ENBG

GET – Front End:

AsAd Technical Specification

User Requirements Document

Issue: 1
Revision: 0

Reference: -
Created: October 2009
Last modified: -

Prepared By: Jérôme PIBERNAT in collaboration with:
Jean-Louis Pedroza
Abdelkader Rebii
Patrick Hellmuth
Arnaud Tizon
Cedric Huss

This document has been prepared using the CERN PSS-05 Templates. The CERN PSS-05 Templates have been prepared by the Information and Programming Technology Group, ECP Division, CERN (The European Laboratory for Particle Physics) and conform to the PSS-05 Software Engineering Standards (ISBN 0-13-106568-8) defined by ESA (European Space Agency) BSSC (Board for Software Standardization and Control).

Abstract

This document describes a first level of specification for AsAd (ASIC-ADC board) which constitutes the GET-system front-end. The document is divided into three parts, the first part describes each AsAd functional blocks, the second part describes the AsAd interfaces, the third part relates to AsAd in its physical context. The purpose of the document is to provide the most complete, accurate, and up-to-date list of specifications for the system being designed.

Document Status Sheet

Table 1 Document Status Sheet

1. Document Title: AsAd technical specification - System and User Requirements Document			
2. Document Reference Number: -			
3. Issue	**4. Revision**	**5. Date**	**6. Reason for change**
1	0	October 2009	Creation

Document Change Record

Table 2 Document Change Record

Document Change Record		**DCR No.**	
		Date	
		Originator	
		Approved By	
1. Document Title			
2. Document Reference Number		[Document Reference Number]	
3. Document Issue / Revision Number		[Document Issue]/[Document Revision Number]	
4. Page	**5. Paragraph**	**6. Reason for Change**	

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose of the document

The purpose of this document is to define the most complete and accurate set of constraints and requirements to drive the design of AsAd starting from a functional point of view to reach a structural approach of this GET subsystem.

1.2 Scope of the system

AsAd (**As**ic-**Ad**c) is the part of **GET** (**G**eneral **E**lectronics for **T**PCs) that processes the analogue TPC's outgoing signals in order to make their informative contents digitally processable by the others GET subsystems represented on figure 1.

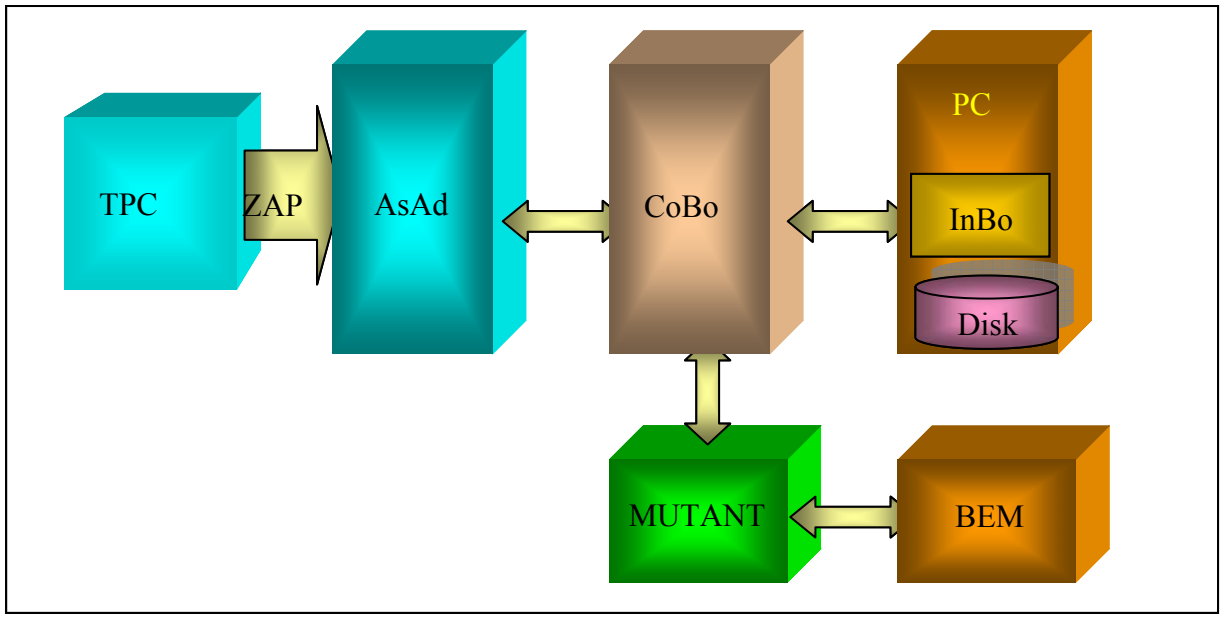


Figure 1: GET Synoptic

The system being discussed in this document is presented as a functional diagram in the figure 2. It consists of:

- Eleven functional blocks
  - Two main blocks:
    - An **A**SIC for **GET** block (blue)
    (⇨ TPC's analogue signals processing)
    - An **A**nalog to **D**igital **C**onversion block (indigo)
    (⇨ Digital conversion of the analogue processing result)

- Three AGET-dedicated secondary blocks:
  - A **S**witched **C**apacitors **A**rray Management block (light green)
  (⇨ Real time signal processing management)
  - A **S**erial **C**ontrol block (green)
  (⇨Delayed time signal processing control)
  - A Test and Calibration Management block (cyan)
  (⇨Analogue processing result and digital conversion checks)
- Six AsAd-dedicated secondary blocks:
  - A Power Supply block (brown)
  (⇨AsAd's local power sources and sources management)
  - A Monitoring block (red)
  (⇨AsAd's operating environment check)
  - A Slow Control Management block (orange)
  (⇨AsAd's own resources control)
  - An Input Management block (purple)
  (⇨AsAd's inputs configuration)
  - An Inspection Management block (grey)
  (⇨AsAd's operations check)
  - An Identification block (black)
  (⇨AsAd's location in the whole GET system)

- Five interfaces
  - Four user interfaces:
    - The AsAd-CoBo Interface
    - The AsAd-Power Supply Unit Interface
    - The AsAd-External Instruments Interface
    - The AsAd-ZAP Interface
  - One expert interface:
    - The AsAd-JTAG Interface

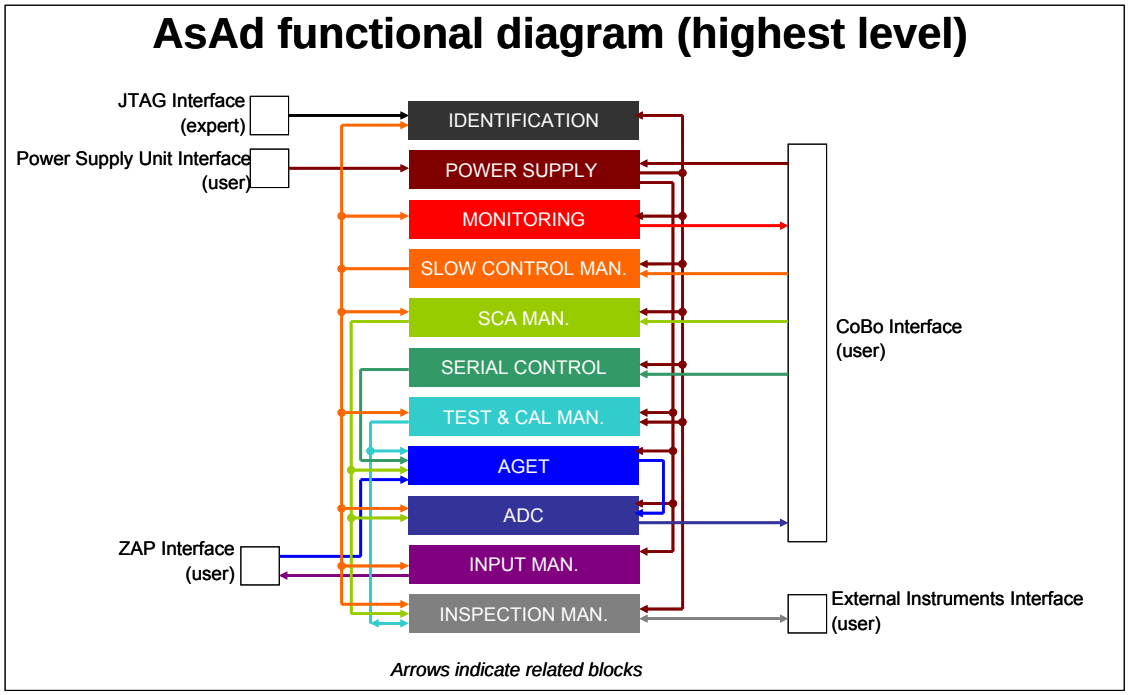


Figure 2: AsAd highest level functional diagram

All functional blocks are described in the first part of the document whereas the second part concerns the interfaces details. The third part gives an overview of AsAd as a physical GET module located into its operating context.

1.3 Acronyms and abbreviations

CENBG
Centre d'Etudes Nucléaires de Bordeaux-Gradignan

GET
General Electronics for TPC's

TPC
Time Projection Chamber

AsAd
ASIC-ADC board or module (the system introduced in this document)

ASIC
Application Specific Integrated Circuit

AGET
ASIC for GET

SCA
Switched Capacitors Array (an AGET subsystem)

ADC
Analog to Digital Conversion or Converter

I/O
Inputs/Outputs

CoBo
Concentration Board

1.4 References

External documents referenced in this document can directly be accessed by following the hypertext link inserted into the text

2 PART ONE: AsAd Functional Blocks

2.1 ASIC for GET

AGET is basically a configurable Application Specific Integrated Circuit able to process simultaneously 64 random analogue signals by amplifying, filtering and sampling them, as well as discriminating them by comparing their magnitude to a user-defined threshold level.
The analogue samples are stored into a Switched Capacitor Array (SCA) and the number of coincident discriminated input signals has to be concurrently monitored in order to trigger the sampling process stop (Multiplicity detection).
Once this process interrupted, the stored samples have to be sequentially extracted from the SCA in order to convert their analogue values into digital data.
Further details on AGET features and processes can be found here

AsAd is able to host up to four AGETs and to provide them with:
- their required power supply and bias voltages (c.f. Power Supply section)
- their required means to be tested and calibrated (c.f. Test & Calibration Management)

AsAd receives from CoBo the parameters allowing AGET operations under any circumstance and is responsible to allocate them correctly. (c.f. Switch Capacitors Array management and Serial Control sections)

AsAd implements devices allowing the digital conversion of the multiplicity and the analogue samples stored into the SCA. It is also able to transfer the converted data to CoBo (c.f. Analogue to Digital Conversion section).

Figure 3 shows AGET as the AsAd main block with its interconnections to the others AsAd blocks.

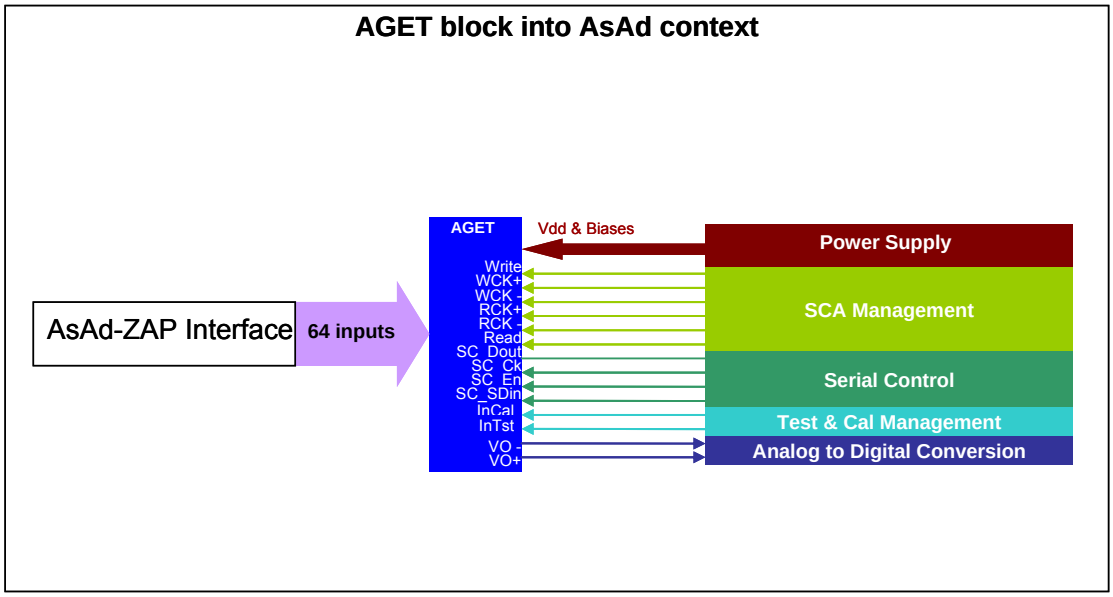


Figure 3: AGET into AsAd context

2.2 Switched Capacitors Array Management for AGET

This block is divided into two sub-blocks:
- The Cycle Controller which is basically responsible to transmit the sampling and readout commands coming from CoBo to AGET.
- The Local Clocks Manager which is in charge of spreading towards AGET the clocks required to make it work properly under any circumstances.

