *ADC-SYNC-FIFO-PTM-DAQ* data acquisition modes can be generally configured after start of ADC data acquisition process, however care must be taken to ensure that ADC FIFO does not overflow prior host DSP on-chip DMA controller will initialize.

Host TMS320 DSP on-chip DMA controller must be configured as the following (refer to the corresponding TI TMS320 DSP user's guide for details how to configure DSP on-chip DMA controllers):

- source transfer address register of DMA controller must be set to the address of *ADC_FIFO_DATA_RG* register (table 2-1)
- source address increment feature must be disabled
- source synchronization event for DMA controller must be configured to the corresponding host DSP external interrupt request input, which will be used to route that host PIOX-16 interrupt request input (IRQ-x, *x*=0..3), which is selected to provide synchronization from ADC IRT controller (for more details refer to subsection "Selection of host PIOX-16 interrupt request input for ADC/DAC IRT synchronization events" below)
- destination transfer address register of DMA controller must be assigned to the base address of DSP memory array in accordance with requirements of DSP application
- for *TORNADO* DSP systems/controllers with TMS320C6xxx DSP, the word element size must be set to either 16-bit (recommended) or 32-bit depending upon particular application requirements, however for *TORNADO-E64xx* DSP controllers, word element size must be always set to 16-bit only (refer to *TORNADO-E6x* User's Guide for more details)
- data transfer counter register of DMA controller must be set to any value, however recommended is the number of desired *ADC scan cycles* multiplied by the number of enabled ADC channels via *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6)

CAUTION

It is the responsibility of host DSP software to configure ADC FIFO PAF/PAE flags offsets and data transfer counter register of DMA controller so, that one ADC IRT data transmission cycle will include as many ADC FIFO output data words as fits into desired integer number of ADC scan cycles, otherwise host DSP software must provide additional efforts to correctly unpack ADC FIFO data on the boundary ADC scan cycle conditions.

- for *TORNADO* DSP systems/controllers with TMS320C6xxx DSP, DMA auto-initialization feature must be enabled upon the application requirements.

CAUTION

ADC FIFO output data, which is transferred to host DSP environment memory by host DSP on-chip DMA controller synchronized by ADC IRT controller, is packed as a continuous series of ADC data packets, each corresponding to either one *ADC scan cycle*, and contains ADC output data words for all enabled ADC channels in accordance with the *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register.

The following is how *HIRQx_SEL_RG* register (*x*=0..3) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-15) must be configured in order to transfer source transfer synchronization events from ADC IRT controller to host DSP on-chip DMA controller:

- *HIRQx_EN* bit of *HIRQx_SEL_RG* register (*x*=0..3) (table 2-15) must be set to the '1' state in order to enable IRQ-x host PIOX-16 interrupt request output
- {*HIRQx_SEL*-[3:0]} bits (or {*HIRQx_SEL*-[4:0]} bits) of *HIRQx_SEL_RG* register (*x*=0..3) (tables 2-15 and 2-16) shall be set to the {0,1,0,1} (or {0,0,1,0,1}) state in order to select interrupt from ADC IRT controller as the start synchronization event for host DSP on-chip DMA controller
- *HIRQx_POL* bit of *HIRQx_SEL_RG* register (*x*=0..3) (table 2-15) must be set upon the type of host TORNADO DSP system/controller. For more details refer to subsection "Configuration of host PIOX-16 interrupt request output for ADC/DAC IRT synchronization events" below.

DAQ_CNTR3_RG register (table 2-4) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM must be set in accordance with tables 2-21 and 2-22 in order to select ADC FIFO flags, which will be used to start and terminate ADC IRT data transmission cycle.

Finally, *ADC0_EN..ADC3_EN* bits of *ADC_CNF_RG* register (table 2-6) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM shall be set to enable those ADC channels, which will be pushed into ADC FIFO in each ADC can cycle.

Timing diagram for ADC IRT data transfer cycles for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode is presented at figure 2-25. Presented example corresponds to the ADC FIFO PAF flag being used to normally terminate ADC synchronous FIFO one-pass data acquisition process and to initialize transmission cycle of ADC IRT controller, whereas ADC FIFO EF is used to terminate ADC IRT transmission cycle. Figure 2-25 assumes that IRQ-0 host PIOX-16 interrupt request being used to generate positive source synchronization events for host DSP on-chip DMA controller.

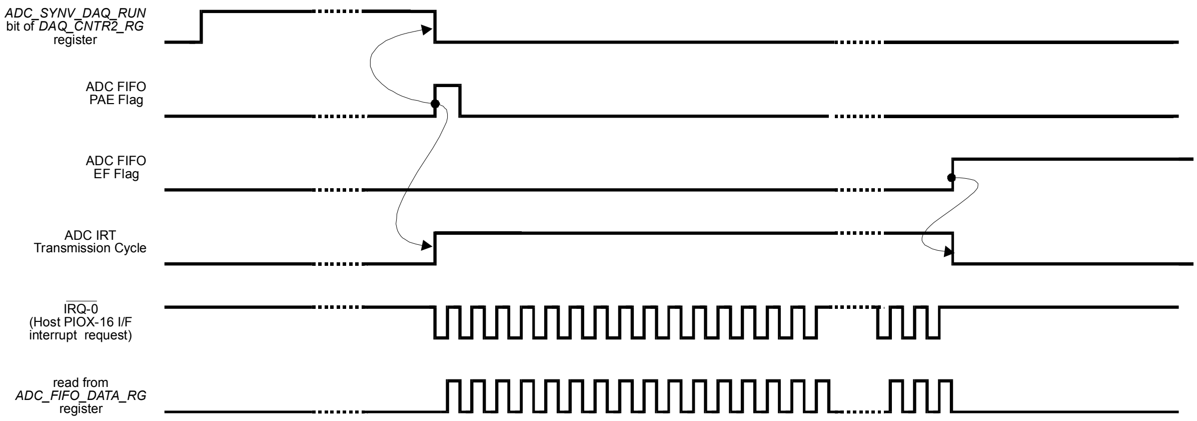

Fig.2-25. ADC IRT data transmission cycle for *ADC-SYNC-FIFO-OPM-DAQ* data acquisition mode and positive ADC IRT data transfer synchronization events.

DAC IRT controller operation for DAC-ASYNC-DAQ and DAC-SYNC-PX-DAQ data acquisition modes

For *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes, only one host TMS320 DSP on-chip DMA controller is required in order to transfer real-time DAC input data from host DSP environment (memory) for any number of enabled DAC channels via *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-6). This host DSP on-chip DMA controller must be synchronized by DAC IRT controller via any of IRQ-x (*x*=0..3) host PIOX-16 interrupt request outputs, which must be configured via the corresponding *HIRQx_SEL_RG* register (*x*=0..3) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-15) to generate host PIOX-16 interrupt requests on DAC IRT synchronization events.

In order to allow real-time DAC input data transfer to host DSP environment using host TMS320 DSP on-chip DMA controller synchronized by DAC IRT controller for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes, host DSP on-chip DMA controller, the corresponding *HIRQx_SEL_RG* register (*x*=0..3) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-15), and *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-6) shall be configured correctly as described below.

CAUTION

Host TMS320 DSP on-chip DMA controller, corresponding *HIRQx_SEL_RG* register (x=0..3, refer to table 2-15) and *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes shall be configured prior starting DAC data acquisition process, otherwise some of real-time DAC input data will be missed and *DAC_IRT_ERR* bit of *ERR_STAT_RG* register (table 2-10) will be set to the '1' state in order to indicate DAC input data upload miss.

Host TMS320 DSP on-chip DMA controller must be configured as the following (refer to the corresponding TI TMS320 DSP user's guide for details how to configure DSP on-chip DMA controllers):

- destination transfer address register of DMA controller must be set to the address of *DAC_IRT_DATA_RG* register (table 2-1)
- destination address increment feature must be disabled
- destination synchronization event for DMA controller must be configured to the corresponding host DSP external interrupt request input, which will be used to route that host PIOX-16 interrupt request input (IRQ-x, *x*=0..3), which is selected to provide synchronization from DAC IRT controller (for more details refer to subsection "Selection of host PIOX-16 interrupt request input for ADC/DAC IRT synchronization events" below)
- source transfer address register of DMA controller must be assigned to the base address of DSP memory array in accordance with requirements of DSP application
- for *TORNADO* DSP systems/controllers with TMS320C6xxx DSP, the word element size must be set to either 16-bit (recommended) or 32-bit depending upon particular application requirements, however for *TORNADO-E64xx* DSP controllers, word element size must be always set to 16-bit only (refer to *TORNADO-E6x* User's Guide for more details)
- data transfer counter register of DMA controller must be set to the number of desired *DAC sampling cycles* multiplied by the number of enabled DAC channels via *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-6)
- for *TORNADO* DSP systems/controllers with TMS320C6xxx DSP, DMA auto-initialization feature must be enabled upon the application requirements.

CAUTION

Real-time DAC input data, which is transferred from host DSP environment memory by host DSP on-chip DMA controller synchronized by DAC IRT controller, is packed as a continuous series of DAC data packets, each corresponding to one *DAC sampling cycle*, and contains DAC input data words for all enabled DAC channels in accordance with the *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (DAC channels are scanned starting from the DAC-0 channel with the DAC-3 channel being the last one).

The following is how *HIRQx_SEL_RG* register (*x*=0..3) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-15) must be configured in order to transfer source transfer synchronization events from DAC IRT controller to host DSP on-chip DMA controller:

- *HIRQx_EN* bit of *HIRQx_SEL_RG* register (*x*=0..3) (table 2-15) must be set to the '1' state in order to enable IRQ-x host PIOX-16 interrupt request output
- {*HIRQx_SEL*-[3:0]} bits (or {*HIRQx_SEL*-[4:0]} bits) of *HIRQx_SEL_RG* register (*x*=0..3) (tables 2-15 and 2-16) shall be set to the {0,1,1,0} (or {0,0,1,1,0}) state in order to select interrupt from DAC IRT controller as the destination synchronization event for host DSP on-chip DMA controller
- *HIRQx_POL* bit of *HIRQx_SEL_RG* register (*x*=0..3) (table 2-15) must be set upon the type of host TORNADO DSP system/controller. For more details refer to subsection "Configuration of host PIOX-16 interrupt request output for ADC/DAC IRT synchronization events" below.

Finally, *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-6) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM shall be set to enable those DAC channels, which will be involved into A/D data acquisition and real-time DAC output data transfer using host DSP on-chip DMA controller synchronized by DAC IRT controller.

Timing diagram for DAC IRT data transfer cycles for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes is presented at figure 2-26. Presented example corresponds to DAC-0, DAC-1 and DAC-3 channels enabled and IRQ-0 host PIOX-16 interrupt request being used to generate positive destination synchronization events for host DSP on-chip DMA controller.

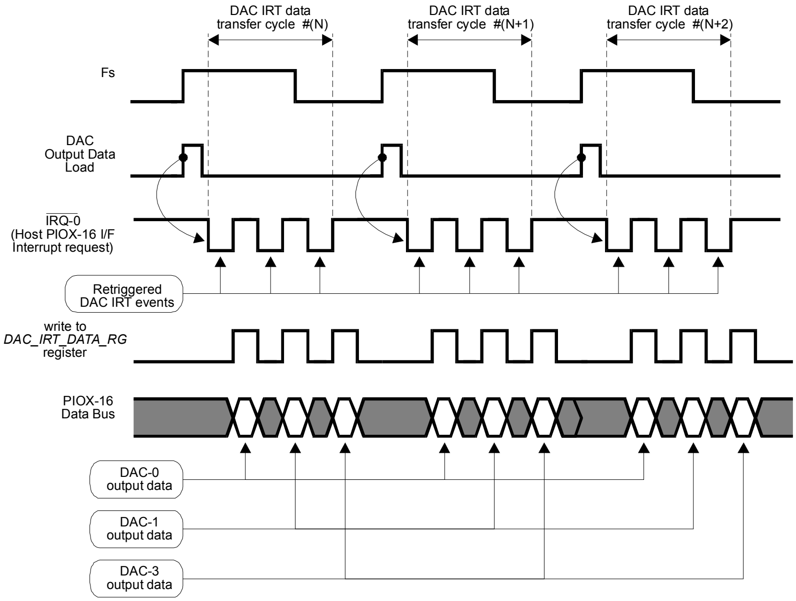
DAC IRT data transfer cycle #(N)
DAC IRT data transfer cycle #(N+1)
DAC IRT data transfer cycle #(N+2)
Fs
DAC Output Data Load
IRQ-0 (Host PIOX-16 I/F Interrupt request)
Retriggered DAC IRT events
write to DAC_IRT_DATA_RG register
PIOX-16 Data Bus
DAC-0 output data
DAC-1 output data
DAC-3 output data

Fig.2-26. DAC IRT data transfer cycles for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes and positive DAC IRT data transfer synchronization events.

