ALPIDE Operations Manual

ALICE ITS ALPIDE development team

July 25, 2016
Version: 0.3
Status: DRAFT

Contents

List of Figures

1 Introduction

The ALPIDE chip is a particle detector based on Monolithic Active Pixels and implemented in a 180 nm CMOS technology for Imaging Sensors. It has been designed for the Upgrade of the Inner Tracking System of the ALICE experiment at the CERN Large Hadron Collider.

The ALPIDE chip measures 15 mm (Y) by 30 mm (X) and contains a matrix of 512 ×1024 (Y×X) sensitive pixels (Fig. 1.1). The pixels are 29.24 μm × 26.88 μm (X×Y). A periphery circuit region of 1.2×30 mm^2 including the readout and control functionalities is present. It is assumed that the chip is observed from the circuits side and oriented such that the periphery is at the bottom. The pixel columns are numbered from 0 to 1023 going from left to right. Pixel rows are numbered from 0 to 511 going from the matrix top side downwards to the bottom one immediately above the periphery.

Each pixel cell contains a sensing diode, a front-end amplifying and shaping stage, a discriminator and a digital section (Fig. 1.2). The digital section includes three hit storage registers (Multi Event Buffer), a pixel masking register and pulsing logic.

The front-end and the discriminator are continuously active. They feature a non-linear response and their transistors are biased in weak inversion. Their total power consumption is 40 nW. The output of the front-end has a peaking time of the order of 2 μs, while the discriminated pulse has a typical duration of 10 μs. The front-end and the discriminator act as an analogue delay line. This allows operating the chip in triggered mode when the latency of the incoming trigger is comparable with the peaking time of the front-end.

A common threshold level is applied to all the pixels. The latching of the discriminated hits in the storage registers is controlled by global STROBE signals. A pixel hit is latched into one of three in-pixel memory cells if a STROBE pulse is applied to the selected cell while the front-end output is above threshold. Three distinct STROBE signals are generated at the periphery and globally applied to all pixels, controlling the storage of the pixel hit information in the pixel event buffers. The generation of the internal STROBE signals can be triggered by an external command (TRIGGER), but it can optionally be initiated by an internal sequencer. The duration of the STROBE pulses is programmable.

In every pixel there is a pulse injection capacitor for injection of test charge in the input of the front-end. A digital-only pulsing mode is also available, forcing the writing of a logic one in the pixel memory cells. The pulsing patterns are fully programmable.

The readout of pixel hit data from the matrix is based on a circuit named Priority Encoder. There are 512 instances of this circuit, one every two pixel columns. The Priority Encoder provides to the periphery the address of the first pixel with a hit in its double column, selecting it according to a hardwired topological priority. During one hit transfer cycle a pixel with a hit is selected, its address is generated and transmitted to the periphery and finally the in-pixel memory element is reset. The address of the next pixel with a hit in the double column is then calculated. This cycle is repeated until the addresses of all pixels initially presenting a valid hit at the inputs of a Priority Encoder have been transmitted to the periphery and all the pixel state registers have been reset. The transfer of the frame data from the matrix to the periphery is therefore zero-suppressed.

Each Priority Encoder is a fully combinatorial circuit and it is steered by sequential logic in the periphery during the readout of a matrix frame. It is implemented in a very narrow region between the pixels, extending vertically over the full height of the columns. There is no free running clock distributed in the matrix and there is no signaling activity if there are no hits to read out. The average energy needed to encode the address of a hit pixel is of

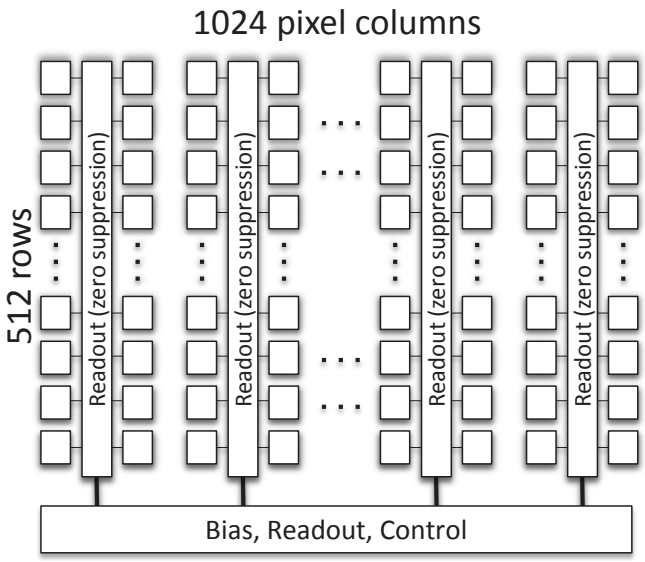

Figure 1.1: General architecture of the ALPIDE chip.