2.2.1 Cycle Controller

This FPGA-implemented controller receives through the AsAd-CoBo Interface the **SCW** and **SCR** signals which respectively represent the sampling and the readout command.
From these signals, the cycle controller is able to:
- generate the "write" and "read" commands sent to AGET
- define an AGET working state

The cycles are defined as follows:
- A high to low transition on SCR when SCW is low defines the general AGET configuration cycle, which stops on a SCW low to high transition.
- A low to high transition on SCW when SCR is low defines the AGET sampling cycle, which stops on a SCW high to low transition.
- A high to low transition on SCW when SCR is low defines the specific AGET "channel-addresses" configuration cycle, which stops on a SCR low to high transition.
- A low to high transition on SCR when SCW is low defines an AGET readout cycle, which stops on a SCR high to low transition.

Mode<0-1> gives the AGET working state in each cycle (table 3)

Cycle	**SCR**	**SCW**	**Mode <1>**	**Mode <0>**
General config.	↓	0	0	0
Sampling	0	↑	0	1
Channel-addresses config.	0	↓	1	0
Readout	↑	0	1	1

Table 3: AGET working cycles

Figure 4 shows the Cycle Controller foreseen architecture, its I/O names and standards are summarized in table 4

Name	Standard voltages	Direction
SCW	LVTTL/LVCMOS 3.3V (17mA input)	CoBo⇨Cycle Ctrl
SCR	LVTTL/LVCMOS 3.3V (17mA input)	CoBo⇨Cycle Ctrl
write	LVTTL/LVCMOS 3.3V	Cycle Ctrl⇨AGET
read	LVTTL/LVCMOS 3.3V	Cycle Ctrl⇨AGET
Mode<0-1>	FPGA standard logic	Cycle Ctrl⇨logic blocks

Table 4: Cycle Controller I/O names, standards and directions

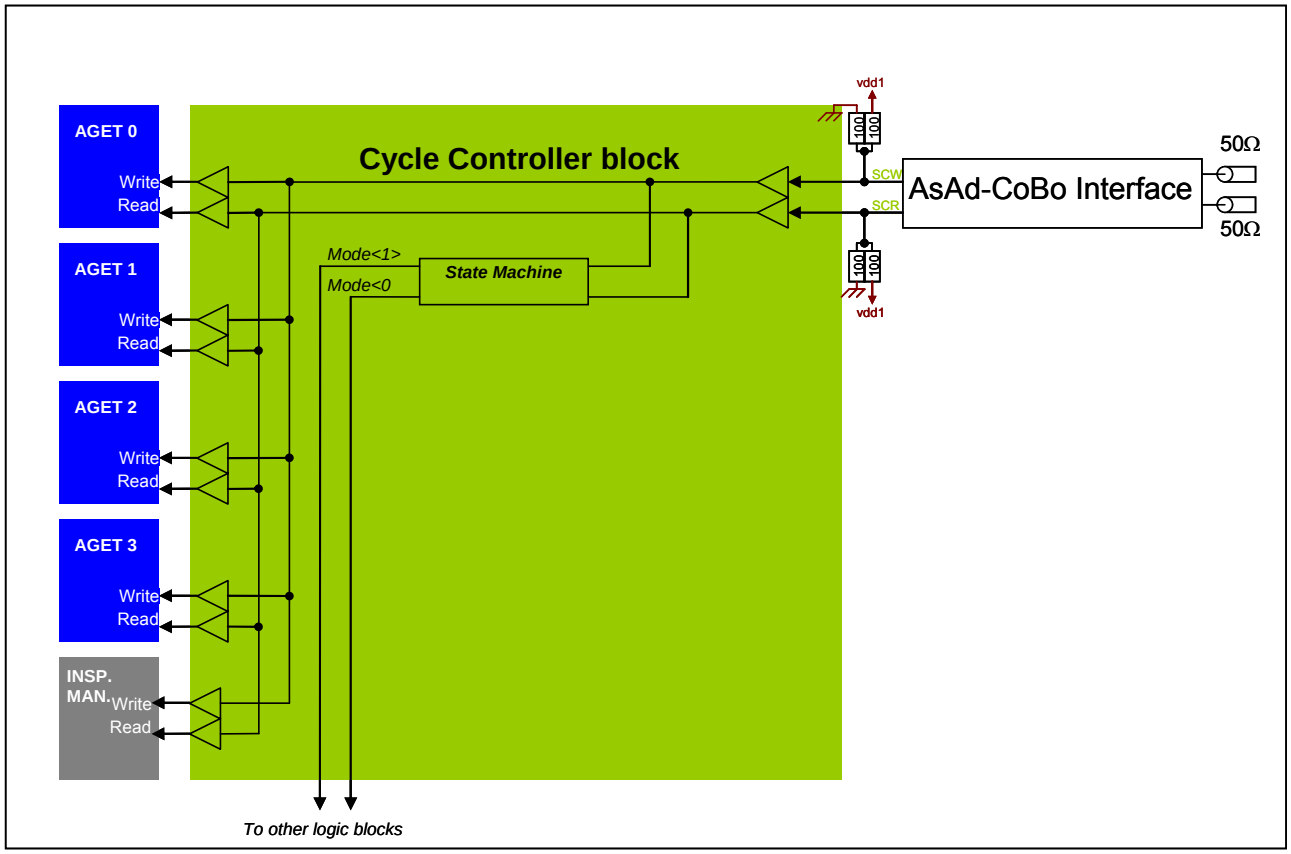


Figure 4: Cycle Controller architecture

2.2.2 Local Clocks Manager

This FPGA-implemented manager receives through the AsAd-CoBo Interface the **CKW**, **CKR** and **CKP** signals which respectively represent the sampling process clock, the readout process clock and the AGET serial transfer process clock.

CKW is a frequency adjustable clock (4MHz to 100MHz) which is continuously sent by CoBo.
CKR is a 25MHz frequency clock which is continuously sent by CoBo.
CKP is only active for data transferring through AGET serial port. Depending on AGET working cycle its frequency is 25MHz (for AGET general configuration mode) or 50MHz (for the specific AGET "channel-addresses" configuration mode)

In order to align the sampling clocks phases between several AsAd, the Local Clocks Manager passes CKW through a PLL and allows delaying it by slow control over a 6ns range by step of 200 ps. The resulting signal WCK is applied to AGET during a sampling cycle (Mode = 01)

The Local Clocks Manager passes CKR through a PLL and generates two signals:
- RCK which is the 25MHz clock used to extract from AGET the multiplicity (Mode = 01) or to extract the analogue samples from the SCA (Mode = 11)
- ACK which is the free running clock used in the analogue to digital conversion. ACK is a 180° phase-shifted image of RCK that can be delayed by slow control over a 6ns range by step of 200ps.

The inspection manager can be provided with WCK, RCK, ACK images at its inputs.

The Local Clocks Manager receives CKP when a serial transfer is initiated by CoBo (Mode = x0) it makes a level conversion on it and applies the converted signal SC_Ck, to the Serial Control block (see Serial Control section)

Figure 5 shows the Local Clocks Manager foreseen architecture, its I/O names and standards are summarized in table 5

Name	Standard voltages	Direction
CKW	LVDS	CoBo⇨Local Clocks Manager
CKR	LVDS	CoBo⇨Local Clocks Manager
CKP	LVDS	CoBo⇨Local Clocks Manager
Mode<0-1>	FPGA standard logic	Cycle Ctrl⇨ Local Clocks Manager
WCK	LVDS	Local Clocks Manager⇨AGET
RCK	LVDS	Local Clocks Manager⇨AGET
ACK	LVDS	Local Clocks Manager⇨ADC
SC_Ck	FPGA standard logic	Local Clocks Manager⇨Serial Controller

Table 5: Local Clocks Manager I/O names, standards and directions

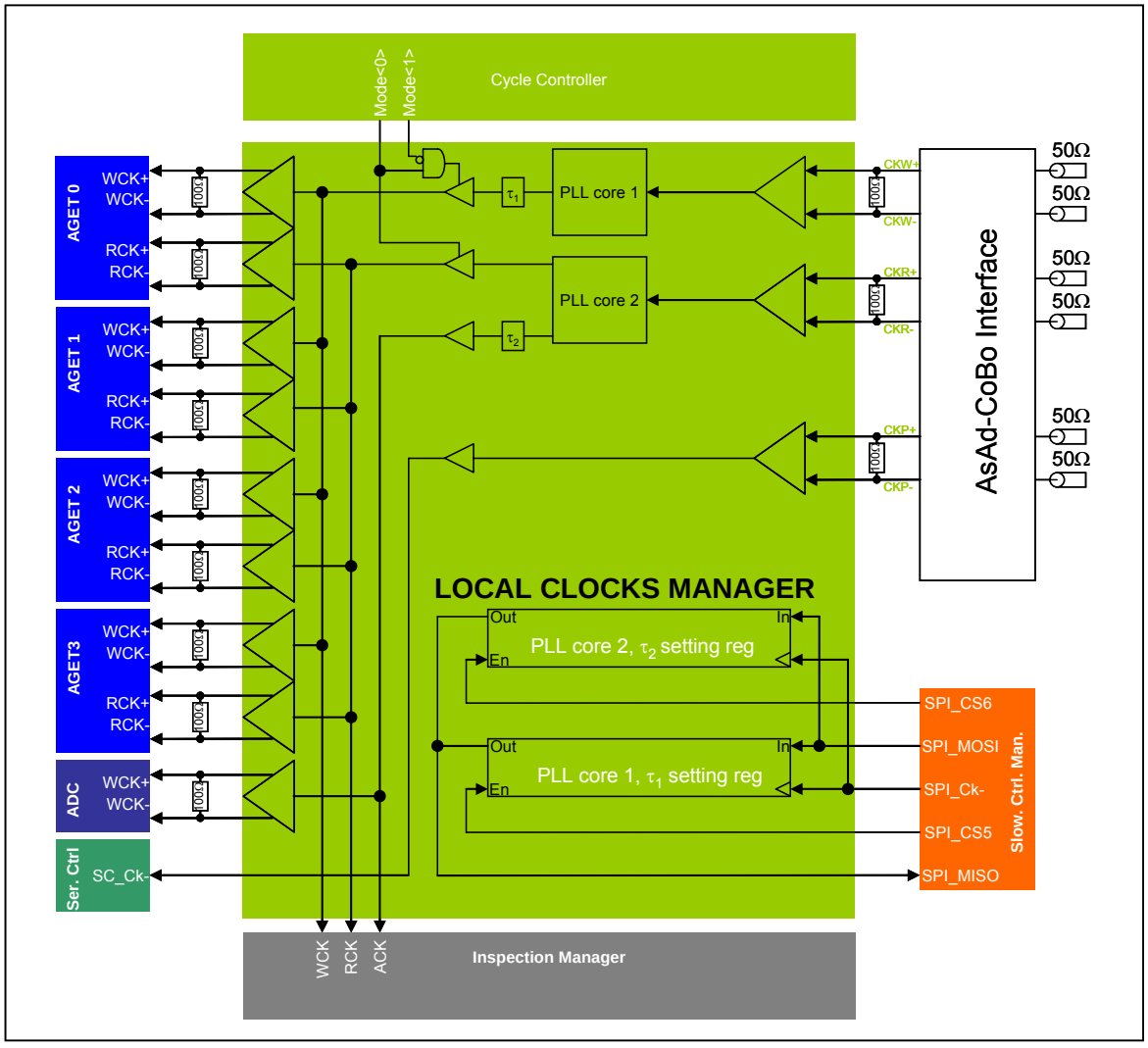


Figure 5: Local Clocks Manager architecture and interconnections

2.3 Serial Control for AGET

This FPGA-implemented controller block receives through the AsAd-CoBo Interface the **CAx** signals, and drives the **AxC** signals which represent in any AGET configuration mode, the data flow AGETx /CoBo (As there is 4 AGET per AsAd this block is replicated 4 times, and then x value can be 0, 1, 2 or 3).
From an AsAd-CoBo point of view, this transfer is synchronous with **CKP** (c. f. Local Clocks Manager section), it is then defined as a 3-wire, full-duplex serial transfer.

Depending on the AGET working mode the transfer sequences are defined as follows:

- When AGET is operated in general configuration mode (Mode = 00, c.f. Cycle Controller section):

A 16-bit preamble is sent from CoBo to AsAd through the CAx link, synchronously with the CKP signal (data present on the CKP rising edge to be latched on the CKP falling edge).
This preamble is stored into one Serial Controller register.
Its Most Significant Byte defines the size in bit of the body message that will follow.
Its Least Significant Byte is used by the serial controller to:

- Activate the SC_En_x signal which must be applied to AGETx during the body-message transfer.
- Enable or inhibit SC_Ck (the CKP image coming from the Local Clocks Manager) during the body transfer. When enabled SC_Ck becomes SC_Ck_x which is applied to AGETx

The 16-bit preamble is followed by a variable-sized body message which is formatted as expected to access any AGET configuration register:

- The first byte sent though CAx link is composed of one bit to set a read or write operation, followed by seven bits which define the address of the targeted AGETx register. The serial controller transfers this byte to AGETx through SC_Din_x.
- The following bits represent the register data.
  - In case of a write operation, the serial controller receives the data from CoBo through the CAx link and drives it to AGETx through SC_Din_x.
  - In case of a read operation, the serial controller receives the data from AGETx through the SC_Dout_x link and drives it to CoBo through the AxC link.

The body message is followed by an 8-bit postamble which represents the 2's complement of the sum of "ones" transferred between AsAd and CoBo in any direction.