Each DAC IRT data transfer cycle for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes is initialized on hardware defined sampling frequency event and will generate as many destination data transfer synchronization events via selected host PIOX-16 interrupt request output in one DAC IRT data transfer cycle, as many DAC channels have been enabled via *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register. DAC IRT controller automatically take care of which particular DAC channels are involved in DAC IRT data transfer cycle in accordance with *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register. Each generated destination transfer synchronization event by DAC IRT controller is used by host DSP on-chip DMA controller to write DAC input data for particular DAC channel. Inside each DAC IRT data transfer cycle, real-time DAC input data for enabled DAC channels are written starting from the first enabled DAC channel with the DAC channels being scanned beginning from the DAC-0 channel and proceeds up to DAC-3 channel.

DAC IRT controller operation for DAC-SYNC-FIFO-OPM-DAQ and DAC-SYNC-FIFO-PTM-DAQ data acquisition modes

For *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes, only one host TMS320 DSP on-chip DMA controller is required in order to transfer DAC FIFO input data from host DSP environment (memory). This host DSP on-chip DMA controller must be synchronized by DAC IRT controller via any of IRQ-x (*x*=0..3) host PIOX-16 interrupt request outputs, which must be configured via the corresponding *HIRQx_SEL_RG* register (*x*=0..3) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-15) to generate host PIOX-16 interrupt requests on DAC IRT synchronization events.

In order to allow transfer of DAC FIFO input data from host DSP environment using host TMS320 DSP on-chip DMA controller synchronized by DAC IRT controller for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes, host DSP on-chip DMA controller, the corresponding *HIRQx_SEL_RG* register (*x*=0..3) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-15), and *DAQ_CNTR3_RG* register (table 2-4) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM shall be configured correctly as described below.

CAUTION

DAC0_EN..DAC3_EN bits of *DAC_CNF_RG* register of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-6) do not effect DAC IRT controller operation for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes, and are used instead by DAC data acquisition controller to define particular DAC channels, which real-time DAC input data will be extracted from DAC FIFO in each sampling period.

For *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes, DAC IRT controller does not concern what particular DAC channels are involved in DAC synchronous data acquisition process and are extracted by DAC data acquisition controller from DAC FIFO in each sampling period. Once any of DAC synchronous FIFO data acquisition modes has been selected, then DAC IRT controller is immediately enabled and starts ‘hunting’ for transmission start event. Active transmission start event for DAC IRT controller for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes is detected in case either DAC FIFO PAE or DAC FIFO EF flag comes active depending upon the state of *DAC_SYNC_FIFO_DAQ_TF_SEL* bit of *DAQ_CNTR3_RG* register (table 2-4) in accordance with table 2-23.

Table 2-23. DAC IRT transmission start event selector.

***DAC_SYNC_FIFO_DAQ_TF_SEL* bit of *DAQ_CNTR3_RG* register**	**transmission start event for DAC IRT controller**
0	DAC FIFO EF flag is set to the ‘1’ state
1	DAC FIFO PAE flag is set to the ‘1’ state

Note: 1. Highlighted configurations correspond to default settings on PIOX-16 reset condition.

Once transmission start event has been detected, then DAC IRT controller begin to continuously generate destination transfer synchronization events for host DSP on-chip DMA controller over selected host PIOX-16 interrupt request line until either transmission termination event will be detected, or in case DAC data acquisition controller will be reset via host DSP write to either *DAC_DAQ_RESET_RG* or *ADDA_DAQ_RESET_RG* registers (table 2-1). Each generated start transfer synchronization event is used by host DSP on-chip DMA controller to write one data word to DAC FIFO. Active transmission termination event for DAC IRT controller is detected in case either DAC FIFO PAF or DAC FIFO FF flag comes active depending upon the state of the *DAC_SYNC_FIFO_IRT_TF_SEL* bit of *DAQ_CNTR3_RG* register (table 2-4) in accordance with table 2-24.

Table 2-24. DAC IRT transmission termination event selector.

***DAC_SYNC_FIFO_IRT_TF_SEL* bit of *DAQ_CNTR3_RG* register**	**transmission termination event for DAC IRT controller**
0	DAC FIFO FF flag is set to the '1' state
1	DAC FIFO PAF flag is set to the '1' state

Note: 1. Highlighted configurations correspond to default settings on PIOX-16 reset

CAUTION

DAC FIFO must already contain real-time DAC input data prior initialization of DAC synchronous data acquisition process, otherwise DAC FIFO underflow condition will occur.

CAUTION

Host TMS320 DSP on-chip DMA controller, corresponding *HIRQx_SEL_RG* register (x=0..3, refer to table 2-15) and *DAQ_CNTR3_RG* register (table 2-4) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM for *DAC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-PTM-DAQ* data acquisition modes can be generally configured after start of DAC data acquisition process, however care must be taken to ensure that DAC FIFO already contains enough real-time DAC input data, which will be extracted by DAC data acquisition controller prior host DSP on-chip DMA controller will initialize, otherwise DAC FIFO underflow condition will occur.

Host TMS320 DSP on-chip DMA controller must be configured as the following (refer to the corresponding TI TMS320 DSP user's guide for details how to configure DSP on-chip DMA controllers):

- destination transfer address register of DMA controller must be set to the address of *DAC_FIFO_DATA_RG* register (table 2-1)
- destination address increment feature must be disabled
- destination synchronization event for DMA controller must be configured to the corresponding host DSP external interrupt request input, which will be used to route that host PIOX-16 interrupt request input (IRQ-x, *x*=0..3), which is selected to provide synchronization from DAC IRT controller (for more details refer to subsection "Selection of host PIOX-16 interrupt request input for ADC/DAC IRT synchronization events" below)
- source transfer address register of DMA controller must be assigned to the base address of DSP memory array in accordance with requirements of DSP application
- for *TORNADO* DSP systems/controllers with TMS320C6xxx DSP, the word element size must be set to either 16-bit (recommended) or 32-bit depending upon particular application requirements, however for *TORNADO-E64xx* DSP controllers, word element size must be always set to 16-bit only (refer to *TORNADO-E6x* User's Guide for more details)

- data transfer counter register of DMA controller must be set to any value, however recommended is the number of desired *DAC sampling cycles* (sampling periods) multiplied by the number of enabled DAC channels via *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-6)

> **CAUTION**
>
> It is the responsibility of host DSP software to configure DAC FIFO PAF/PAE flags offsets and data transfer counter register of DMA controller so, that one DAC IRT data transmission cycle will include as many DAC FIFO input data words as fits into desired integer number of DAC sampling cycles (sampling periods), otherwise host DSP software must provide additional efforts to correctly pack DAC FIFO data on the boundary DAC sampling cycle conditions.

- for *TORNADO* DSP systems/controllers with TMS320C6xxx DSP, DMA auto-initialization feature must be enabled upon the application requirements.

> **CAUTION**
>
> DAC input data, which is transferred from host DSP environment memory by host DSP on-chip DMA controller synchronized by DAC IRT controller, is packed as a continuous series of DAC data packets, each corresponding to either one *DAC sampling cycle*, and contains DAC output data words for all enabled DAC channels in accordance with the *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register.

The following is how *HIRQx_SEL_RG* register (*x*=0..3) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (table 2-15) must be configured in order to transfer source transfer synchronization events from DAC IRT controller to host DSP on-chip DMA controller:

- *HIRQx_EN* bit of *HIRQx_SEL_RG* register (*x*=0..3) (table 2-15) must be set to the ‘1’ state in order to enable IRQ-x host PIOX-16 interrupt request output
- {*HIRQx_SEL*-[3:0]} bits (or {*HIRQx_SEL*-[4:0]} bits) of *HIRQx_SEL_RG* register (*x*=0..3) (tables 2-15 and 2-16) shall be set to the {0,1,1,0} (or {0,0,1,1,0}) state in order to select interrupt from DAC IRT controller as the destination synchronization event for host DSP on-chip DMA controller
- *HIRQx_POL* bit of *HIRQx_SEL_RG* register (*x*=0..3) (table 2-15) must be set upon the type of host TORNADO DSP system/controller. For more details refer to subsection “Configuration of host PIOX-16 interrupt request output for ADC/DAC IRT synchronization events” below.

DAQ_CNTR3_RG register (table 2-4) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM must be set in accordance with tables 2-23 and 2-24 in order to select DAC FIFO flags, which will be used to start and terminate DAC IRT data transmission cycle.

Finally, *DAC0_EN..DAC3_EN* bits of *DAC_CNF_RG* register (table 2-6) of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM shall be set to enable those DAC channels, which will be pushed into DAC FIFO in each DAC can cycle.

Timing diagram for DAC IRT data transfer cycles for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode is presented at figure 2-27. Presented example corresponds to the DAC FIFO PAE flag being used to normally terminate DAC synchronous FIFO one-pass data acquisition process and to initialize transmission cycle of DAC

IRT controller, whereas DAC FIFO PAF is used to terminate DAC IRT transmission cycle. Figure 2-27 assumes that IRQ-0 host PIOX-16 interrupt request being used to generate positive destination synchronization events for host DSP on-chip DMA controller. Note, that fill-in of DAC FIFO is performed prior DAC synchronous data acquisition process will be initialized.

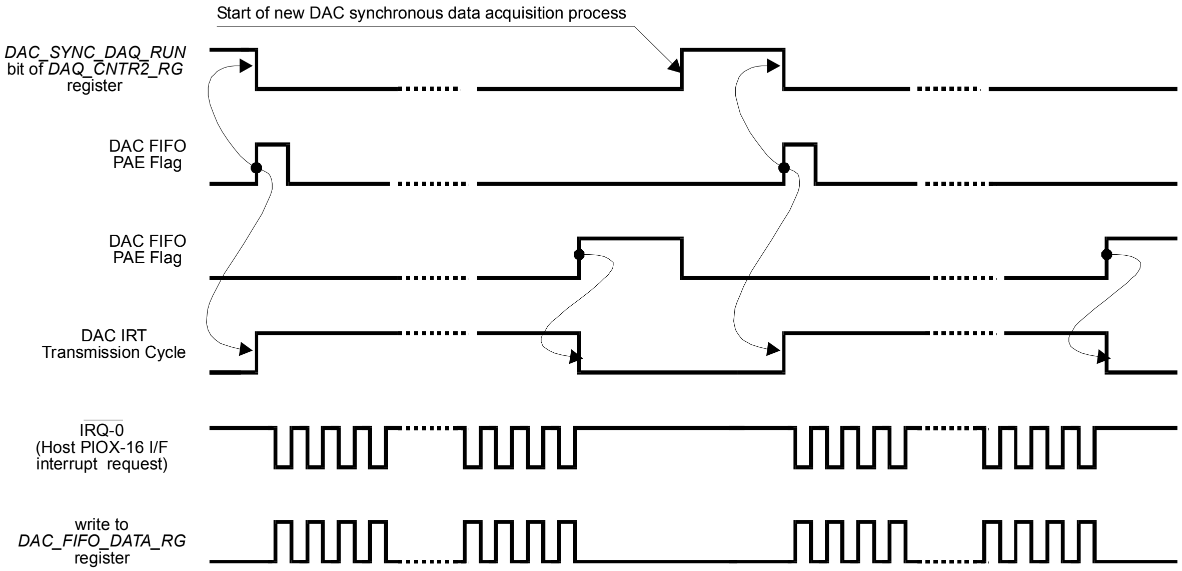
Start of new DAC synchronous data acquisition process
DAC_SYNC_DAQ_RUN bit of DAQ_CNTR2_RG register
DAC FIFO PAE Flag
DAC FIFO PAE Flag
DAC IRT Transmission Cycle
IRQ-0 (Host PIOX-16 I/F interrupt request)
write to DAC_FIFO_DATA_RG register

Fig.2-27. DAC IRT data transmission cycle for *DAC-SYNC-FIFO-OPM-DAQ* data acquisition mode and positive DAC IRT data transfer synchronization events.