- When AGET is operated in the specific "address configuration" mode (Mode = 10, c.f. Cycle Controller section) :

The same 16-bit preamble previously defined is sent from CoBo to AsAd to be equally processed by the serial controller

The 16-bit preamble is followed by a variable-sized body message (most often 69-bit) which is formatted as expected to access the specific AGET "address configuration" register:

- the first bit sent though CAx set a read or write operation of the "address configuration" register. The serial controller transfers this bit to AGETx through the SC_Din_x link.
- the following bits represent the register data (addresses configuration)
  - In case of a write operation, the serial controller receives the data from CoBo through the CAx link and drives it to AGETx through the SC_Din_x link.
  - In case of a read operation, the serial controller receives the data from AGETx through the SC_Dout_x link and drives it to CoBo through the AxC link.

No postamble is used in this mode.

Figure 6 shows the Serial Controller foreseen architecture, its I/O names and standards are summarized in table 6

Name	Standard voltages	Direction
CAx	LVDS	CoBo⇨Serial Controller
AxC	LVDS	Serial Controller ⇨ CoBo
SC_Ck	FPGA standard logic	Local Clocks Manager⇨Serial Controller
Mode<0-1>	FPGA standard logic	Cycle Controller⇨ Serial Controller
SC_Din_x	LVTTL/LVCMOS 3.3V	Serial Controller ⇨AGETx
SC_Dout_x	LVTTL/LVCMOS 3.3V	AGETx ⇨ Serial Controller
SC_En_x	LVTTL/LVCMOS 3.3V	Serial Controller ⇨AGETx
SC_Ck_x	LVTTL/LVCMOS 3.3V	Serial Controller ⇨AGETx

Table 6: Serial Controller I/O names, standards and directions

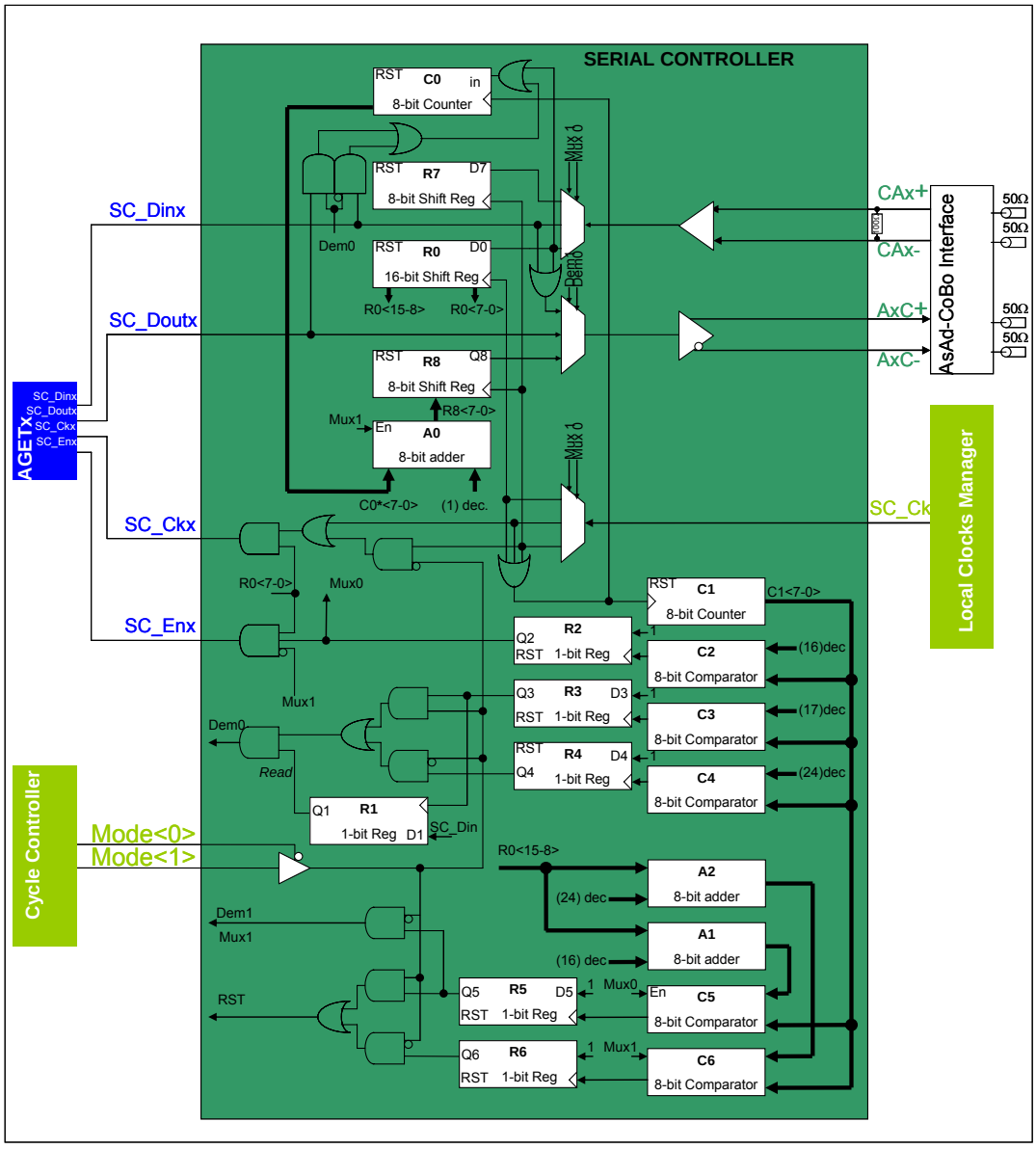


Figure 6: Serial Controller architecture and interconnections

2.4 Test & Calibration for AGET

AGET is able to perform:

- A **functional** test of each of its channels by means of a channel-dedicated internal capacitor
- A global **test** of its channels by means of channels-group-dedicated internal capacitors
- A **calibration** of its channels by means of channels-group-dedicated external capacitors

For each of these three operating modes, AsAd provides a pulse generator which output voltage swing generates across the selected capacitors, the charge injected at the channels inputs. This generator is made of a 14-bit current-mode DAC (AD9707) followed by a transimpedance amplifier.
The procedure used in each mode is fully set by slow control (c.f. slow control management section).
Assuming the DAC settings have already been made by slow control, a 24-bit word has to be sent to the FPGA test & calibration manager (c.f. figure 4-1)
The 24-bit word is formatted as follows:

- If high, the most significant bit indicates a calibration procedure. If low, a test mode procedure or a functional mode procedure is selected. In any case, the most significant bit value must be chosen according to the AGET settings previously defined.
- The following bit value defines the charge range in which the calibration will be made, when a calibration procedure has been selected. A high value corresponds to the highest charge ranges achievable, whereas a low value corresponds to lower ranges.
- The 14 following bits values define the pulse generator output voltage magnitude
- The further bit value indicates the way of triggering the pulse generator. A high value corresponds to an external triggering mode (c.f. Inspection Manager section) whereas a low value indicates an internal triggering mode, which is defined by the 6 least significant bit values.
- For internal triggering mode, the 6 least significant bits correspond to the delay introduced between the instant of the 24-bit word reception and the instant of the DAC triggering by the CKO signal. The minimal delay must not be lower than 40ns. Assuming this minimal delay equal to 40ns, the six bits enable to delay the DAC trigger output from 40ns to (40+6x) ns by step of x ns.

From a hardware point of view, the two most significant bits enable to select the position of the analogue switches (TS3A44159) involved in the generator output voltage handling. If a calibration procedure has been chosen, up to four charge ranges can be used by first selecting a 1pF±1% or a 10pF±1% external capacitor and then by limiting the pulse generator output voltage swing. Table 7 defines the four calibration ranges available.

	Charge range	Voltage eq. range	DAC current eq. range
	q_{max} (pC)	V_{max} (mV)	I_{max} (µA)
1pF	±0.12	±120	±60
	±0.24	±240	±120
10pF	±1	±100	±50
	±10	±1000	±500

Table 7: Calibration ranges available

The 1mA full scale output current DAC is defined by a 1.225V ±0.1% external voltage source reference (AD1580) ; as stated in figure 7, the output current DAC is voltage converted by a transimpedance amplifier made of an AD8062.
Table 8 summarizes the Test & Calibration block I/O names and properties.

AsAd Name	Properties	Interconnections
SPI_CK	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ Test & Cal
SPI_MOSI	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ Test & Cal
SPI_MISO	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ Test & Cal
SPI_CS1*	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ Test & Cal
SPI_CS2	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ Test & Cal

CKO	LVTTL/LVCMOS 3.3V (active on a low to high transition)	Inspection Man. ⇨ Test & Cal Test & Cal ⇨Inspection Man.
Vdd	3.3V DC (Block Supply)	Slow Ctrl. Man. ⇨ Test & Cal
InTest0	Analogue voltage (±1V dynamics)	Test & Cal⇨AGET0
InCal0	Analogue current	Test & Cal⇨AGET0
InTest1	Analogue voltage (±1V dynamics)	Test & Cal⇨AGET1
InCal1	Analogue current	Test & Cal⇨AGET1
InTest2	Analogue voltage (±1V dynamics)	Test & Cal⇨AGET2
InCal2	Analogue current	Test & Cal⇨AGET2
InTest3	Analogue voltage (±1V dynamics)	Test & Cal⇨AGET3
InCal3	Analogue current	Test & Cal⇨AGET3

Table 8: AGET Test & Calibration block I/O names and properties

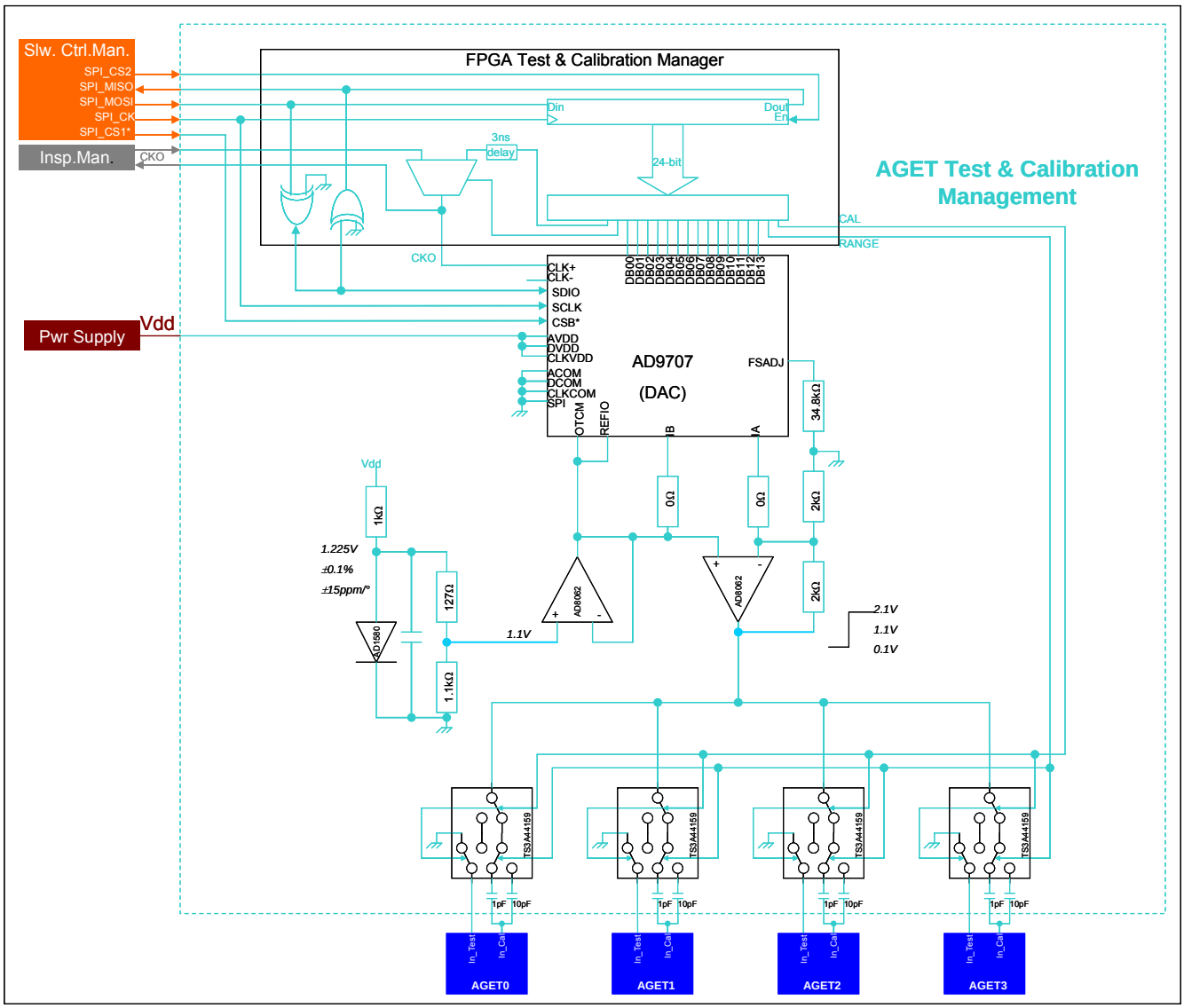


Figure 7: AGET Test & Calibration block architecture and interconnections

2.5 Analogue to Digital Conversion

This block receives from AGET output, either a signal representing the channels triggered sum during the sampling process (Mode = 01), either the analogue samples stored into the SCA during the readout process (Mode = 11).
These analogue signals are converted into digital values at the 25MHz frequency given by ACK (c.f. Local Clocks Manager section).

The block architecture consists of four 12-bit analogue to digital converters (each one relates to one AGET), one voltage reference and a PLL, all embedded into the chip ADS6422.
The ADS6422 power supply inputs are all connected to the AsAd main Vdd (c.f. power supply section), the device settings can be adjusted by slow control (c.f. slow control management section) once grounded its parallel input (PDN, CFG<4-1>).
The main default settings are:
- All conversion channels enabled
- Use of the internal voltage reference (1,5V input common mode voltage has to be then equal to AGET output common mode voltage)
- 1V peak to peak input full scale range
- Conversion processes initiated for all channels on ACK rising edges
- LVDS output signals driven through 100Ω differential impedances
- 2-wire, 1x frame clock, 12x serialisation with SDR bit clock, byte wise MSB-first mode outputs format (c.f. figure 8)

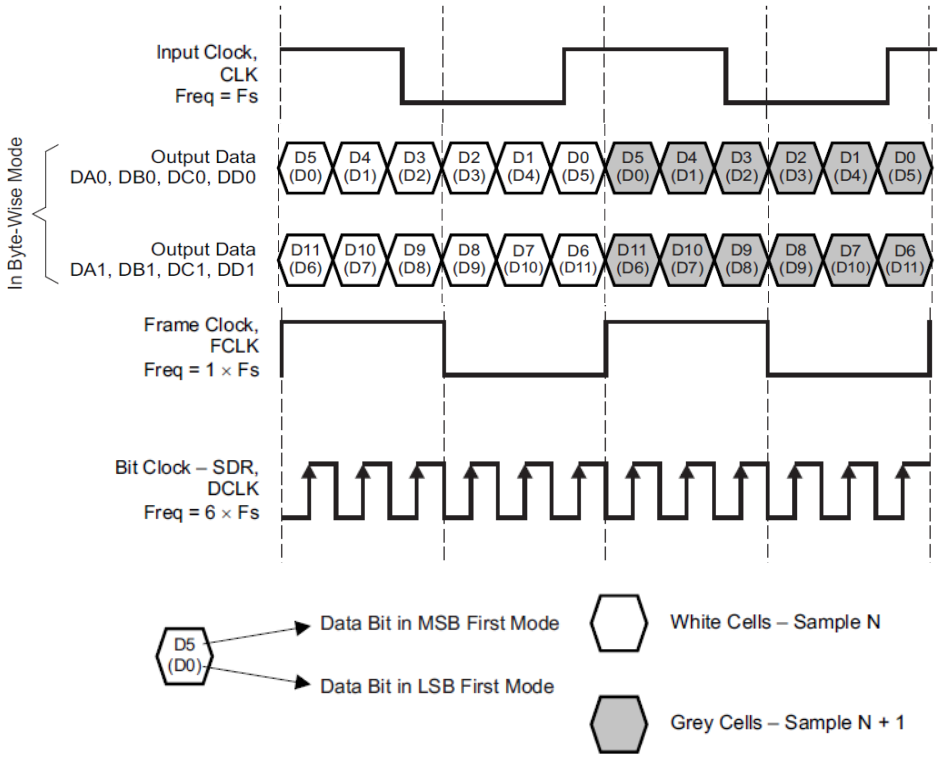


Figure 8: Analogue to Digital Conversion outputs format *(ADS6422 datasheet extraction)*

For each ADC, the 12-bit digital output data is transferred to CoBo in serial mode through two wires. Each wire enables to transfer a 6-bit data which can be unserializided into CoBo by using the ADC bit-clock (CKB) and the ADC frame clock (CKF) which is aligned with the 12-bit word boundary. These clocks signals are generated from ACK by the PLL embedded into the conversion device.

Figure 9 shows the AD conversion block architecture; its I/O names and properties are summarized in table 9

AsAd Name	ADS6422 IO Name	Properties	Direction
Vdd	AVDD, LVDD	3.3V supply voltage	Power Man.⇨ADC
ACK+, ACK-	CLKP, CLKM	LVDS	Loc. Clocks. Man. ⇨ADC
VO0+, VO0-	INA_P, INA_M	Analogue differential	AGET0 ⇨ ADC
VO1+, VO1-	INB_P, INB_M	Analogue differential	AGET1 ⇨ ADC
VO2+, VO2-	INC_P, INC_M	Analogue differential	AGET2 ⇨ ADC
VO3+, VO3-	IND_P,IND_M	Analogue differential	AGET3 ⇨ ADC
	CAP	2nF to Ground	
SPI_CK*	SCLK	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ ADC
SPI_MOSI	SDATA	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ ADC
SPI_CS0*	SEN*	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ ADC
GND	PDN	0V	
GND	CFG<4-1>	0V	
VOCM	VCM	1.5V (DC level)	ADC⇨ all AGET (option)
OA0+, OA0-	DA0_P, DA0_M	LVDS	ADC⇨ CoBo Interface
OA1+, OA1-	DA1_P, DA1_M	LVDS	ADC⇨ CoBo Interface
OB0+, OB0-	DB0_P, DB0_M	LVDS	ADC⇨ CoBo Interface
OB1+, OB1-	DB1_P, DB1_M	LVDS	ADC⇨ CoBo Interface
OC0+, OC0-	DC0_P, DC0_M	LVDS	ADC⇨ CoBo Interface
OC1+, OC1-	DC1_P, DC1_M	LVDS	ADC⇨ CoBo Interface
OD0+, OD0-	DD0_P, DD0_M	LVDS	ADC⇨ CoBo Interface
OD1+, OD1-	DD1_P, DD1_M	LVDS	ADC⇨ CoBo Interface
CKB+, CKB-	DCLKP, DCLKM	LVDS	ADC⇨ CoBo Interface
CKF+, CKF-	FCLKP, FCLKM	LVDS	ADC⇨ CoBo Interface

Table 9: AD conversion I/O names and properties

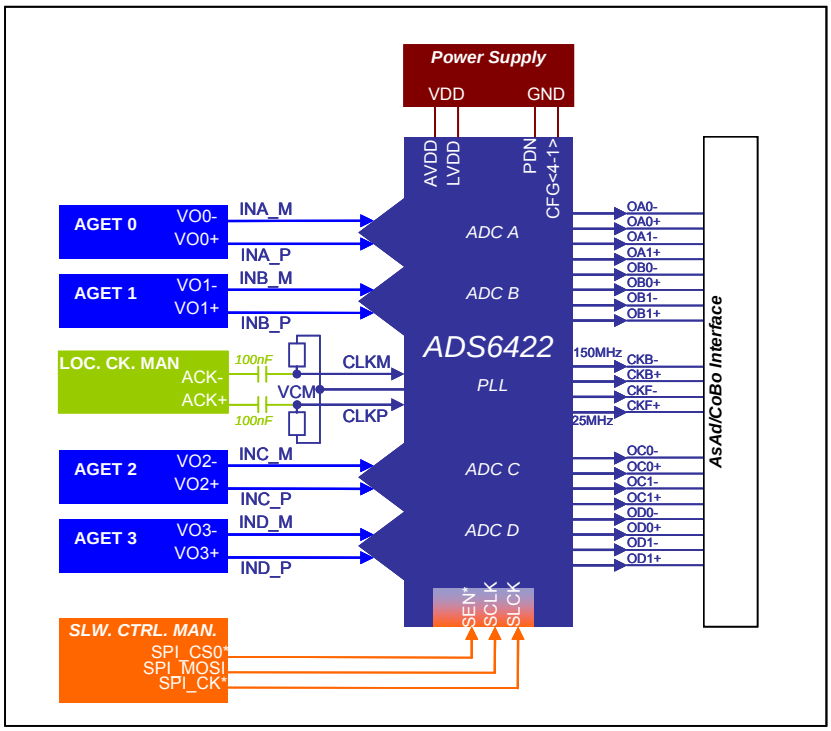


Figure 9: AD conversion architecture and interconnections

2.6 Power Supply

ASAD board uses the GET power supply unit to make its own power supply sources.
Four voltage sources are available into AsAd:
AsAd main voltage source, **Vdd = 3.3V**, which is used to supply all active devices excepting those involved in the monitoring process.
AsAd secondary voltage source, **Vdd1 = 3.3V**, which is especially dedicated to supply the monitoring active devices.
AsAd LVDS voltage source, **Vdd2 = 2.5V**, which is dedicated to supply all the LVDS buffers.
AsAd FPGA voltage source, **Vdd3=1.5V**, which is dedicated to supply the AsAd FPGA core as well as the PLL cores.

Each voltage source is obtained from an ultra low dropout (ULDO) voltage regulator in order to minimize the dissipated power.
The TPS74401 ULDO is chosen not only because of its low dropout feature for high current supplies, but also because of its ability to be easily controlled (Enable input and Power Good output flag).
As this regulator requires a 5V reference voltage, a LTC1516 DC/DC converter which is not sensitive to the AsAd magnetic field environment has been chosen to generate the reference.
The AsAd Power Supply block also uses a Power Manager embedded into the AsAd FPGA that is able to deal with the regulators Enable and Power Good signals as well as the Alert signal coming from the monitoring core device (c.f. Monitoring section)
Figure 10 shows the AsAd Power Supply architecture; its I/O names and properties are summarized in table 10

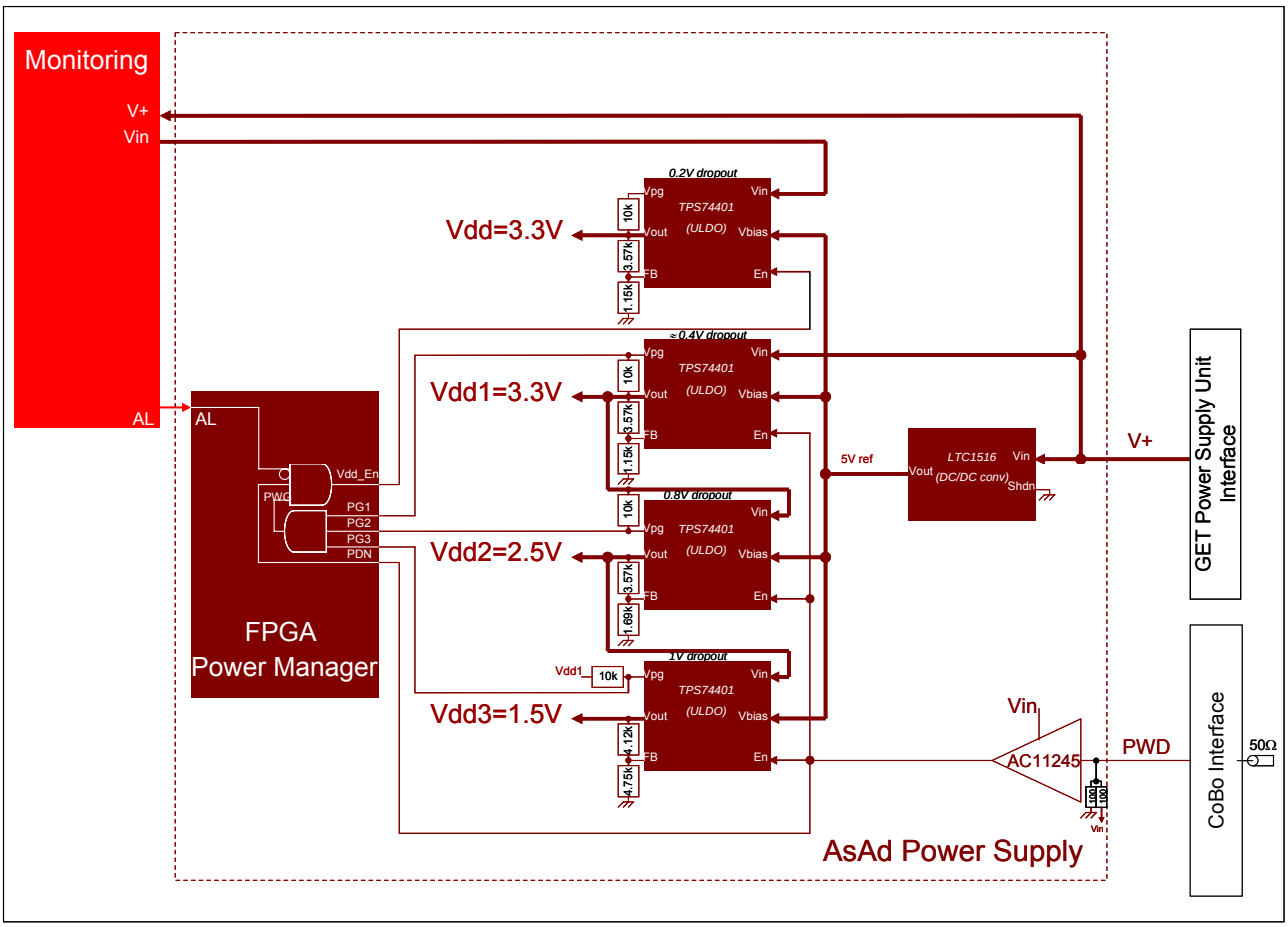


Figure 10: Power Supply architecture and interconnections

Name	Standard voltages	Direction
V+	Global power supply voltage	GET Pwr Unit ⇨ AsAd Pwr Supply
Vin	3.5V DC (main power supply input)	Monitoring ⇨ AsAd Pwr Supply
PWD	LVTTL/LVCMOS 3.3V (17mA input)	CoBo ⇨ AsAd Pwr Supply
AL	LVTTL/LVCMOS 3.3V	Monitoring ⇨ AsAd Pwr Supply
PWG	LVTTL/LVCMOS 3.3V	AsAd Pwr Supply ⇨ Insp. Man.
Vdd	3.3V DC (main pwr. supply output)	AsAd Pwr Supply ⇨ AGET AsAd Pwr Supply ⇨ ADC AsAd Pwr Supply ⇨ Test & Cal
Vdd1	3.3V DC (second. Pwr. supply output)	AsAd Pwr Supply ⇨ FPGA LVCMOS IO AsAd Pwr Supply ⇨ Monitoring
Vdd2	2.5V DC (LVDS power supply)	AsAd Pwr Supply ⇨ LVDS buffers
Vdd3	1.5V DC (core FPGA power supply)	AsAd Pwr Supply ⇨ FPGA core

Table 10: Power Supply I/O names and properties

Operation:

Even when AsAd is connected to the GET power supply unit, AsAd devices are not supplied until **PWD** goes high (CoBo controls AsAd power up).
If PWD is high, then Vdd1, Vdd2 and Vdd3 are enabled (FPGA IO and core as well as AsAd Monitoring are supplied).
When each of all these voltage sources have reached 90% of their nominal value the PWG signal goes high (This signal level can be checked, c.f. Inspection Manager section)
When PWG is high, AsAd main power source is enabled. The main power supply voltage and current are checked by the monitoring. If any limit stored into the Monitoring core device is exceeded, the alert signal (AL) goes high and the AsAd main power source is switched off. The failure can be found by reading the Monitoring status registers (Slow Control operation)

2.7 Monitoring

ASAD board is able to monitor the temperature at which it is operated, its main power supply voltage and current levels (Vdd, Idd).
These measurements are obtained from sensors connected to one monitoring core device, the ADT7519. The settings of this device are adjusted by slow control (c.f. slow control management section).