Configuration of host PIOX-16 interrupt request output for ADC/DAC IRT synchronization events

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM can generate ADC/DAC IRT synchronization events with either positive or negative polarity for host DSP on-chip DMA controller over any of four host PIOX-16 interrupt request lines (IRQ-0..3). Refer to Appendix B for more details about host PIOX-16 interface.

CAUTION

Always check with your host *TORNADO* DSP system/controller user's guide and user's guide(s) for other installed SIOX rev.B/C DCM (if any) in order to learn what particular interrupt requests from host PIOX-16 interface can be used to generate external interrupt requests to host DSP.

Some *TORNADO* DSP systems/controllers (for example, *TORNADO-P3x/P6xxx/E6202/E6203/E64xx*) provide flexible software configured interrupt request selectors for each external interrupt requests of on-board DSP, whereas other *TORNADO* DSP systems/controllers (for example, *TORNADO-3x/6x/E3x/E62/E67*) feature unconfigured or partially configured PCB wired map of on-board interrupt request sources and external interrupt requests of on-board DSP.

CAUTION

In case your host *TORNADO* DSP system/controller uses on-board shared interrupt requests for on-board SIOX rev.B/C and PIOX/PIOX-16 expansion interface sites and there are other SIOX rev.B/C DCM installed, then you must select that interrupt request input of host PIOX/PIOX-16 interface, which is not used by other installed SIOX rev.B/C DCM.

CAUTION

You must select two different interrupt request inputs of host PIOX-16 interface of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM in order to transfer ADC IRT and DAC IRT synchronization events separately to host TMS320 DSP on-chip DMA controllers.

After particular host PIOX-16 interrupt request line, which will be used to transfer ADC IRT or DAC IRT synchronization events to DSP on-chip DMA controller, has been selected, the corresponding *HIRQx_SEL_RG* interrupt selector register (*x*=0..3, refer to table 2-15) must be configured correctly in order to transfer interrupt request from ADC/DAC IRT controller (table 2-16) with either positive or negative interrupt request polarity.

In order to configure particular host PIOX-16 interrupt request output of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM to generate either ADC IRT or DAC IRT synchronization events, the following bits of the corresponding *HIRQx_SEL_RG* register (*x*=0..3, table 2-15) shall be configured:

- *HIRQx_EN* bit of the corresponding *HIRQx_SEL_RG* register (*x*=0..3) (table 2-15) must be set to the ‘1’ state in order to enable IRQ-x host PIOX-16 interrupt request output
- {*HIRQx_SEL*-[4:0]} bits) of the corresponding *HIRQx_SEL_RG* register (*x*=0..3, tables 2-15 and 2-16) shall be set to {0,0,1,0,1} or {0,0,1,1,0} state in order to provide interrupt from ADC IRT and DAC IRT controllers correspondingly as the synchronization event for host DSP on-chip DMA controller(s)
- *HIRQx_POL* bit of *HIRQx_SEL_RG* register (*x*=0..3) (table 2-15) must be set upon the type of host TORNADO DSP system/controller as described below in order to provide correct output polarity of host PIOX-16 interrupt request output signal of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, which is used to generate either ADC IRT or DAC IRT synchronization events once this interrupt request line is used to generate synchronization events for host DSP on-chip DMA controller rather than interrupt request to host DSP.

CAUTION

AD16/16Q/1M-DA4/16Q/1M AD/DA DCM allows to transfer ADC/DAC IRT synchronization events to host DSP on-chip DMA controller with software configured either positive or negative polarity over any of IRQ-0..3 interrupt request inputs of host PIOX-16 interface in order to provide compatibility with virtually any host *TORNADO* DSP system/controller.

When setting *HIRQx_POL* polarity bit of *HIRQx_SEL_RG* register (*x*=0..3, table 2-15) in order to select polarity of ADC/DAC synchronization event transferred over the corresponding IRQ-x (*x*=0..3) host PIOX-16 interface interrupt request output, the following has to be considered:

- *HIRQx_POL* bit of *HIRQx_SEL_RG* register must be set to the '0' state for host *TORNADO-3x/P3x/E3x/P62/P6202/P6203/P67/P64xx/E6202/E6203/E64xx* DSP systems and controllers in order to select positive polarity for transferred ADC/DAC IRT synchronization event
- *HIRQx_POL* bit of *HIRQx_SEL_RG* register must be set to the '1' state for *TORNADO-62/67/E62/E67* DSP systems and controllers in order to select negative polarity for transferred ADC/DAC IRT synchronization event.

CAUTION

ADC IRT positive polarity synchronization event, which is passed over active low host PIOX-16 interrupt request inputs, results in active low host PIOX-16 interrupt request, whereas ADC IRT negative polarity synchronization event results in active high host PIOX-16 interrupt request.

Timing diagram at figures 2-28a and 2-28b show ADC IRT data transfer cycles for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes for positive and negative polarity correspondingly of generated ADC IRT synchronization events. Presented example corresponds to ADC-0, ADC-1 and ADC-3 channels enabled and IRQ-0 host PIOX-16 interrupt request being used to generate source synchronization events for host DSP on-chip DMA controller.

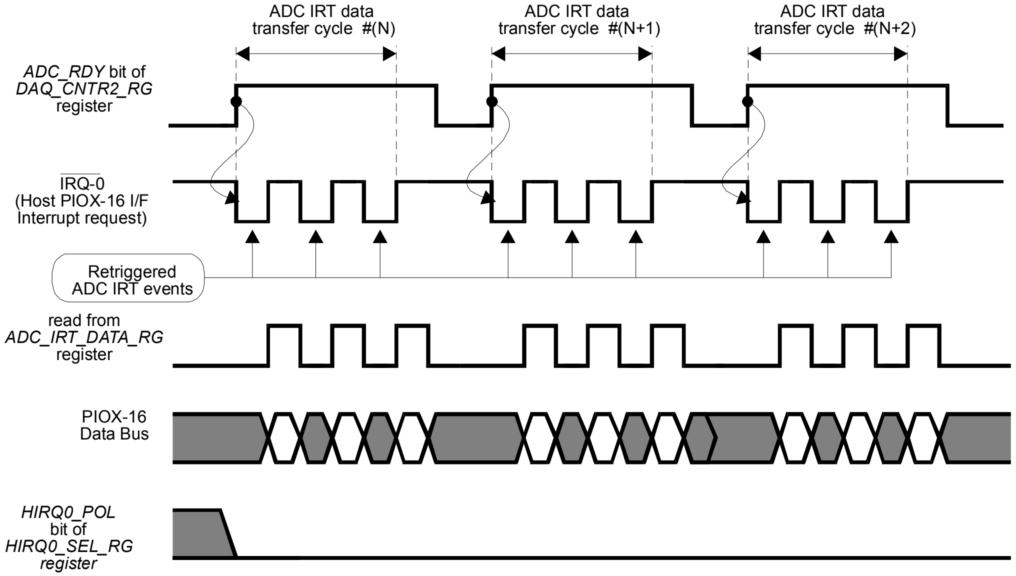
ADC IRT data transfer cycle #(N)
ADC IRT data transfer cycle #(N+1)
ADC IRT data transfer cycle #(N+2)
ADC_RDY bit of *DAQ_CNTR2_RG* register
IRQ-0 (Host PIOX-16 I/F Interrupt request)
Retriggered ADC IRT events
read from *ADC_IRT_DATA_RG* register
PIOX-16 Data Bus
HIRQ0_POL bit of *HIRQ0_SEL_RG* register

Fig.2-28a. ADC IRT data transfer cycles for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes and positive ADC IRT data transfer synchronization events.

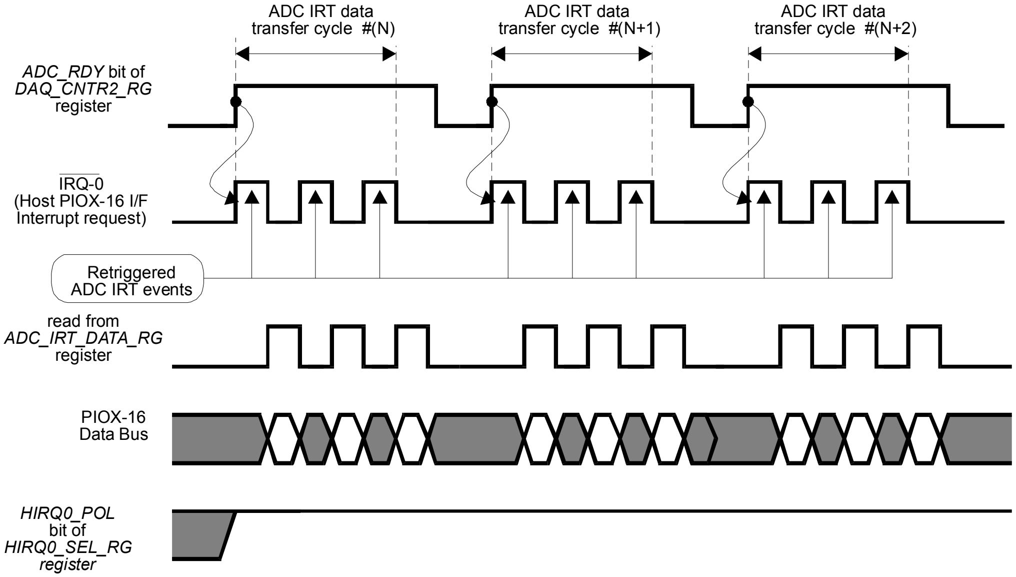
ADC IRT data transfer cycle #(N)
ADC IRT data transfer cycle #(N+1)
ADC IRT data transfer cycle #(N+2)
ADC_RDY bit of *DAQ_CNTR2_RG* register
IRQ-0 (Host PIOX-16 I/F Interrupt request)
Retriggered ADC IRT events
read from *ADC_IRT_DATA_RG* register
PIOX-16 Data Bus
HIRQ0_POL bit of *HIRQ0_SEL_RG* register

Fig.2-28b. ADC IRT data transfer cycles for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes and negative ADC IRT data transfer synchronization events.

error conditions for ADC IRT controller

In case either *ADC-ASYNC-DAQ* or *ADC-SYNC-PX-DAQ* data acquisition modes are selected and host DSP on-chip DMA controller synchronized by ADC IRT controller are used to transfer real-time ADC output data to host DSP environment, then ADC IRT controller continuously performs real-time tracking for ADC IRT error condition.

ADC IRT error condition is detected in case host DSP on-chip DMA controller synchronized by ADC IRT controller is not able to read ADC output data for all enabled ADC channels prior next ADC data comes ready for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes. This error condition corresponds to insufficient traffic reserved over DSP data bus for such a high ADC sampling frequency.

ADC IRT error condition is indicated via the ‘1’ state of *ADC_IRT_ERR* bit of *ERR_STAT_RG* register, which will remain in the ‘1’ state until reset by host DSP software. ADC IRT error condition can e used to generate host PIOX-16 interrupt request.

For more details about errors processing refer to subsection “Errors processing” inside “Data Acquisition Control” section earlier in this chapter.

error conditions for DAC IRT controller

In case either *DAC-ASYNC-DAQ* or *DAC-SYNC-PX-DAQ* data acquisition modes are selected and host DSP on-chip DMA controller synchronized by DAC IRT controller are used to transfer real-time DAC input data from host DSP environment, then DAC IRT controller continuously perform real-time tracking for DAC IRT error condition.

DAC IRT error condition is detected in case host DSP on-chip DMA controller synchronized by DAC IRT controller was not able to write DAC input data for all enabled DAC channels prior next DAC sampling cycle occurs for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes. This error condition corresponds to insufficient traffic reserved over DSP data bus for such a high DAC sampling frequency.