Once the ADT7519 has been supplied by the AsAd secondary power source Vdd1 (c.f. power supply section), it makes an auto-calibration during 5ms. It is then able to scan and convert into digital data, sequentially:

- its own power supply voltage (AsAd secondary power source voltage Vdd1)
- its own operating temperature (Tint)
- the temperature measured at AsAd hottest point (Text)
- AsAd main power source current (Idd)
- AsAd main power source voltage (Vdd)

For each item 16 measurements are made and their mean values are stored into 8-bit registers (in 2's complement format for the temperature measurement, in natural binary for all others measurements). The 1.4 kHz conversion frequency is supplied by internal means into the ADT7519.
The register contents can be accessed at any time by slow control.
The slow control enables as well to store into the ADT7519 the top and bottom limits not to exceed for each item.
If any limit is exceeded the ADT7519 gives an alert signal which is used to:

- inform CoBo that a failure occurred
- switch off AsAd main power source

The secondary power source, which is only used to supply the ADT7519 and the slow control manager, is still switched on allowing thus CoBo to find the point of failure.
Figure 11 shows the monitoring block architecture.

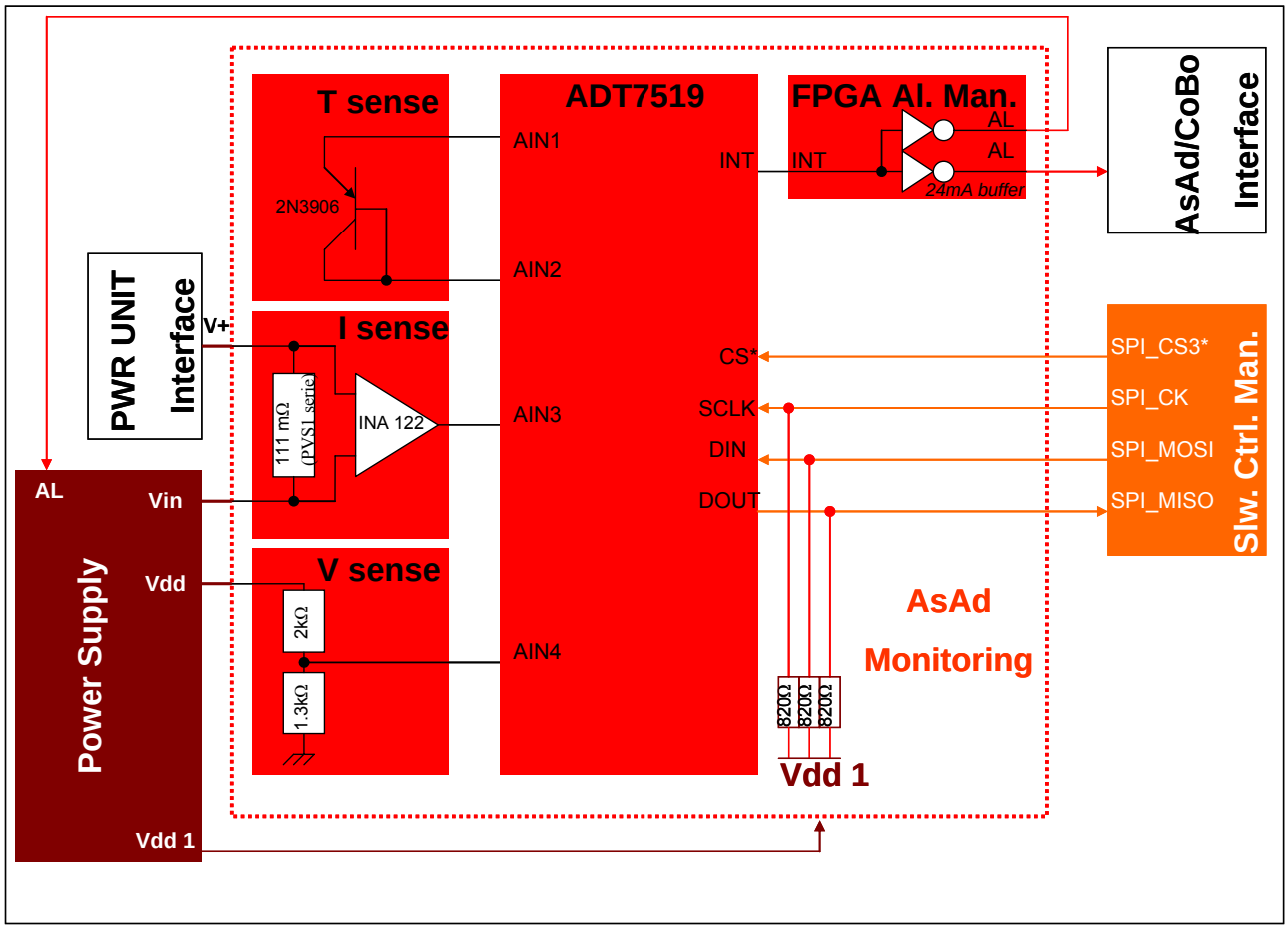


Figure 11: Monitoring architecture and interconnections

This architecture is composed of :

- One temperature sensor (Text measurement) made of a semi-conductor junction (2N3906 transistor)
- One current sensor (Idd measurement) made of a PVS1 resistor and an INA122 differential amplifier
- One voltage sensor (Vdd measurement) made of a voltage divider.
- One core device, the ADT7519.
- One Alert Manager FPGA block made of inverters-buffers. The aim of the alert manager is to disable all the FPGA IO except those concerning the slow control transfers when an alert has been sent by the monitoring core device.

The monitoring I/O names and properties are summarized in table 11. The ADT7519 (monitoring core) connections are summarized in table 12.

AsAd Name	Block IO Name	Properties	Direction
V+	I sense input	GET power unit voltage	Power Interface ⇨Monitoring
Vin	I sense output	3.5V DC voltage	Monitoring⇨ Power Supply
Vdd	V sense input	3.3V DC voltage	Power Supply ⇨ Monitoring
Vdd1	VDD	3.3V DC block supply	Power Supply ⇨ Monitoring
AL	AL	LVTTL/LVCMOS 3.3V	Monitoring ⇨ CoBo Interface Monitoring ⇨ Power Supply

SPI_CK	SCLK	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ Monitoring
SPI_MOSI	DIN	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ Monitoring
SPI_MISO	DOUT	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ Monitoring
SPI_CS3*	CS*	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ Monitoring

Table 11: Monitoring I/O names and properties

ADT7519 IO Name	Properties	Direction
VDD	3.3V supply voltage	Power Man. Vdd1⇨ ADT7519
Vref-IN	3.3V voltage	Power Man. Vdd1⇨ ADT7519
AIN1, AIN2	Analogue inputs	Tsense ⇨ADT7519
AIN3	Analogue input	Isense ⇨ ADT7519
AIN4	Analogue input	Vsense ⇨ ADT7519
INT	LVTTL/LVCMOS 3.3V	ADT7519 ⇨ FPGA Alert Man
SCLK	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ ADT7519
DIN	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ ADT7519
DOUT	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ ADT7519
CS	LVTTL/LVCMOS 3.3V	Slow Ctrl. Man. ⇨ ADT7519

Table 12: Core device (ADT7519) I/O names and properties

2.8 Slow Control

The slow control is an AsAd-CoBo data transfer which enables CoBo to:

- define the behaviour of all AsAd configurable devices (excepting AGET) by writing instructions into their configuration registers
- check the working state of these devices by reading into their status registers

The data transfer is defined as a full duplex synchronous serial transfer since:

- the transfer from CoBo to AsAd is made by one link (**CAS**)
- the transfer from AsAd to CoBo is made by one link (**ACS**)
- CoBo always masters the 10MHz **CKS** clock that is only active during the data transfer process.

The data transfer protocol consists of one preamble followed by a body message, itself followed by a postamble.

2.8.1 AsAd Slow Control Manager data-flow handling overview

The AsAd slow control manager is an FPGA implemented device which is able to handle the data-flow between CoBo and any AsAd devices excepting AGET.

From an AsAd/CoBo point of view, the slow control manager acts as follows:

- When CoBo wants to write a data into an AsAd register, it sends through CAS the data flow containing the preamble, the body message and the postamble. This data flow is received by the Slow Control Manager and is sent back to CoBo through ACS with an updated postamble (the postamble format will be described hereafter).
- When CoBo wants to read into an AsAd register, it sends through CAS the data flow containing the preamble, the body message and the postamble. This data flow is sent back to CoBo by the Slow Control Manager which updates first the body message by inserting the data extracted from the AsAd register and updates then the postamble.

From an AsAd Slow Control Manager/AsAd device point of view, the slow control manager acts as follows:

- It stores the preamble sent by CoBo and generates from it the signal enabling the device serial port (Chip Select)
- It transfers then to the device the body message received from CoBo through its MOSI output (Master Out Slave In) and, in case of a read operation, receives from the device an extracted data through its MISO input (Master In Slave Out).

The Slow Control Manager is basically in charge to switch from CoBo to one AsAd device or to switch from an AsAd device to CoBo, the correct data flow.

To complete all these operation any bit that has to be stored into a register must be present at a CKS rising edge to be latched on the following falling edge, any bit that have to be extracted from a register is present on a CKS falling edge and can be processed on the following CKS leading edge

2.8.2 General description of the data flow

The 8-bit preamble four Most Significant Bits define an AsAd device address which is used by the Slow Control Manager to generate the signal enabling the device serial port. The four Least Significant Bits define the size in byte of the body message that follows. These bits are used by the Slow Control Manager to check the data transfer frame length.

The 8-bit preamble is followed by a variable-sized body message which is formatted as expected to access any device register.

The body message is itself followed by an 8-bit postamble which represents the 2's complement of the sum of "ones" transferred between AsAd and CoBo in any direction

Preamble content

Nine devices can be accessed by slow control:

- The ADS6422 (*ADC)* which converts the AGET analogue data
- The AD9707 (*DAC)* which generates the AGET test or calibration pulses
- The FPGA implemented test & calibration manager (*TCM*) which controls the AGET test mode procedures
- The ADT7519 (*AMD*) which is the AsAd monitoring core device
- The FPGA implemented identification memory (*AIM*)
- The FPGA implemented local clocks manager first device (*LC1*) which controls the AGET sampling clock
- The FPGA implemented local clocks manager second device (*LC2*) which controls the AGET readout clock and the AD conversion clock
- The FPGA implemented AsAd input manager (*INP*)
- The FPGA implemented AsAd inspection manager (*ISP*)

Table 13 shows the preamble format linked to each accessed device within read or write mode.

Device Reference	Decimal Address	Binary Address (Preamble MSB)	Access	Body size in bytes	Binary Body Size in bytes (Preamble LSB)
ADC	1	0000	write	2	0001
DAC	2	0001	write	2	0001
			read	2	0001
TCM	3	0010	write	3	0010
			read	3	0010
AMD	4	0011	write	**3**	0010
			read	**4**	0011
AIM	5	0100	read	4	0011
LC1	6	0101	write	10	1001
			read	10	1001
LC2	7	0110	write	10	1001
			read	10	1001
INP	8	0111	write	1	0000
			read	1	0000
ISP	9	1000	write	1	0000
			read	1	0000

Table 13: Slow Control preamble content

Body content
As the body content varies depending on each accessed device within read or write mode, the format is given for each combination:

- The ADC is an eight write-only 11-bit registers device requiring a body message that specify a register address (5 most significant bits) followed by a 11bit data (table 14). The ADC registers map will be given into a complementary document.

(MSB) Address					Data										(LSB)
A4	A3	A2	A1	A0	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Table 14: Body content format by writing a data into an ADC register

- The DAC is an eight 8-bit registers device accessible within read or write mode that requires a body message which specifies a R/W operation (most significant bit) followed by a 00 format code and 5 address bits. This first byte is followed by a data byte (table 15). The DAC registers map will be given into a complementary document.

(MSB) Instruction			Address					Data							(LSB)
R/W	0	0	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0

Table 15: Body content format by reading or writing into a DAC register

- The TMC is a 24-bit register device accessible within read or write mode. In any mode, the body message is a 24 bit data (MSB first).

- The AMD is a twenty seven read or write 8-bit registers device. By writing into any of its 17 R/W registers the body message is formatted as follows in table 16.

(MSB) Instruction byte								Address byte							
1	0	0	1	0	0	0	0	A7	A6	A5	A4	A3	A2	A1	A0

Data byte							(LSB)
D7	D6	D5	D4	D3	D2	D1	D0

Table 16: Body content format by writing into an AMD register

By reading from any of its 10 read only registers or 17 R/W registers the body message is formatted as follows in table 17.

(MSB) Instruction byte								Address byte							
1	0	0	1	0	0	0	0	A7	A6	A5	A4	A3	A2	A1	A0

Instruction byte								Data byte							(LSB)
1	0	0	1	0	0	0	1	D7	D6	D5	D4	D3	D2	D1	D0

Table 17: Body content format by reading from an AMD register

The Monitoring core registers map will be given into a complementary document.

- The AIM is a 32-bit read only register device which is defined in the identification section. When it receives a read instruction the body message transferred is a 32-bit data (MSB first).
- The LC1 and LC2 each represent an 80-bit register accessible within read or write mode. In any mode the body message transferred is an 80-bit data (LSB first as stated in table 18)

(LSB)																			(MSB)
D0																			D79

Table 18: Body content format by reading or writing into the LC1 or LC2 registers

- The INP is an 8-bit register accessible within read or write mode. In any mode the body message transferred is an 8-bit data (MSB first).
- The ISP is an 8-bit register accessible within read or write mode. In any mode the body message transferred is an 8-bit data (MSB first).

2.9 Inputs Management

This FPGA implemented manager which is accessible by slow control, enables to bias or not the anti-sparkling protection circuits which are implemented into ZAP, as well as connect or disconnect 64 AGET inputs from 64 detector outputs.

Figure 12 shows the Input Manager foreseen architecture, its output names and standards are summarized in table 19
By default AGET inputs are disconnect from the detector outputs by setting low the
SWZ<3-0> levels and the anti-sparkling protection are biased by setting low the DBi<3-0>. These voltages are applied to the TS3A5018 analogue switches control inputs.

Name	Standard voltages	Direction
DBi<3-0>	LVTTL/LVCMOS 3.3V	Input Manager ⇨ ZAP Interface
SWZ<3-0>	LVTTL/LVCMOS 3.3V	Input Manager ⇨ ZAP Interface

Table 19: Inputs Manager output names, standards and direction

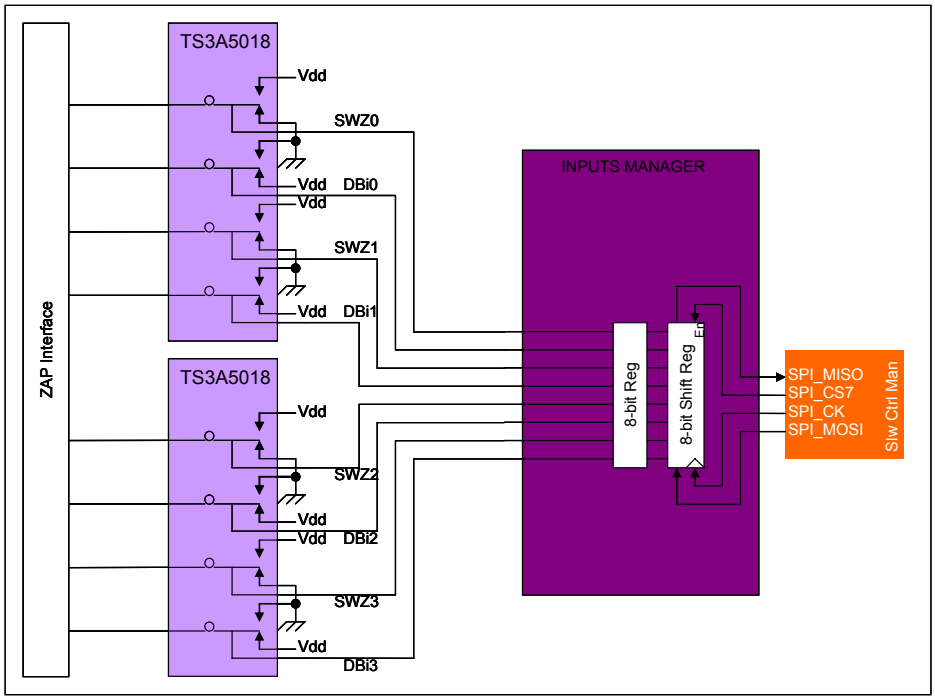


Figure 12: Inputs Manager architecture and interconnections

2.10 Inspections Management

This FPGA implemented manager receives or drives some AsAd "critical" signals by the External Instruments Interface.
The signals to be received or to be driven are chosen by slow control.
The Inspection Manager can drive to the interface:

- The signals Write (SCA write command) coming from the Cycle Controller and WCK (sampling clock) coming from the Local Clocks Manager
- Or the signals Read (SCA read command) coming from the cycle controller and RCK (readout clock) coming from the Local Clocks Manager
- Or the signals RCK and ACK (readout clock and AD conversion clock) coming from the Local Clocks Manager
- Or the signal PWG (Power supply good) coming from the Power Manager
- Or the signal CKO (Calibration pulse trigger sent by slow control) coming from the test and calibration manager

The Inspection Manager can receive from the interface:

- The signal CKO (Calibration pulse trigger sent by an external generator)

Figure 13 shows the Inspection Manager foreseen architecture, its I/O names and standards are summarized in table 20

Name	Standard voltages	Direction
CKO	LVTTL/LVCMOS 3.3V	Test & Cal. Man. ⇨ Inspection Manager
PWG	LVTTL/LVCMOS 3.3V	Power Manager ⇨ Inspection Manager
WCK	LVTTL/LVCMOS 3.3V	Local Clocks Manager ⇨ Inspection Manager
RCK	LVTTL/LVCMOS 3.3V	Local Clocks Manager ⇨ Inspection Manager
ACK	LVTTL/LVCMOS 3.3V	Local Clocks Manager ⇨ Inspection Manager
Write	LVTTL/LVCMOS 3.3V	Cycle Controller ⇨ Inspection Manager
Read	LVTTL/LVCMOS 3.3V	Cycle Controller ⇨ Inspection Manager
DIR	LVTTL/LVCMOS 3.3V	Inspection Manager ⇨Ext. Inst. Interface
ISP1	LVTTL/LVCMOS 3.3V	Inspection Manager ⇨Ext. Inst. Interface
ISP2	LVTTL/LVCMOS 3.3V	Inspection Manager ⇨Ext. Inst. Interface

Table 20: Inspection Manager I/O names, standards and direction

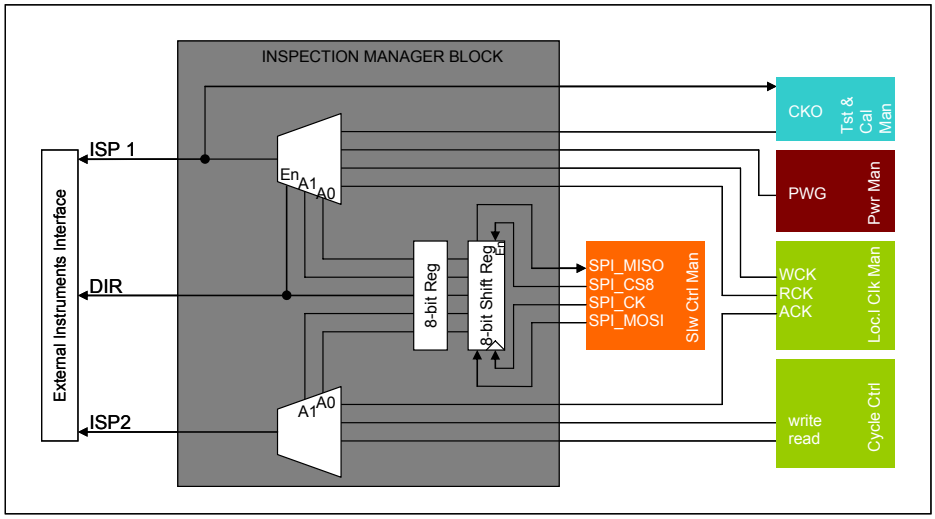


Figure 13: Inspection Manager architecture and interconnections

2.11 Identification

The block in use to identify AsAd is implemented into its FPGA on-chip user non volatile memory (Actel A3PE1500). It consists of a 32-bit register which content is written by the FPGA JTAG interface.

The register content is readable by slow control. Its format is the same defined for all GET hardware sub-systems and is given in table 21.

I<31-28>	I<27-24>	I<23-20>	I<19-16>	I<15-12>	I<11-0>
User defined	Slot Number	TPC reference	Hard release	Board Type	Serial number

Table 21: Identification format

3 PART TWO: AsAd Interfaces

3.1 AsAd-CoBo Interface

The AsAd -CoBo interconnection is completed by a 3 meters length cable which consists of double-shielding (foil and braid) enclosing 34 twisted pairs connected to VHDCI connectors (68-contacts). This cable supports SCSI transfers up to 320Mbps (it is then well suited for high transfer rates through 50Ω differential pairs).
The connexion diagram for this cable used as the AsAd-CoBo interconnection is given in figure 14. Table 22 gives for each signal name, its position, its standard and direction, as well as its general description.

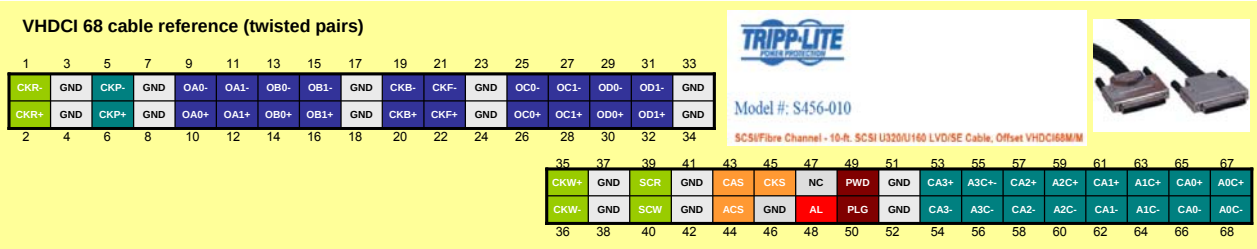


Figure 14: AsAd-CoBo interconnection

Position	Name	Description	Standard	Direction
1,2	CKR	SCA Read Clock	LVDS	ASAD⇦ COBO
5,6	CKP	AGET Serial Control Clock	LVDS	ASAD⇦ COBO
9,10	OA0	ADC A output LSB	LVDS	ASAD⇨COBO
11,12	OA1	ADC A output MSB	LVDS	ASAD⇨COBO
13,14	OB0	ADC B output LSB	LVDS	ASAD⇨COBO
15,16	OB1	ADC B output MSB	LVDS	ASAD⇨COBO
19,20	CKB	ADC's Bit Clock	LVDS	ASAD⇨COBO
21,22	CKF	ADC's Frame Clock	LVDS	ASAD⇨COBO
25,26	OC0	ADC C output LSB	LVDS	ASAD⇨COBO
27,28	OC1	ADC C output MSB	LVDS	ASAD⇨COBO
29,30	OD0	ADC D output LSB	LVDS	ASAD⇨COBO
31,32	OD1	ADC D output MSB	LVDS	ASAD⇨COBO
35,36	CKW	SCA Write Clock	LVDS	ASAD⇦ COBO
39	SCR	SCA read	LVCMOS 3.3V	ASAD⇦ COBO
40	SCW	SCA write	LVCMOS 3.3V	ASAD⇦ COBO

Position	Name	Description	Standard	Direction
43	CAS	Slow Control Data in	LVCMOS 3.3V	ASAD⇦ COBO
44	ACS	Slow Control Data out	LVCMOS 3.3V	ASAD⇨COBO
45	CKS	Slow Control Clock	LVCMOS 3.3V	ASAD⇦ COBO
47	NC	Not Connected		
48	AL	Alarm Flag	LVCMOS 3.3V	ASAD⇨COBO
49	PWD	ASAD switch on/off	LVCMOS 3.3V	ASAD⇦ COBO
50	PLG	ASAD presence	LVCMOS 3.3V	ASAD⇨COBO
53,54	CA3	AGET3 Serial Control Data in	LVDS	ASAD⇦ COBO
55,56	A3C	AGET3 Serial Control Data out	LVDS	ASAD⇨COBO
57,58	CA2	AGET2 Serial Control Data in	LVDS	ASAD⇦ COBO
59,60	A2C	AGET2 Serial Control Data out	LVDS	ASAD⇦ COBO
61,62	CA1	AGET1 Serial Control Data in	LVDS	ASAD⇦ COBO
63,64	A1C	AGET1 Serial Control Data out	LVDS	ASAD⇨COBO
65,66	CA0	AGET0 Serial Control Data in	LVDS	ASAD⇦ COBO
67,68	A0C	AGET0 Serial Control Data out	LVDS	ASAD⇨COBO

Positions 3, 4, 7, 8, 17,18, 23, 24, 33, 34, 37, 38, 41, 42, 46, 51 52 are grounded (GND =OV)

Table 22: AsAd-CoBo interconnection details

The AsAd connector that matches with the previously described cable, as well as its pinout, are represented on figure 15