DAC IRT error condition is indicated via the ‘1’ state of *DAC_IRT_ERR* bit of *ERR_STAT_RG* register, which will remain in the ‘1’ state until reset by host DSP software. DAC IRT error condition can e used to generate host PIOX-16 interrupt request.

For more details about errors processing refer to subsection “Errors processing” inside “Data Acquisition Control” section earlier in this chapter.

Chapter 3. Installation and Configuration

This chapter includes instructions for installation of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM onto the mainboard of host *TORNADO* DSP system/controller and for connection to external equipment.

3.1 Installation onto *TORNADO* DSP System/Controller Mainboard

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM must be installed as standard PIOX-16 DCM onto host *TORNADO* DSP system/controller mainboard.

For installation of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM into PIOX-16 I/O expansion site of *TORNADO* DSP system/controller follow the recommendations below:

1. Switch off the power of host PC or of external power supply.
2. Remove *TORNADO* DSP system mainboard from PC slot, or *TORNADO* DSP controller from the target device compartment.
3. Ensure that two *TORNADO* on-board spacers for mounting PIOX-16 AD/DA DCM are installed into the corresponding holes on *TORNADO* mainboard (fig.3-1). If spacers are not installed, then install spacers, which are enclosed with *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* shipment package.

CAUTION

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM requires that short (10 mm long) spacers shall be used to install onto *TORNADO* PC plug-in DSP system mainboard, and long (12.5 mm long) spacers shall be used to install onto *TORNADO-E* stand-alone DSP controller mainboard.

4. Pick-up *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM from the shipment packaging and orient it parallel to *TORNADO* mainboard over PIOX-16 AD/DA DCM area. Safely plug-in on-board JP1 host PIOX-16 connector of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM into the corresponding 16-bit PIOX-16 site header of host *TORNADO* mainboard (fig.3-1a). In case host *TORNADO* mainboard provides on-board 32-bit PIOX interface site, then you have to plug *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM into the 16-bit PIOX-16 sub-connector of host PIOX interface site at host *TORNADO* mainboard (fig.3-1b).

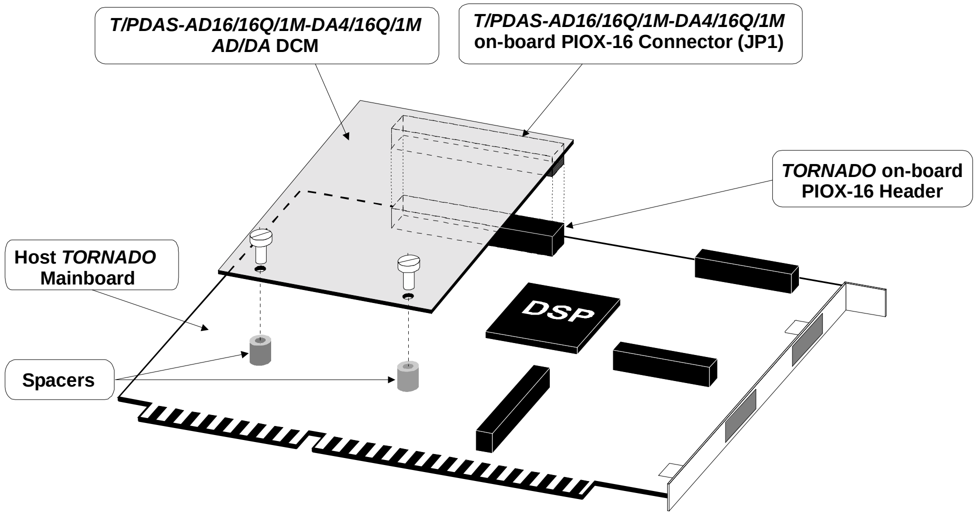
T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM
T/PDAS-AD16/16Q/1M-DA4/16Q/1M on-board PIOX-16 Connector (JP1)
TORNADO on-board PIOX-16 Header
Host TORNADO Mainboard
Spacers
DSP

Fig. 3-1a. Installation of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM into 16-bit PIOX-16 site of host *TORNADO* DSP system.

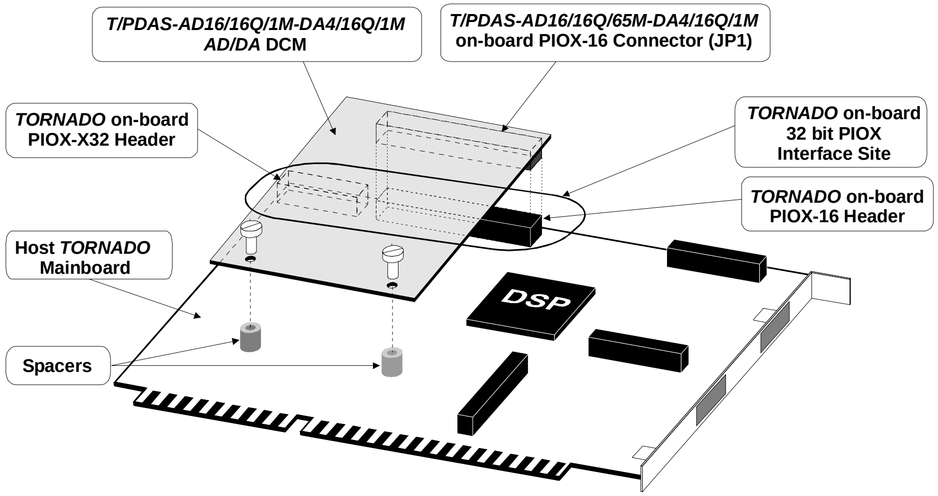
T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM
T/PDAS-AD16/16Q/65M-DA4/16Q/1M on-board PIOX-16 Connector (JP1)
TORNADO on-board PIOX-X32 Header
TORNADO on-board 32 bit PIOX Interface Site
TORNADO on-board PIOX-16 Header
Host TORNADO Mainboard
Spacers
DSP

Fig. 3-1b. Installation of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM into 32-bit PIOX site of host *TORNADO* DSP system.

4. Screw in *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM to the spacers at *TORNADO* mainboard.
5. Connect *T/X-XIOB/PDAS1M* external I/O board to *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM via two flat cables installed into on-board JP2 and JP3 connectors (refer to Appendix C for more details about *T/X-XIOB/PDAS1M* external I/O board for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM and how to connect it via two supplied flat cables).

6. In case *TORNADO* PC plug-in DSP system is used for installation of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, then install *TORNADO* mainboard into PC chassis slot and screw it to the rear panel of PC.
7. In case *TORNADO* PC plug-in DSP system is used for installation of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, then install and screw *T/X-XIOB/PDAS1M* external I/O board to the rear panel of PC.
8. Plug-in *T/X-SIOCS/PDAS1M* and *T/X-AIOCS/PDAS1M* external I/O cable (refer to Appendix C) sets to JP11 and JP10 external I/O connectors of *T/X-XIOB/PDAS1M* external I/O board.
9. Connect external I/O equipment and external analog I/O sources to *T/X-SIOCS/PDAS1M* and *T/X-AIOCS/PDAS1M* external I/O cable sets.
10. Switch on power of host PC or of external power supply.

3.2 Connection to external signal I/O equipment

Connection of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM to external analog I/O equipment is performed by means of on-board JP2 connector (fig.A-1) and by means of optional *T/X-XIOB/PDAS1M* external I/O board and *T/X-AIOCS/PDAS1M* external I/O cable set (Appendix C).

Connection of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM to external synchronization and digital I/O equipment is performed by means of on-board JP3 connector (fig.A-1) and by means of optional *T/X-XIOB/PDAS1M* external I/O board and *T/X-SIOCS/PDAS1M* external I/O cable set (Appendix C).

CAUTION

It is highly recommended to plug-in and unplug external I/O cables into/from on-board connectors of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM during the power is switched off.

The ground signal of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM has no galvanic isolation neither from host *TORNADO* DSP system/controller, nor from the PC ground and chassis, nor from external I/O peripherals and devices.

CAUTION

When connecting external analog I/O equipment to *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM via on-board JP2 connector, be aware that all analog inputs/outputs of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM are DC coupled. If required, external DC isolation capacitors shall be used.

Appendix A. On-board Connectors.

This Appendix includes a summary description for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* on-board connectors.

A.1 Board Layout

Board layout for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* on-board connectors is presented at fig.A-1.

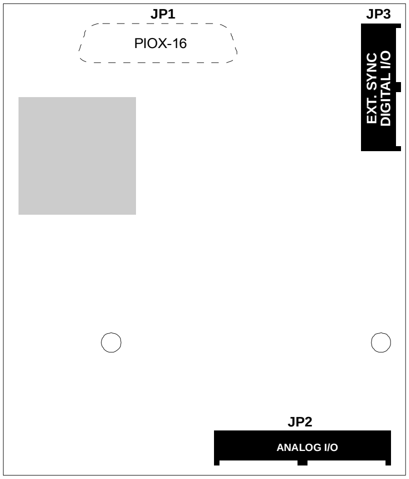

Fig.A-1. On-board connectors for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

Table A-1 contains the list of on-board connectors.

Table A-1. On-board connectors of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

connector ID	**Description**
JP1	Host PIOX-16 interface site male header. Pinout of JP1 host PIOX-16 connector is presented in Appendix B of this manual and in the user’s guide of host *TORNADO* DSP system/controller, which is used for installation of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.
JP2	External analog I/O connector, which includes AIN-0..15 analog inputs and AOUT-0..3 analog outputs. This connector can be either used for connection to external equipment or can connect to optional *T/X-XIOB/PDAS1M* external I/O board via flat cable. Refer to the corresponding subsection below and to Appendix C for more details.

JP3	External synchronization and I/O connector for connection to external digital equipment. This connector can be either used for connection to external equipment or can connect to optional *T/X-XIOB/PDAS1M* external I/O board via flat cable. Refer to the corresponding subsection below and to Appendix C for more details.

A.2 Pinout for JP2 external analog I/O connector

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM on-board JP2 external analog I/O connector has been designed for connection of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM to external analog equipment and comprises of signals for AIN-0..15 analog inputs and AOUT-0..3 analog outputs.

Pinout of JP2 analog I/O connector for *T/PDAS-AD16/16Q/1M-A4/16Q/1M* AD/DA DCM is presented at fig.A-2, whereas the signal description is presented in table A-2.

On-board JP2 connector is 40-pin guarded 2mm straight mail header from Samtec Inc (www.samtec.com). The mating plug p/n is TCSD-20-01-N, which is assumed to be used with the 16-wire 2mm flat ribbon cable.

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM has been designed to connect external synchronization devices via *T/X-XIOB/PDAS1M* external I/O board, which connects to *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* on-board JP2 connector via 40-pin 2 mm flat cable (refer to Appendix C for more details), which is included as standard with the shipment package.

However, in case customer application dos not require to use *T/X-XIOB/PDAS1M* external I/O board and requires application specific cable for connection directly to JP2 analog I/O connector at *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, then mating plugs are available from either MicroLAB Systems or Samtec Inc upon request.

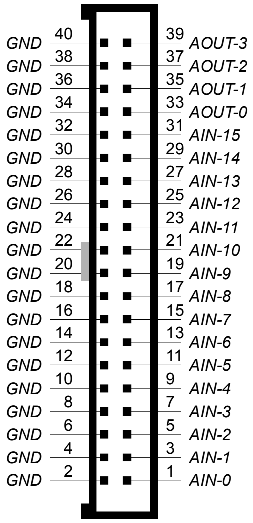

Fig. A-2. Pinout for JP2 external analog I/O connector of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

Table A-2. Signal description for JP2 external analog I/O connector of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

signal name	**signal type**	**Description**	**reference information**
AIN-0..15	AIN	Analog inputs.	sections 2-2 and 2-3
AOUT-0..3	AOUT	Analog outputs.	sections 2-2 and 2-3
GND	-	Ground.	

Notes: 1. Signal types: *AIN* – analog input; *AOUT* – analog output.

A.3 Pinout for JP3 external synchronization and I/O connector

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM on-board JP3 external synchronization and I/O connector has been designed for connection of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM to external digital synchronization and clock equipment, and for general purpose digital I/O via GPIO-0..3 pins. All signals of JP3 connector comply with 3v/5v TTL logic.

Pinout of JP3 external synchronization and I/O connector for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM is presented at fig.A-3, whereas the signal description is presented in table A-3.

On-board JP3 connector is 16-pin guarded 2mm straight mail header from Samtec Inc (www.samtec.com). The mating plug p/n is TCSD-08-01-N, which is assumed to be used with the 16-wire 2mm flat ribbon cable.

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM has been designed to connect external synchronization devices via *T/X-XIOB/PDAS1M* external I/O board, which connects to *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* on-board JP3 connector via 16-pin 2 mm flat cable (refer to Appendix C for more details), which is included as standard with the shipment package.

However, in case customer application dos not require to use *T/X-XIOB/PDAS1M* external I/O board and requires application specific cable for connection directly to JP3 external synchronization and I/O connector at *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, then mating plugs are available from either MicroLAB Systems or Samtec Inc upon request.

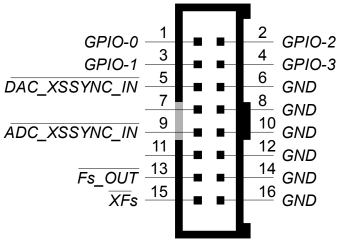

Fig. A-3. Pinout for JP3 external synchronization and I/O connector of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

Table A-3. Signal description for JP3 external synchronization and I/O connector of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

signal name	**signal type**	**Description**	**reference information**
XSYNC_ADC_IN *XSYNC_DAC_IN*	3v/5v TTL/IN	External active low start synchronization inputs for ADC and DAC data acquisition controllers correspondingly.	section 2-3
XFs	3v/5v TTL/IN	External active low sampling frequency input from external equipment.	section 2-3
Fs_OUT	3v/5v TTL/OUT	Active low sampling frequency output to external equipment.	section 2-3
GPIO-0 *GPIO-1* *GPIO-2* *GPIO-3*	3v/5v TTL/IO	General purpose programmable I/O.	section 2-2 tables 2-14a and 2-14b
GND	-	Ground.	

Notes: 1. Signal types: *TTL/IN* - TTL compatible digital input; *TTL/OUT* - TTL compatible digital output; *TTL/IO* - TTL compatible digital input/output.

Appendix B. PIOX-16 Interface Site

This appendix contains information about *TORNADO* PIOX-16 interface site specifications. This description is general to all *TORNADO* DSP systems/controllers, whereas different *TORNADO* boards with different DSP platforms may differ in the number of interrupts requests via PIOX-16 interface and in timer/IO pin specifications. Refer to your particular *TORNADO* user's guide for more details.

B.1 General Description

TORNADO architecture allows expansion of the on-board DSP I/O resources via on-board 16-bit parallel I/O expansion interface site (PIOX-16) (fig.B-1), which is designed to carry compatible AD/DA DCM (AD/DA DCM).

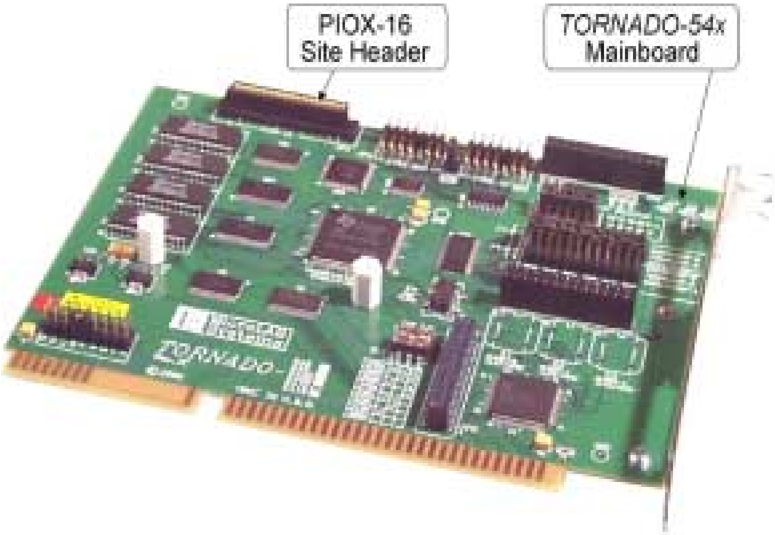

Fig.B-1. PIOX-16 site at *TORNADO-54x* board.

Some *TORNADO* boards (typically 32-bit *TORNADO* DSP systems for PC) provide 16-bit PIOX-16 site as a subset of on-board 32-bit PIOX interface site, whereas other *TORNADO* boards (typically *TORNADO* stand-alone DSP controllers and 16-bit *TORNADO* DSP systems for PC) provide PIOX-16 site only. Refer to your host *TORNADO* board user's guide for information about particular PIOX or PIOX-16 interface site installed.

Figure B-2 demonstrates installation of PIOX-16 AD/DA DCM into PIOX-16 site of host *TORNADO* DSP system/controller.

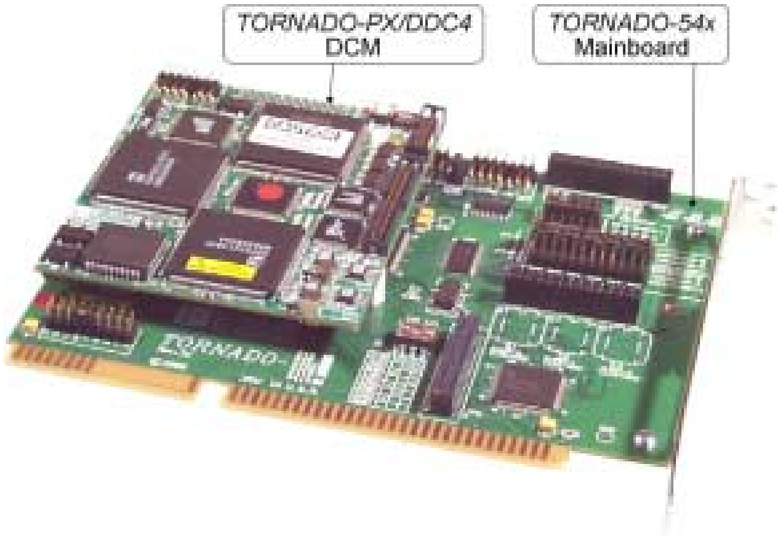

Fig.B-2. TORNADO-54x board with PIOX-16 AD/DA DCM installed.

B.2 PIOX-16 Interface Site Connector and Signals

TORNADO PIOX-16 interface site comprises of signals for SIO-0 and SIO-1 logical serial ports, DSP on-chip TM/XIO-0/1 timers/IO pins, three DSP interrupts, PIOX-16 reset control, and power ±5V/±12V power supplies.

PIOX-16 interface appears as the 64Kx16 sub-area of DSP external memory or I/O resources. PIOX-16 features 16-bit data transfer cycles.

PIOX-16 connector and signal description

PIOX-16 connector is a high-density DDK 50-pin DHB-series dual-row female connector with 0.05” pin pitch. Compatible PIOX-16 plugs for customer designed AD/DA DCM are available upon request from MicroLAB Systems.

PIOX-16 connector pinout is presented at fig B-3, whereas signal description for PIOX-16 connector is presented in table B-1.

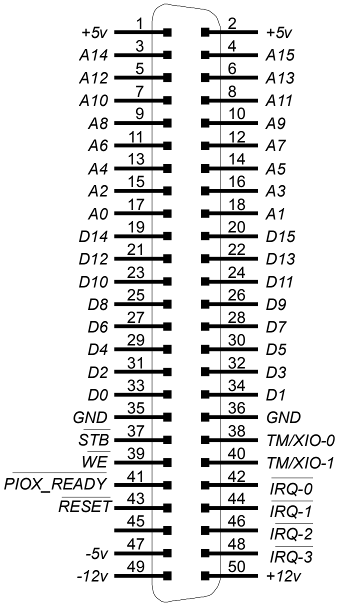

Fig.B-3. PIOX-16 connector pinout (top view).

Table B-1. PIOX-16 signal description.

Signal name	**signal type**	**description**
Address and Data Bus		
A0..A15	O	DSP address bus.
D0..D15	I/O	DSP data bus.
Data Transfer Control		
$\overline{STB}$	O	Active low PIOX-16 data transfer strobe.
$\overline{WE}$	O	Active low PIOX-16 write enable signal.
$\overline{PIOX\_READY}$	I	Active low PIOX-16 data ready acknowledge signal. This signal is generated by PIOX-16 AD/DA DCM in order to complete transmission cycle over PIOX-16 interface. This input has pull-up resistor.

DSP Timers, Reset and Interrupt Requests		
TM/XIO-0 *TM/XIO-1*	I/O/Z	These signals are typically connected to the DSP on-chip TIMER-0 and TIMER-1 I/O pins and can be software configured by DSP as either timer or I/O pin. However, in some *TORNADO* boards (for example *TORNADO-54x* board) these signals can be controlled by on-board I/O controller.
$\overline{RESET}$	O	Active low PIOX-16 reset signal. Some *TORNADO* boards (for example *TORNADO-3x* boards) wires this signal directly from the DSP reset signal, and PIOX-16 plugged-in AD/DA DCM reset is asserted simultaneously with *TORNADO* on-board DSP reset. However some *TORNADO* boards (for example *TORNADO-54x/6x* etc. boards) feature dedicated PIOX-16 site reset signal, which is controlled by *TORNADO* on-board DSP for better synchronization between the DSP software and PIOX-16 AD/DA DCM operation.
$\overline{IRQ-0}$ $\overline{IRQ-1}$ $\overline{IRQ-2}$ $\overline{IRQ-3}$	I	Active low external interrupt request lines for *TORNADO* on-board DSP. These lines are pulled up. Note, that IRQ-2 and IRQ-3 interrupt request input are not available for all *TORNADO* DSP systems/controllers (refer to your *TORNADO* DSP system/controller user's guide for details about on-board PIOX-16 interface).
Power Supplies		
GND		Ground.
+5v		+5v power (from ISA-bus).
+12v		+12v power (from ISA-bus).
-5v		-5v power (from ISA-bus).
-12v		-12v power (from ISA-bus).

Note: 1. Signal type is denoted as the following: *I* - input, *O* - output, *Z* - high impedance.

PIOX-16 site signal levels

Signal levels for PIOX-16 interface signals correspond to that for the CMOS/TTL signals with I_{OL}=2ma and I_{OH}=-0.3ma load currents.

CAUTION

Some *TORNADO* boards (*TORNADO-30/31/32L/542L*) provide PIOX-16 interface signal levels compatible with that for CMOS/TTL only, whereas other *TORNADO* boards (*TORNADO-33/54xx/6x/E6x/P6x/E3x/E6x*) provide PIOX-16 interface signal levels universal for both 3V TLL and standard TTL. Refer to documentation for your particular *TORNADO* board for information about PIOX-16 interface signal levels.

timing diagram for PIOX-16 data transmission cycle

Figure B-4 presents timing diagram for typical PIOX-16 data transmission cycle for *TORNADO-54x* DSP system. This data transfer timing is known as the industry standard MOTOROLA mode and assumes usage of data strobe signal and write enable signal.

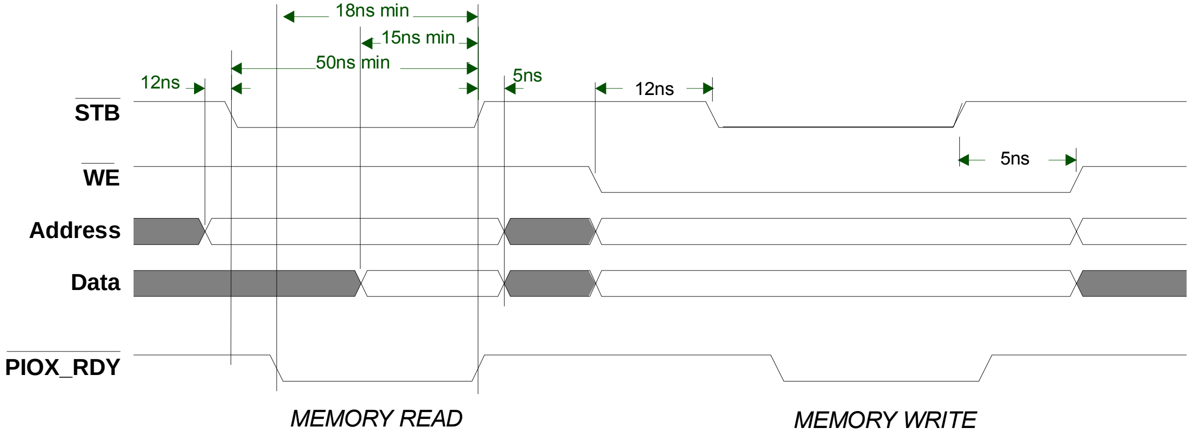

Fig.B-4. Timing diagram of PIOX-16 data transfer for *TORNADO-54x*.