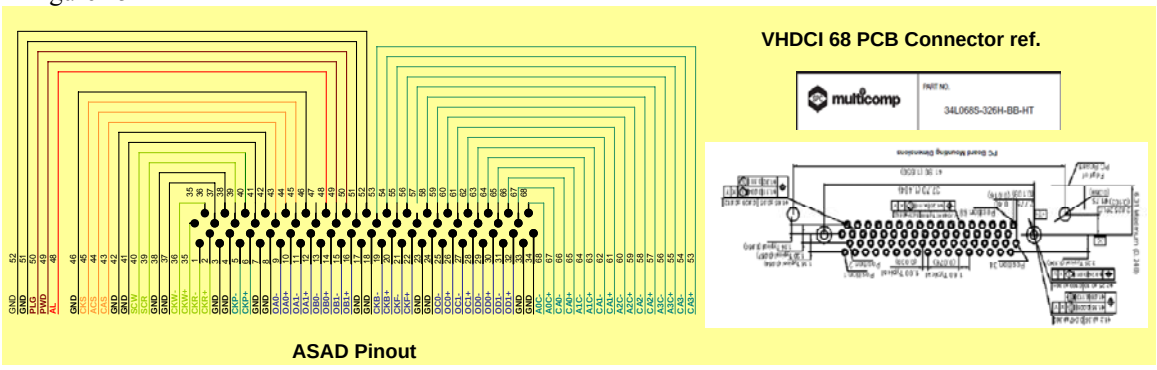
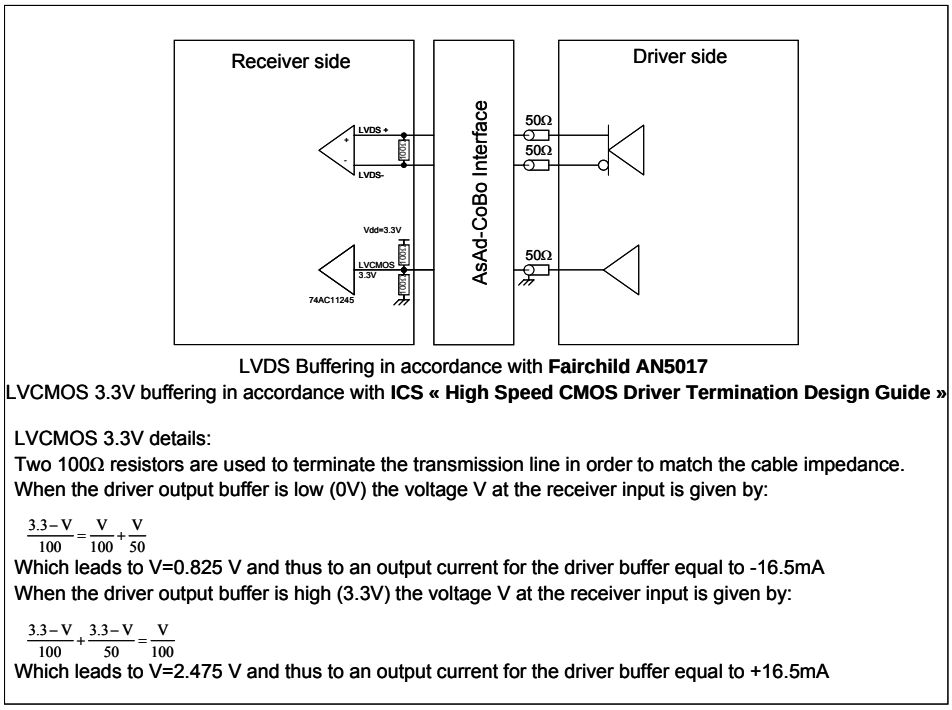


Figure 15: AsAd connection side

Buffering strategy:

All signals that have to be transferred through the AsAd-CoBo Interface are LVDS or LVTTL/LVCMOS 3.3V compatible (excepting PLG which will be described here-after).
The global buffering scheme used for each of these standards is given in figure 16. To save room-space first, and to have enhanced performances on the AsAd side, it is foreseen to include into the AsAd FPGA all the LVDS buffers (jitter + skew < 20ps rms). The LVTLL/LVCMOS 3.3V receivers must be outside the FPGA (The maximal voltage admissible for a FPGA buffer input in a low state is 0.4V, versus 0.9V for a 74AC11245 as an example device).



LVDS Buffering in accordance with **Fairchild AN5017**
LVCMOS 3.3V buffering in accordance with **ICS « High Speed CMOS Driver Termination Design Guide »**

LVCMOS 3.3V details:
Two 100Ω resistors are used to terminate the transmission line in order to match the cable impedance.
When the driver output buffer is low (0V) the voltage V at the receiver input is given by:

$$\frac{3.3-V}{100}=\frac{V}{100}+\frac{V}{50}$$

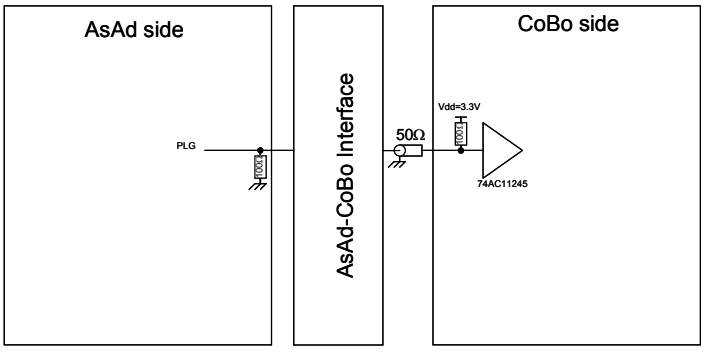
Which leads to V=0.825 V and thus to an output current for the driver buffer equal to -16.5mA
When the driver output buffer is high (3.3V) the voltage V at the receiver input is given by:

$$\frac{3.3-V}{100}+\frac{3.3-V}{50}=\frac{V}{100}$$

Which leads to V=2.475 V and thus to an output current for the driver buffer equal to +16.5mA

Figure 16: AsAd-CoBo buffering scheme

The PLG connection enables CoBo to detect AsAd presence. Following the previous scheme, this connection is made as represented in figure 17.



When AsAd is disconnected, the CoBo input buffer voltage is 3.3V
When AsAd is connected, the CoBo input buffer voltage is 0.825V

Figure 17: AsAd presence from a CoBo point of view

3.2 AsAd-GET Power Unit Interface

The AsAd connector that allows linking the GET power supply unit is one of the M250 series ATI connectors (There is no M250 description available on the internet but this series are vey similar with the M220)
This connector is a 2 gold plated contacts (15A/ctc) male connector with latches that exhibits a 6 mΩ resistive contact. Its metallic body enables to meet the EMC requirements by using an interconnection cable which shielding is ground-connected on both sides of the cable.
Figure 18 is a diagram showing how to connect to the GET Power Supply Unit using the ATI M250 series connector:
AsAd is linked to a power dispatching module by a 30cm length shielded cable.
Each dispatcher enables to feed four AsAd board and is itself linked by a M250 4-contacts connector to a 200W power supply module. The M250 4-contacts connector allows a load regulation of the supply voltage (V+ - V-) applied at each dispatcher input.

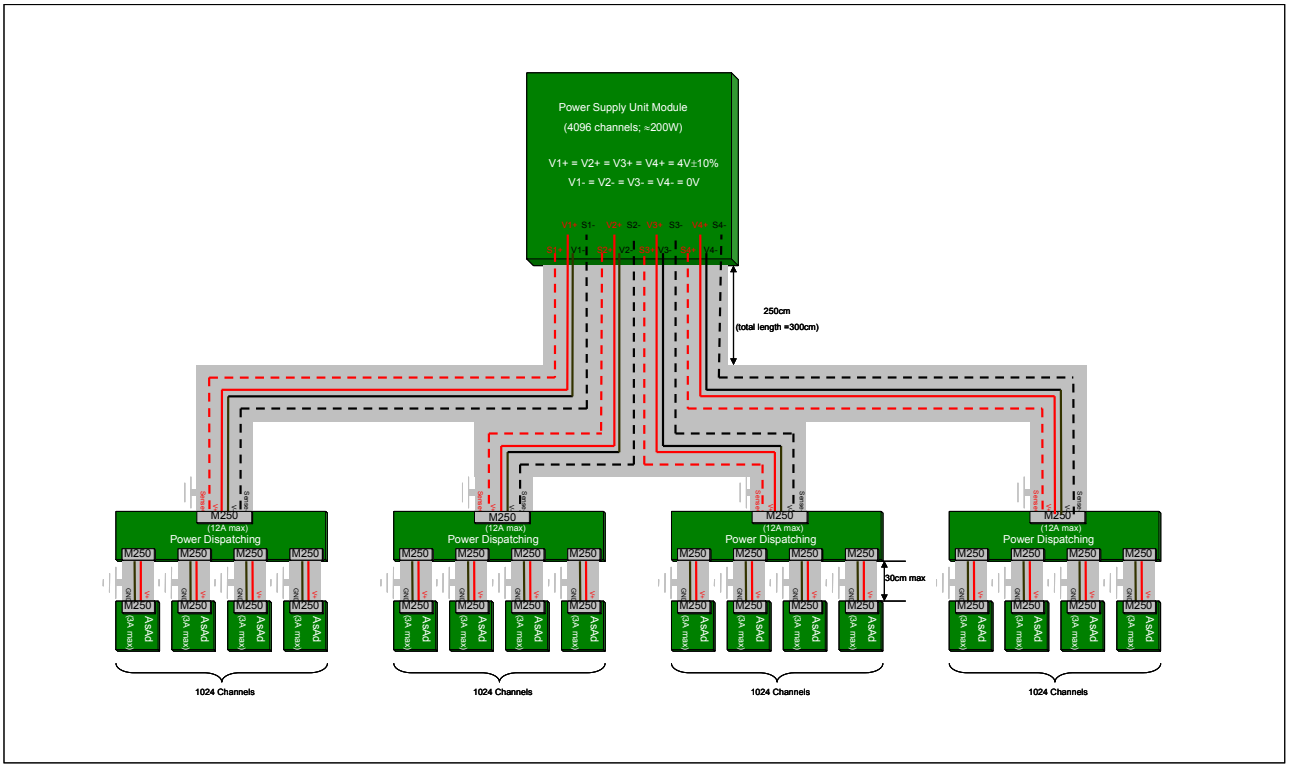


Figure 18: AsAd-GET Power Supply Unit Interconnection

Behind the power connector, AsAd uses a fuse able to protect the AsAd Power Supply block if an overload occurs.

3.3 AsAd-External Instruments Interface

The AsAd External Instruments Interface is directly linked to the Inspection Manager inputs/outputs (described in the first part of this document).
This interface is composed of a 74AC11245 chip connected to two LEMO elbow sockets for printed circuit (EPL.00.250.NTN) as shown in figure 19.

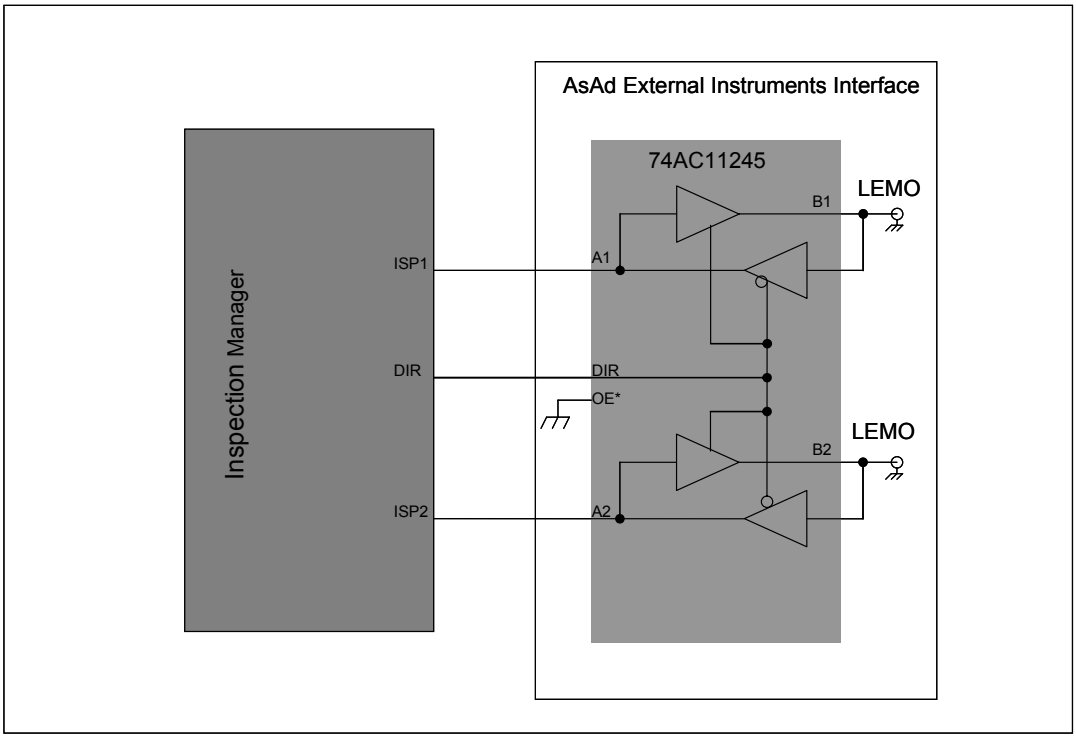


Figure 19: AsAd-External Instruments Interface

3.4 AsAd-ZAP Interface

For the AsAd-ZAP Interface a 68 contacts connector is at least required (64 contacts to connect the AGET inputs + 1 contact for the protection diode biasing + 1 contact to control the 64 AGET inputs from the detector outputs + 2 GND). Additional pins may be required depending on the amount of current needed to bias the ZAP protection diodes, and depending on the way of shielding ZAP.
To minimize the crosstalk on ZAP, the connector pitch should not be less than 1.27 mm.
A 2 rows connector would have in this case a minimal length of 43.18mm (68 contacts x 1mm pitch / 2 rows). 4 connectors of this type required for AsAd lead to an AsAd edge length that cannot be lower than 172.2mm (≈ 20cm).
Taken into account the ASIC size (based on the AFTER packaging) and the bus size to route its 64 inputs, the AsAd length can't anyway be lower than 20 cm (c. f. this document part 3 AsAd board dimensions section).
Under these conditions an ERNI connector (ref 114805) can be used as the AsAd-ZAP interface.
Nevertheless, others solutions are still studied as more than 68 contacts can be required at the AsAd-ZAP interface and as the ASIC packaging could change in a future version. Four rows connectors are particularly under the scope to achieve the AsAd-ZAP interface.

3.5 AsAd-JTAG Interface

This expert interface enables to upgrade AsAd firmware and to define an ID tag for it (c.f. Identification section in the first part of the document).
This interface description is given into an ACTEL application note.

4 PART THREE: AsAd in its physical context

4.1 AsAd board size

One of the major constraints for AsAd is to host four AGET and provide them with signals having the best immunity against noise, parasites and crosstalk.
Under these conditions the best relative placement between AGET and the AsAd-ZAP interface connector is presented in figure 20.

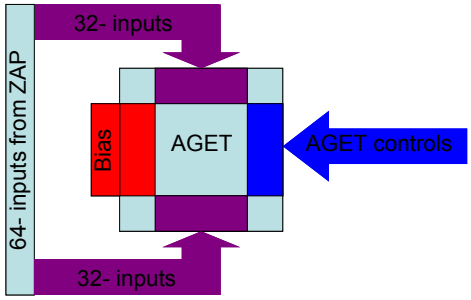


Figure 20: AGET/AsAd-ZAP Interface relative placement

In this placement configuration, there are indeed no clocks signals crossing the input signals at any level (from a board layer point of view) and the room-space taken by the biasing components do not increase AsAd edge length.
Assuming a class 5 routing for AsAd (conductors minimal width and isolation between conductors both equal to150µm) the 64 inputs bus width is approximately equal to 19.2mm (64x2x150µm). AGET itself represents a 30.2 x 30.2 mm² area, which leads to define one dimension for the physical block shown in figure 20 equal to 49.4mm (30.2mm+19.2mm). Assuming this basic pattern replicated for times, AsAd edge length cannot be lower than 197.6mm (figure 21)

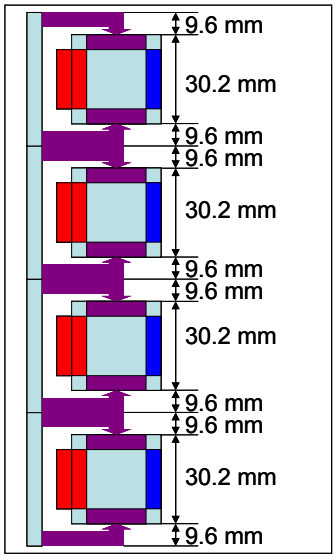


Figure 21: AsAd minimal edge length

AsAd foreseen length is thus 20 cm.
Its width is deduced from the FPGA and ADC placements on the symmetry axis of the pattern shown in figure 21. The figure 22 represents a foreseen view of the AsAd dimensions regarding to its components locations.

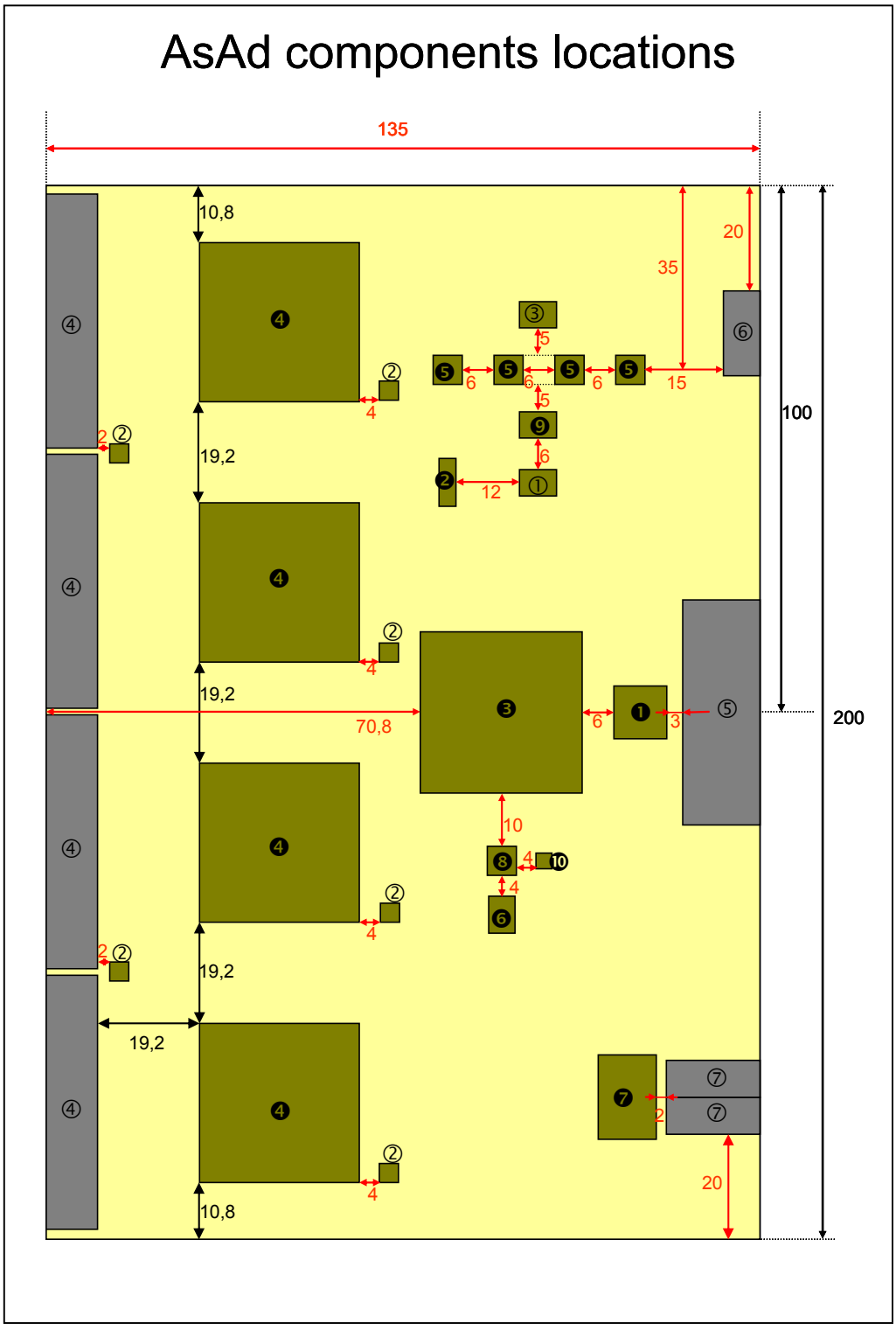


Figure 22: AsAd size and component locations

The yellow area represents AsAd size, the grey areas represent AsAd main connectors, and the green areas represent the AsAd components. Each component is referenced by a number (black background for the components dissipating power, transparent background for the others), a description for each one is given in table 23.
Dimensions may vary with the availability of the components in the chosen packages, that's the reason why some dimensions are drawn in red. AsAd width should anyway be 135mm ±10%.

Ref.	Function	Length (mm)	Width (mm)	Heigth (mm)	Power max (mW)
❶	ADC	10	10	0,9 ±0,1	1250
❷	Rsense	9,1	3,2	3,3 ±0,1	900
❸	FPGA	30,6	30,6	3,9 ±0,2	750
❹	ASIC	30,2	30,2	1,5 ±0,1	720
❺	Regulator	5,5	5,5	1 ±0,1	450
❻	Transimpedance Ampli.	7	5	1,6 ±0,15	400
❼	Buffers	16	11	2,6 ±0,1	165
❽	DAC	5,5	5,5	1 ±0,1	60
❾	Monitoring core	7	5	1,6 ±0,15	10
❿	VRef for DAC	3	3	1 ±0,15	1
①	Transimpedance Ampli.	7	5	1,6 ±0,15	0,28
②	Switches	3,65	3,65	0,9 ±0,1	0,26
③	DC/DC converter	7	5	1,6 ±0,15	0,07
④	Connector	48,2	9,7	4,2	
⑤	Connector	42,7	14,65	6	
⑥	Connector	16,1	6,9	4,6	
⑦	Connector	17,7	7	7	

Table 23: AsAd components sizes and maximal dissipated power

4.2 AsAd Module (EMC and cooling aspects)

AsAd board must be enclosed into a Faraday cage to prevent it from troubleshooting caused by electromagnetic fields. The board and its shielding constitute the AsAd module.
The manufacturing of the shielding mechanics is done by machining in the mass. Therefore, connection of 0 V to the chassis should not generate disrupting currents flows in the sensitive areas. The electrical continuity with the removable top of the module is provided by metallic screws (additional conductive seals may also be used). The AsAd modules must be connected to the mechanical ground with a load impedance as low as possible. Figure 23 shows an artistic view of the shielding module shape.

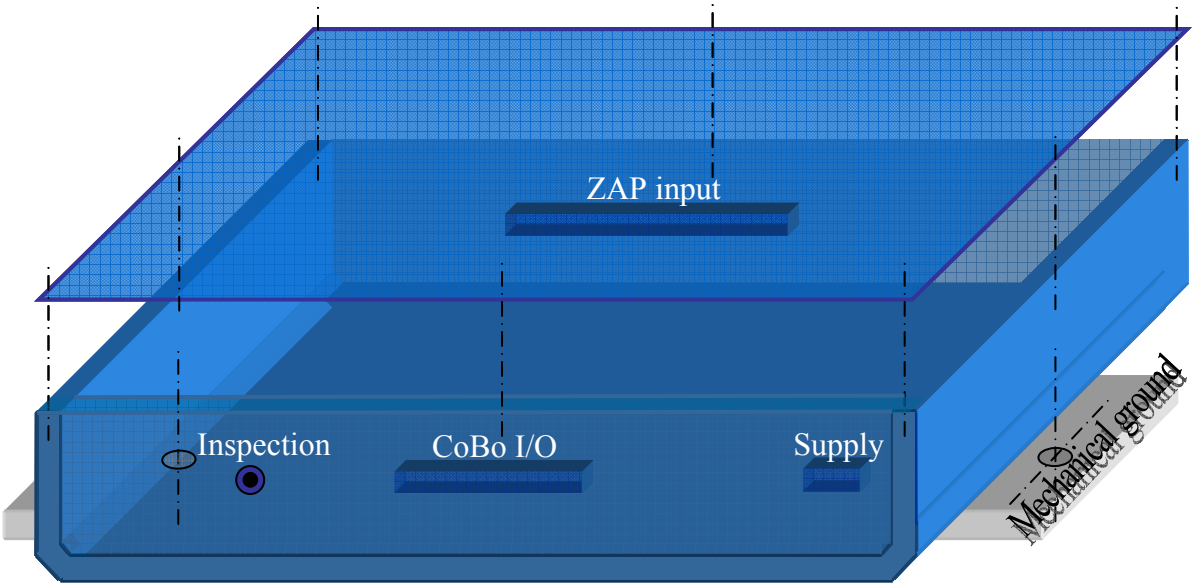


Figure 23: Shielding shape

Because of the AsAd electromagnetic environment, as well as the AsAd reduced mounting space, the use of a liquid cooling solution to dissipate the power consumed is required (An installation of liquid cooling comprises additional equipment: a pump, a flow controller, a storage tank and finally a heat exchanger).
The estimated maximal power dissipated by AsAd is deduced from table 23, it is approximately equal to 9W.
The cooling system is integrated in the top of the board housing. A thermal conductivity between AsAd components and the top of the module is provided by graphite films (figure 24). Assuming an operating temperature equal to 20° with a margin of ±4°C the liquid input/output system can be connected to 4 AsAd modules. The total height of the housing with cooling will be approximately 12 mm. Sealing is ensured by friction-stir of the metal constituting the top.

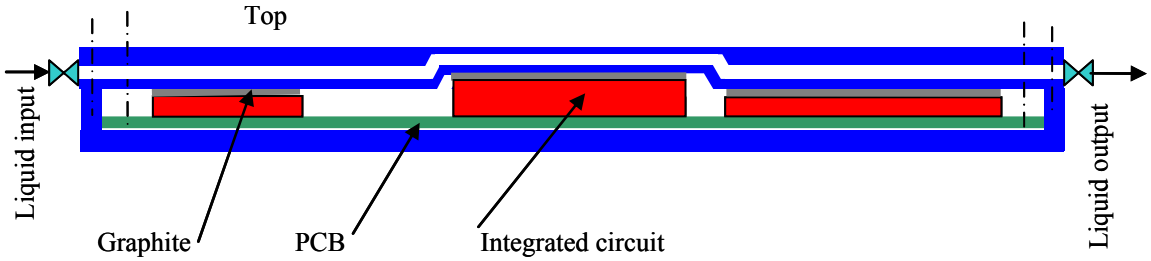


Figure 24: AsAd module cooling profile

Appendix A: AsAd functional diagram

AsAd Architecture