Other *TORNADO* DSP systems/controllers (*TORNADO-3x/6x/P3x/P6x/E3x/E6x*, etc) provide similar timing for PIOX-16 data transmission cycles with the only differences applied to specific timing parameters. Refer to documentation for your particular *TORNADO* board for information about PIOX-16 timing specifications.

B.3 Physical Dimensions for PIOX-16 AD/DA DCM

Physical dimensions for PIOX-16 AD/DA DCM are presented at fig.B-5. This information is intended for those customers, who need to design custom PIOX-16 AD/DA DCM.

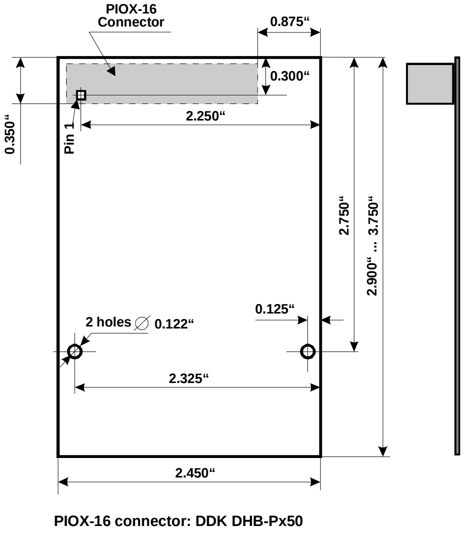

Fig.B-5. Physical dimensions for PIOX-16 AD/DA DCM.

Appendix C. External Cable Sets

This appendix contains information about *T/X-XIOB/PDAS1M* external I/O board and *T/X-SIOCS/PDAS1M* and *T/X-AIOCS/PDAS1M* external cable sets for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

C.1 Connection of external I/O devices to *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM

T/PDAS-AD16/16Q/1M-DA4/16Q/1M AD/DA DCM provides on-board JP2 external analog I/O connector and JP3 external synchronization and I/O connector (refer Appendix A) for connection to external synchronization and analog I/O devices.

Although external analog I/O, synchronization and digital I/O devices can connect directly to *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM on-board JP2 and JP3 connectors, however, for more convenience, it is recommended to connect *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM to external analog and digital I/O world by means of optional *T/X-XIOB/PDAS1M* external I/O board and two *T/X-SIOCS/PDAS1M* and *T/X-AIOCS/PDAS1M* external I/O cable sets (fig.C-1), which come standard with *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM shipment package.

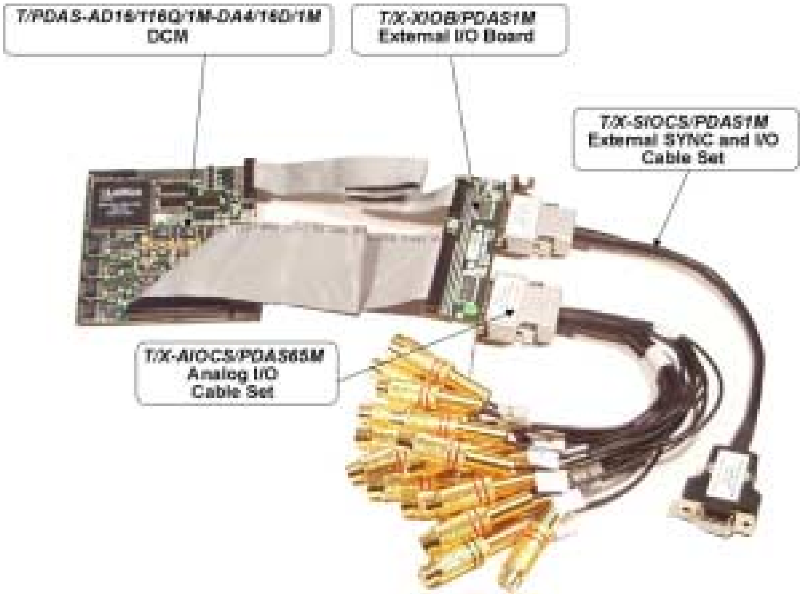
T/PDAS-AD16/116Q/1M-DA4/16D/1M DCM
T/X-XIOB/PDAS1M External I/O Board
T/X-SIOCS/PDAS1M External SYNC and I/O Cable Set
T/X-AIOCS/PDAS65M Analog I/O Cable Set

Fig. C-1. *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM with *T/X-XIOB/PDAS1M* external I/O board and *T/X-SIOCS/PDAS1M* and *T/X-AIOCS/PDAS1M* external cable sets.

Optional *T/X-XIOB/PDAS1M* external I/O board can be installed at the rear panel of PC chassis and converts *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM on-board JP2 and JP3 connectors into two remote external analog I/O and synchronization with digital I/O connectors, which are used for connection to *T/X-SIOCS/PDAS1M* and *T/X-AIOCS/PDAS1M* external cable sets.

Both *T/X-SIOCS/PDAS1M* and *T/X-AIOCS/PDAS1M* external cable sets provide industry standard front-end connectors for connection to external synchronization and digital I/O devices and to analog I/O devices correspondingly.

C.2 *T/X-XIOB/PDAS1M* External I/O Board

T/X-XIOB/PDAS1M external I/O board (fig.C-2) comes standard with *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM and must be used for connection to external devices via *T/X-SIOCS/PDAS1M* and *T/X-AIOCS/PDAS1M* external cable sets, which provide industry standard end-user connectors and connects to *T/X-XIOB/PDAS1M* external I/O board via rear panel of PC chassis.

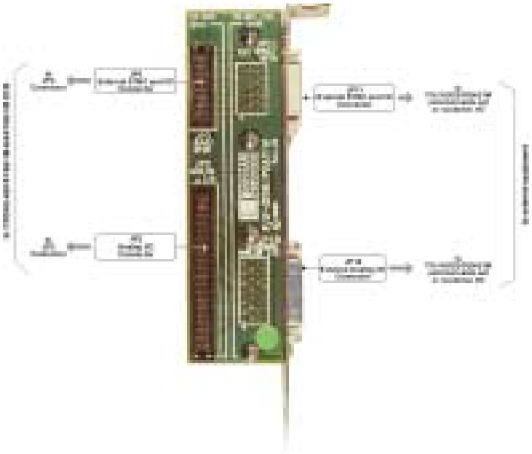

Fig. C-2. *T/X-XIOB/PDAS1M* external I/O board.

installation

T/X-XIOB/PDAS1M external I/O board either installs in a separate slot at the rear panel of PC chassis, or can be installed directly onto the rear mounting bracket of *TORNADO* PC plug-in DSP systems in order to save space inside PC chassis compartment.

schematic diagram for T/X-XIOB/PDAS1M external I/O connector board

T/X-XIOB/PDAS1M external I/O board comprises of the following on-board connectors:

- Input JP2 guarded 2mm 40-pin straight male header, which must connect to *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM on-board JP2 analog I/O connector via 8" 2mm flat ribbon cable
- Input JP3 guarded 2mm 16-pin straight male header, which must connect to *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM on-board JP3 external synchronization and I/O connector via 8" 2mm flat ribbon cable.
- Output JP10 external analog I/O connector, which comprises of analog I/O signals from on-board JP2 connector and which allows connection to external analog I/O devices via *T/X-AIOCS/PDAS1M* external cable set. *T/X-XIOB/PDAS1M* on-board JP10 connector is 26-pin half pitch DHA-RA26 series receptacles from Fujikura-DDK Ltd (www.ddkconnectors.com). In case customer needs to

design his own application specific cable for connection to JP10 connector instead of using provided *T/X-AIOCS/PDAS1M* external cable set, then the mating plug for JP10 connector (plug p/n DHA-PC26-2G and metal hood p/n DHA-HPA26) is available upon request from MicroLAB Systems.

- Output JP11 external I/O connector, which comprises of the signals from on-board JP3 connector and which allow connection of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM to external synchronization and I/O devices via *T/X-SIOCS/PDAS1M* external cable set. *T/X-XIOB/PDAS1M* on-board JP11 connector is 20-pin half pitch DHA-RA20 series receptacles from Fujikura-DDK Ltd (www.ddkconnectors.com). In case customer needs to design his own application specific cable for connection to JP11 connector instead of using provided *T/X-SIOCS/PDAS1M* external cable set, then the mating plug for JP11 connector (plug p/n DHA-PC20-2G and metal hood p/n DHA-HPA20) is available upon request from MicroLAB Systems.

Fig.C-3 shows generic schematic diagram of *T/X-XIOB/PDAS1M* external I/O board for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. Signals description for on-board JP2 external analog I/O and JP3 external synchronization and digital I/O connectors for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM are provided in tables A-2 and A-3 from Appendix A.

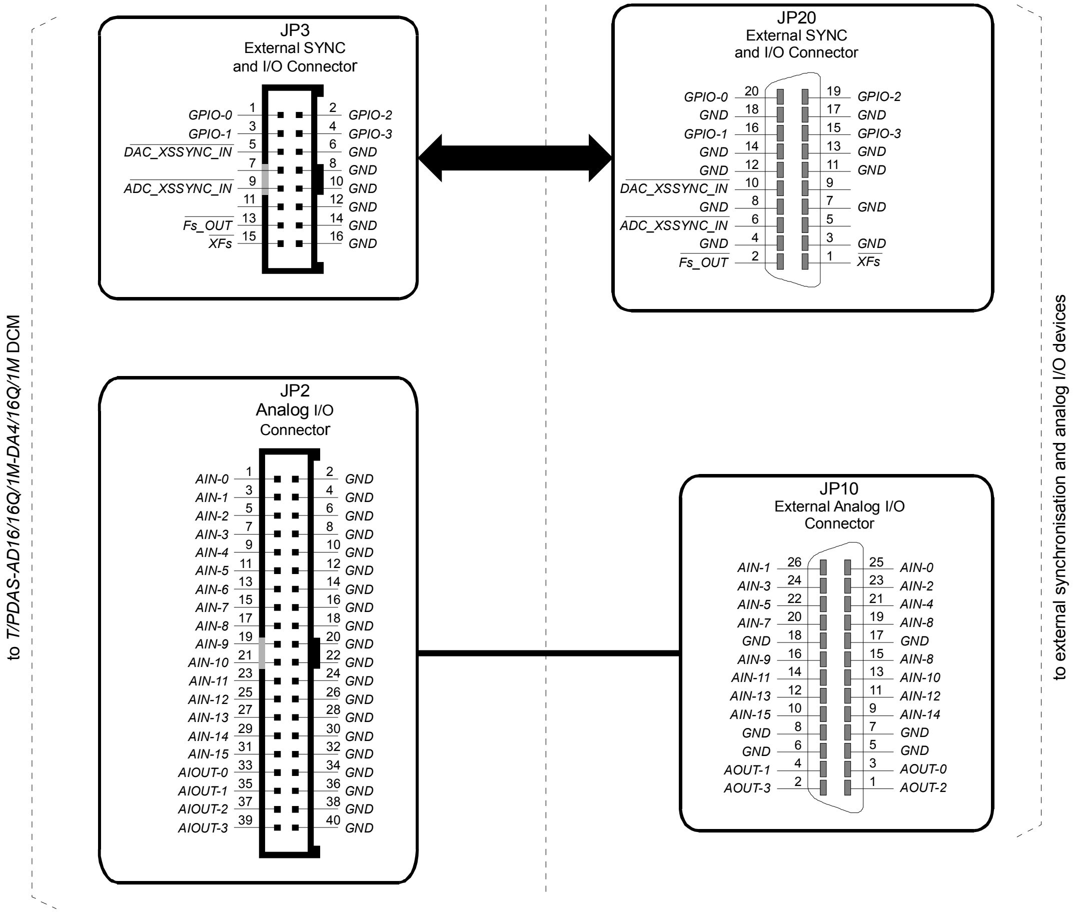

Fig.C-3. Schematic diagram of *T/X-XIOB/PDAS1M* external I/O board.

C.3 *T/X-SIOCS/PDAS1M* and *T/X-AIOCS/PDAS1M* External I/O Cable Sets

T/X-SIOCS/PDAS1M and *T/X-AIOCS/PDAS1M* external cable sets (fig.C-4) come standard with *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM and shall be used for connection of *T/X-XIOB/PDAS1M* external I/O board to external synchronization and digital I/O and analog I/O devices correspondingly using industry standard front-end connectors.

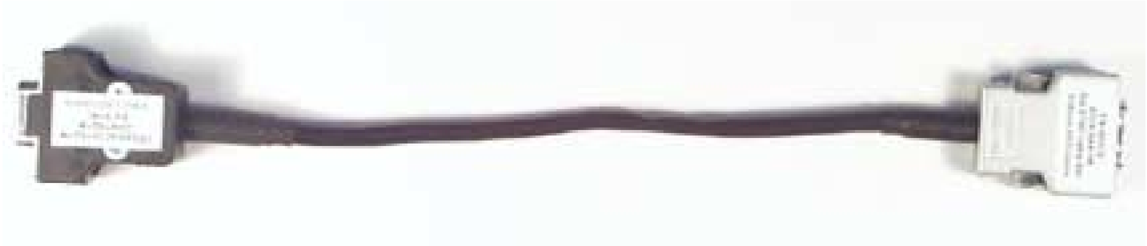

Fig. C-4a. *T/X-SIOCS/PDAS1M* external I/O cable set.

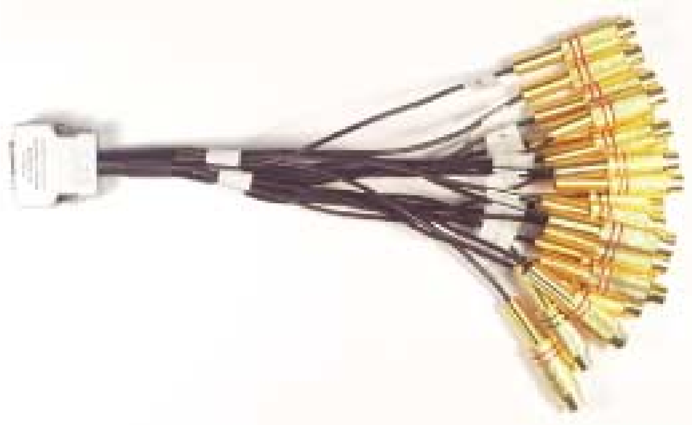

Fig. C-4b. *T/X-AIOCS/PDAS1M* external I/O cable set.

T/X-SIOCS/PDAS1M and *T/X-AIOCS/PDAS1M* external cable sets for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM plugs into JP11 and JP10 on-board connectors correspondingly of *T/X-XIOB/PDAS1M* external I/O board for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM (refer to fig.C-1 and section "*T/X-XIOB/PDAS1M* external I/O board" earlier in this appendix) and shall be used for connection to external devices via industry standard front-end connectors.

schematic diagram for T/X-SIOCS/PDAS1M external cable set

T/X-SIOCS/PDAS1M external cable set converts external synchronization and general purpose I/O signal from 20-pin JP11 on-board connector of *T/X-XIOB/PDAS1M* external I/O board for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM into industry standard DBH-15 high-density 15-pin female connector.

T/X-SIOCS/PDAS1M external cable set is a recommended selection for connection of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM to external synchronization and digital I/O devices via *T/X-XIOB/PDAS1M* external I/O board.

Schematic diagram of *T/X-SIOCS/PDAS1M* external cable set is presented at figure C-5. Signal description is provided in table A-3 of Appendix A for JP3 external synchronization and digital I/O connector of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM.

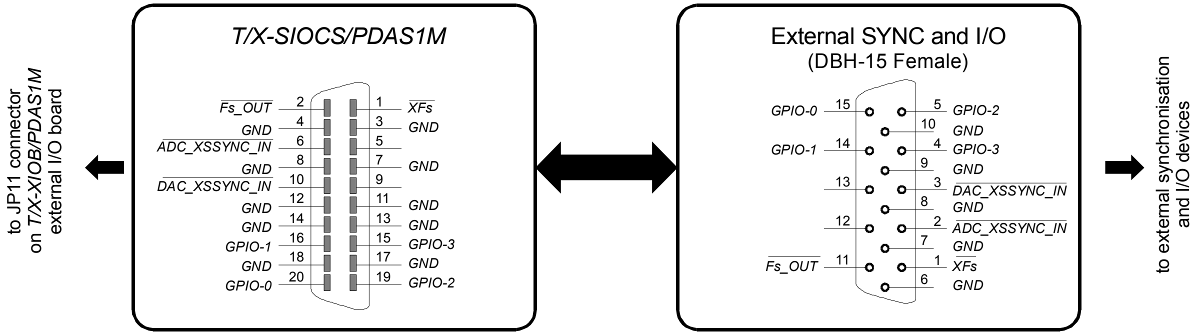

Fig.C-5. Schematic diagram of *T/X-SIOCS/PDAS1M* external cable set.

schematic diagram for T/X-AIOCS/PDAS1M external cable set

T/X-AIOCS/PDAS1M external cable set converts analog I/O signals from 26-pin JP10 on-board connector of *T/X-XIOB/PDAS1M* external I/O board for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM into twelve industry standard female RCA jacks (fig.C-6).

T/X-AIOCS/PDAS1M external cable set is a recommended selection for connection of *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM to external analog I/O devices via *T/X-XIOB/PDAS1M* external I/O board.

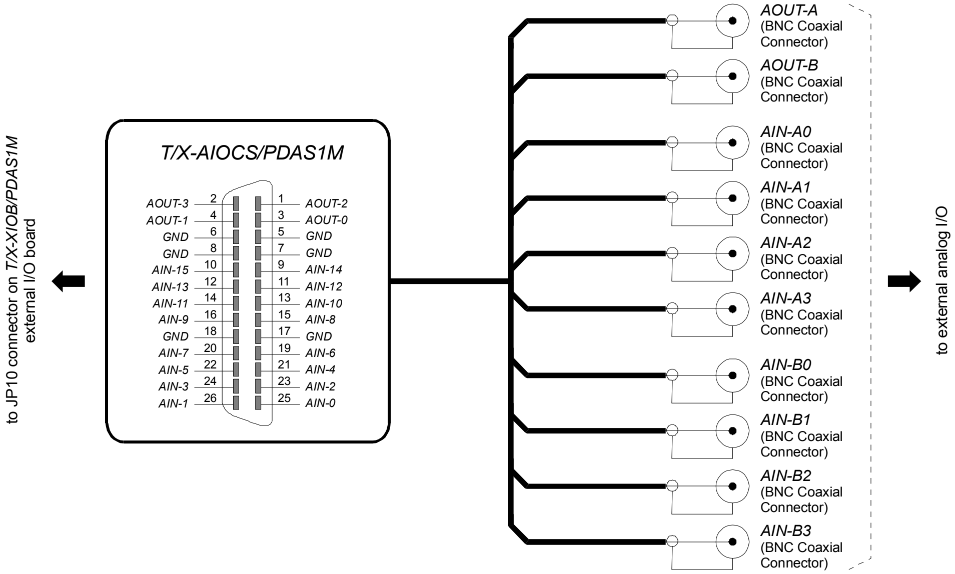
to JP10 connector on T/X-XIOB/PDAS1M external I/O board
T/X-AIOCS/PDAS1M
AOUT-3 2
AOUT-1 4
GND 6
GND 8
AIN-15 10
AIN-13 12
AIN-11 14
AIN-9 16
GND 18
AIN-7 20
AIN-5 22
AIN-3 24
AIN-1 26
1 AOUT-2
3 AOUT-0
5 GND
7 GND
9 AIN-14
11 AIN-12
13 AIN-10
15 AIN-8
17 GND
19 AIN-6
21 AIN-4
23 AIN-2
25 AIN-0
AOUT-A (BNC Coaxial Connector)
AOUT-B (BNC Coaxial Connector)
AIN-A0 (BNC Coaxial Connector)
AIN-A1 (BNC Coaxial Connector)
AIN-A2 (BNC Coaxial Connector)
AIN-A3 (BNC Coaxial Connector)
AIN-B0 (BNC Coaxial Connector)
AIN-B1 (BNC Coaxial Connector)
AIN-B2 (BNC Coaxial Connector)
AIN-B3 (BNC Coaxial Connector)
to external analog I/O

Fig.C-6. Schematic diagram of *T/X-AIOCS/PDAS1M* external cable set.

Appendix D. Glossary of Terms.

This Glossary contains definition for terms and other synchronism used along in this databook.

A

Acquisition

Acquisition (sampling) of analog input signal by sampling-hold circuit of ADC converter chip, which is typically implemented as the charge of internal ADC on-chip capacitor. Acquisition of analog input signal is an inevitable part of A/D converter operation and is performed before actual A/D conversion takes place in order to guarantee that converted signal will be not changed during A/D conversion. Refer to section 2.3 for more details.

ADC

Analog-to-digital converter chip. *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* is AD/DA DCM uses ultra-high accurate and high-performance ADC chips. Refer to sections 2.1 and 2.3 for more details.

ADC-0, ADC-1, ADC-2, ADC-3

On-board four A/D converter channels, which comprise of four analog inputs, four analog input amplifiers, analog input multiplexer and A/D converter chip. Refer to sections 2.1 and 2.3 for more details.

ADC-ASYNC-DAQ

Asynchronous data acquisition mode for ADC data acquisition controller. Refer to sections 2.2 and 2.3 for more details.

ADC-SYNC-FIFO-OPM-DAQ

Synchronous data acquisition mode for ADC data acquisition controller, which uses ADC FIFO to temporary store real-time ADC output data and terminates itself normally of software selected ADC FIFO flag condition. Refer to sections 2.2 and 2.3 for more details.

ADC-SYNC-FIFO-PTM-DAQ

Synchronous data acquisition mode for ADC data acquisition controller, which uses ADC FIFO to temporary store real-time ADC output data and never terminates itself. Refer to sections 2.2 and 2.3 for more details.

ADC-SYNC-PX-DAQ

Synchronous data acquisition mode for ADC data acquisition controller, which allows direct read of real-time ADC output data via host PIOX-16 interface. Refer to sections 2.2 and 2.3 for more details.

ADC_CNF_RG

ADC channels configuration register, which is used to select particular ADC channels, which will be involved into ADC data acquisition process and ADC IRT data transfer. Refer to sections 2.2, 2.3 and 2.4 for more details.

ADC_DAQ_RESET_RG

Write-only register, which must be used for reset ADC FIFO, ADC data acquisition controller, ADC IRT controller and ADC data acquisition errors. Refer to sections 2.2 and 2.3 for more details.

ADC_FIFO_SIN_RG

Write-only register, which must be used to program offset values for PAF/PAE flags of ADC FIFO. Refer to sections 2.2 and 2.3 for more details.

ADC_FIFO_DATA_RG

Read-only ADC FIFO data register. Refer to sections 2.2 and 2.3 for more details.

ADC_FIFO_DATA_MSW_RG

Read-only most significant word from ADC FIFO, which contains end-of-scan indicator for ADC FIFO data. Refer to sections 2.2 and 2.3 for more details.

ADC_IMUX_CNF1_RG, ADC_IMUX2_CNF_RG

ADC input multiplexers configuration registers. Refer to sections 2.2 and 2.3 for more details.

ADC_IRT_DATA_RG

ADC output data register, which must be used as the source input data register for host DSP on-chip DMA controller synchronized by ADC IRT controller in order to download real-time ADC output data via host PIOX-16 interface for all enabled ADC channels for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes. Refer to sections 2.2, 2.3 and 2.4 for more details.

ADC FIFO

On-board ADC FIFO, which is used to temporary store real-time ADC output data prior they will be read by host DSP. Refer to sections 2.1 and 2.3 for more details.

ADC IMUX

Four software configured either 4:1/SE or 2:1/DIFF ADC analog input multiplexers for each of ADC channels, which are used to multiplex AIN-0..15 analog inputs to the inputs of on-board four A/D converter chips. Refer to sections 2.1, 2.2 and 2.3 for more details.

ADC Data Acquisition Controller

A part of on-board DAQCU unit, which is used to control ADC data acquisition process. Refer to sections 2.1 and 2.3 for more details.

ADC IRT Controller

ADC interrupt retriggable transmission (IRT) controller for generation of retriggable source synchronization events to host DSP on-chip DMA controller via host PIOX-16 interrupt request inputs in order to synchronize transfer of real-time ADC output data and ADC FIFO output data to host DSP environment using host DSP on-chip DMA controller. Refer to sections 2.2, 2.3 and 2.4 for more details.

ADC sampling cycle

A part of ADC asynchronous data acquisition process, which performs analog signal acquisition and A/D conversion for selected analog inputs for all ADC channels. Typically, start of ADC sampling cycle does not provide guaranteed time jitter parameter. Refer to section 2.3 for more details.

ADC sampling packet

A part of ADC synchronous data acquisition process, which can comprise of several ADC scan cycles and performs analog signal acquisition and A/D conversion for pre-defined analog inputs for all ADC channels. ADC sampling packets feature accurate timing specification. Refer to section 2.3 for more details.

ADC scan cycle

A part of ADC sampling packet during ADC synchronous data acquisition process, which performs analog signal acquisition and A/D conversion for pre-selected analog inputs for all ADC channels. ADC scan cycles feature accurate timing specifications. Refer to section 2.3 for more details.

ADC0_DATA_RG, ADC1_DATA_RG, ADC2_DATA_RG, ADC3_DATA_RG

ADC output data registers for each of on-board ADC channels, which are used for direct read of real-time ADC output data via host PIOX-16 interface for *ADC-ASYNC-DAQ* and *ADC-SYNC-PX-DAQ* data acquisition modes. Refer to sections 2.2 and 2.3 for more details.

ADDA_DAQ_RESET_RG

Write-only register, which must be used for simultaneous reset of ADC/DAC FIFO, ADC/DAC data acquisition controllers, ADC/DAC IRT controllers and data acquisition errors. Refer to sections 2.2 and 2.3 for more details.

B

C

D

DAC

Digital-to-analog converter chip. *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* is AD/DA DCM uses ultra-high accurate and high-performance DAC chips with double-buffer internal architecture. Refer to sections 2.1 and 2.3 for more details.

DAC-0, DAC-1, DAC-2, DAC-3

On-board four D/A converter channels, which comprise of D/A converter chip, analog output amplifier, and analog output. Refer to sections 2.1 and 2.3 for more details.

DAC-ASYNC-DAQ

Asynchronous data acquisition mode for DAC data acquisition controller. Refer to sections 2.2 and 2.3 for more details.

DAC-SYNC-FIFO-OPM-DAQ

Synchronous data acquisition mode for DAC data acquisition controller, which uses DAC FIFO to temporary store real-time DAC input data and terminates itself normally of software selected DAC FIFO flag condition. Refer to sections 2.2 and 2.3 for more details.

DAC-SYNC-FIFO-PTM-DAQ

Synchronous data acquisition mode for DAC data acquisition controller, which uses DAC FIFO to temporary store real-time DAC input data and never terminates itself. Refer to sections 2.2 and 2.3 for more details.

DAC-SYNC-PX-DAQ

Synchronous data acquisition mode for DAC data acquisition controller, which allows direct write of real-time DAC input data via host PIOX-16 interface. Refer to sections 2.2 and 2.3 for more details.

DAC_CNF_RG

DAC channels configuration register, which is used to select particular DAC channels, which will be involved into DAC data acquisition process and DAC IRT data transfer. Refer to sections 2.2, 2.3 and 2.4 for more details.

DAC_DAQ_RESET_RG

Write-only register, which must be used for reset DAC FIFO, DAC data acquisition controller, DAC IRT controller and DAC data acquisition errors. Refer to sections 2.2 and 2.3 for more details.

DAC_FIFO_DATA_RG

Write-only DAC FIFO data register. Refer to section 2.2 for more details.

DAC_FIFO_SIN_RG

Write-only register, which must be used to program offset values for PAF/PAE flags of DAC FIFO. Refer to sections 2.2 and 2.3 for more details.

DAC_IRT_DATA_RG

DAC input data register, which must be used as the destination output data register for host DSP on-chip DMA controller synchronized by DAC IRT controller in order to upload real-time DAC input data via host PIOX-16 interface for all enabled DAC channels for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes. Refer to sections 2.2, 2.3 and 2.4 for more details.

DAC FIFO

On-board DAC FIFO, which is used to temporary store real-time DAC input data from host DSP prior they will be written to DAC input registers. Refer to sections 2.1 and 2.3 for more details.

DAC Data Acquisition Controller

A part of on-board DAQCU unit, which is used to control DAC data acquisition process. Refer to sections 2.1 and 2.3 for more details.

DAC IRT Controller

DAC interrupt retriggable transmission (IRT) controller for generation of retriggable destination synchronization events to host DSP on-chip DMA controller via host PIOX-16 interrupt request inputs in order to synchronize transfer of real-time DAC input data and DAC FIFO input data from host DSP environment using host DSP on-chip DMA controller. Refer to sections 2.2, 2.3 and 2.4 for more details.

DAC sampling cycle

A short part of DAC asynchronous and synchronous data acquisition processes, which performs load of DAC output register in order to update DAC analog outputs for all DAC channels.. Refer to section 2.3 for more details.

DAC0_DATA_RG, DAC1_DATA_RG, DAC2_DATA_RG, DAC3_DATA_RG

DAC input data registers for each of on-board DAC channels, which are used for direct write of real-time DAC input data via host PIOX-16 interface for *DAC-ASYNC-DAQ* and *DAC-SYNC-PX-DAQ* data acquisition modes. Refer to sections 2.2 and 2.3 for more details.

DAQ

Data acquisition.

DAQ_CNTR1_RG

Data acquisition control register, which must be used to configure ADC/DAC data acquisition modes and data acquisition options for ADC/DAC data acquisition controllers. Refer to sections 2.2 and 2.3 for more details.

DAQ_CNTR2_RG

Data acquisition control register, which must be used to control and get status of ADC/DAC synchronous data acquisition processes. Refer to sections 2.2 and 2.3 for more details.

DAQ_CNTR3_RG

Data acquisition control register, which must be used to select termination flags for *ADC-SYNC-FIFO-OPM-DAQ* and *DAC-SYNC-FIFO-OPM-DAQ* data acquisition modes, and to select start/termination flags for ADC IRT and DAC IRT controllers. Refer to sections 2.2, 2.3 and 2.4 for more details.

DAQ_SYNC_RG

Data acquisition control register, which must be used to select hardware defined sampling frequency source, and start synchronization modes and timing options for ADC/DAC data acquisition controllers. Refer to sections 2.2 and 2.3 for more details.

DAQCU

Data acquisition control unit, which performs control of ADC/DAC data acquisition processes. Refer to sections 2.1, 2.2 and 2.3 for more details.

DCM

Daughter-card module. *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* is AD/DA DCM, which plugs into PIOX-16 interface site of host *TORNADO* DSP system/controller.

DMA

Direct memory access controller, which is the part of TMS320 digital signal processor. For more details, refer to original TI TMS320 DSP documentation.

DSP

Digital Signal Processor.

E

ERR_CLR_RG

Write-only register, which must be used to clear data acquisition errors. Refer to sections 2.2 and 2.3 for more details.

ERR_STAT_RG

Read-only register, which must be used to read status of data acquisition errors. Refer to sections 2.2 and 2.3 for more details.

F

FIFO

On-board high density First-In-First-Output device, which is used to store ADC/DAC data. Refer to sections 2.1, 2.2 and 2.3 for more details.

FIFO_STAT_RG

Read-only register, which must be used to read status of ADC/DAC FIFO flags. Refer to sections 2.2 and 2.3 for more details.

G

GPIO-0/1/2/3

General purpose digital I/O bits, which can be used for control of external peripherals. *GPIO-0/1/2/3* are controlled via *GPIO_DIR_RG* and *GPIO_DATA_RG* registers. Refer to section 2.2 for more details.

GPIO_DIR_RG, GPIO_DATA_RG

Registers, which are used to set direction and read/write data to the *GPIO-0..3* digital I/O pins. Refer to section 2.2 for more details.

H

Host PIOX-16 interface

T/PDAS-AD16/16Q/1M-DA4/16Q/1M on-board host PIOX-16 interface, which is used to install onto host *TORNADO* DSP system/controller and for communication with host DSP. Refer to section 2.5 and Appendix B for more details.

HIRQ0_SEL_RG, HIRQ1_SEL_RG, HIRQ2_SEL_RG, HIRQ3_SEL_RG,

Registers, which shall be used to enable and to select a particular interrupt request source for each of host PIOX-16 interrupt request lines IRQ-0..3. Refer to section 2.2 for more details.

I

IRQ-0, IRQ-1, IRQ-2, IRQ-3

Host PIOX-16 interface interrupt request inputs, which are used to send interrupt request to host DSP of host *TORNADO* DSP system/controller from *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. Refer to section 2.5 and Appendix B for more details.

IRT

Interrupt retriggable transmission (IRT). Refer to ADC IRT controller or DAC IRT controller and to sections 2.2, 2.3. and 2.4 for more details.

J

K

L

M

N

O

P

PFG

On-board programmable sampling frequency generator, which can be used to set sampling frequency for on-board ADC and DAC data acquisition controllers. Refer to sections 2.2 and 2.3 for more details.

PFG_CNTR1_RG, PFG_CNTR2_RG, PFG_CNTR3_RG

Configuration registers for on0board PFG, which shall be used to set PFG output frequency. Refer to sections 2.2 and 2.3 for more details.

PIOX

32-bit Parallel I/O eXpansion interface site for compatible daughter-card modules (AD/DA DCM) at *TORNADO* PC plug-in DSP systems. PIOX comprises of 16-bit PIOX-16 interface site and PIOX-X32 32-bit extension interface site. Refer to sections 2.6 and 3.1, and Appendix B for more details.

PIOX-16

16-bit Parallel I/O eXpansion interface site for compatible daughter-card modules (AD/DA DCM) at *TORNADO* PC plug-in DSP systems and controllers. *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM plugs into PIOX-16 interface site on host *TORNADO* DSP system/controller. Refer to sections 2.6 and 3.1, and Appendix B for more details.

Q

R

S

Sampling

See 'ACQUISITION'.

Sampling frequency

The frequency at which acquisition (sampling) of analog signals is performed. For *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM, sampling frequency can be sourced form on-board PFG, either of two PIOX-16 timer/IO pins (*TM/XIO-0/1*), or external sampling frequency source. Refer to section 2.2 and 2.3 and Appendix B for more details.

T

T/X-XIOB/PDAS1M

External I/O board for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. Refer to Appendix C for more details.

T/X-AIOCS/PDAS1M

External analog I/O cable set for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. Refer to Appendix C for more details.

T/X-SIOCS/PDAS1M

External synchronization and digital I/O cable set for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. Refer to Appendix C for more details.

TM/XIO-0, TM/XIO-1

Timer/IO pins of PIOX/PIOX-16 interface of *TORNADO* DSP systems and controllers, which can be used as the hardware defined sampling frequency source for *T/PDAS-AD16/16Q/1M-DA4/16Q/1M* AD/DA DCM. Refer to section 2.2 and 2.3 and Appendix B for more details.

U

V

W

X

XIM_ERR_RG

Expansion error interrupt register, which must be used to set interrupt enable masks for data acquisition errors. Refer to section 2.2 for more details.

Y

Z