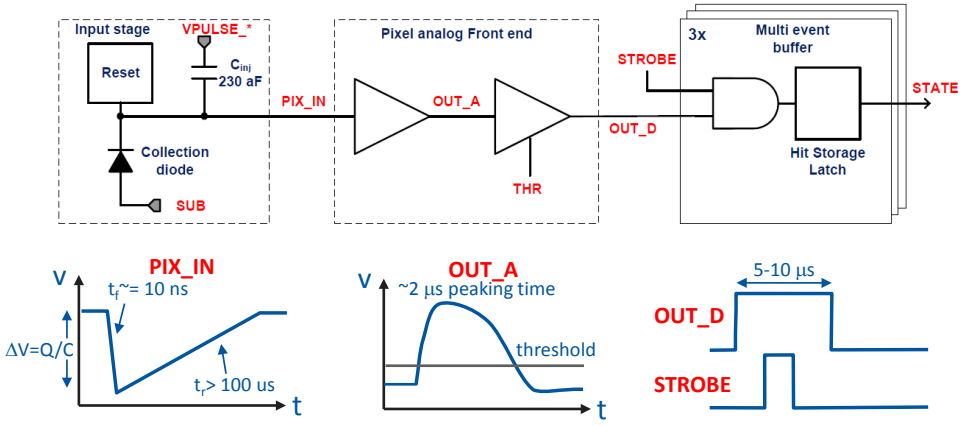

Figure 1.2: Block diagram of the ALPIDE pixel cell.

the order of 100 pJ. Power is consumed proportionally to the readout rate and to the frame occupancy. The Priority Encoders also implement the buffering and distribution of readout and configuration signals to the pixels.

The readout of the matrix is organized in 32 regions (512×32 pixels), each of them with 16 double columns being read out by 16 Priority Encoder circuits (Fig. 2.1). There are 32 corresponding readout modules (Region Readout Units) in the chip periphery, each one executing the readout of a submatrix. The sixteen Double Columns inside each region are read out sequentially, while the thirty-two submatrices are read out in parallel.

The Priority Encoders are driven by state machines in the Region Readout Units. These modules contain de-randomizing memories and perform additional data reduction and formatting. The data from the 32 region readout blocks are assembled and formatted by a Top Readout Unit module. Two major readout modes are supported, one in which the strobing and readout are triggered externally and a second one in which frames are continuously integrated and read out, with programmable duration of the strobe assertion intervals.

Hit data can be transmitted on two different data interfaces according to one of three alternative operating modes envisaged for the application in the Upgraded ALICE ITS: *Inner Barrel chip*, *Outer Barrel Master* and *Outer Barrel Slave*. A 1.2 Gb/s Serial Data port with differential signalling is the largest capacity data readout interface and the primary one for the Inner Barrel Module chips. The serial data are 8b/10b encoded, therefore the maximum data throughput is 960 Mb/s. The serial port can optionally operate at reduced line rates (600 Mb/s or 400 Mb/s).

The same interface is intended to be used for the transmission of data off-detector by the Outer Barrel Master chips, using a bit rate of 400 Mb/s (320 Mb/s payload). The Master chips also collect the data of a set of neighboring Outer Barrel Slave chips and forward their data off-detector on the differential link.

A bidirectional parallel data port with single-ended signaling is also present, with a capacity of 320 Mb/s. It enables the implementation of the data exchange between the Outer Barrel Slave chips and the corresponding Master. All the functionalities related to the communication between Master and Slave chips on the parallel bus are implemented in the module called Data Management Unit.

The ALPIDE chip has custom control interfaces. There is a differential control port (DCNTRL) supporting bi-directional (half duplex) serial signaling at 40 Mb/s on differential links. A second single ended control line (CNTRL) is also available. These interfaces and the related control logic enable the interconnection of multiple chips on a multi-point control bus with a hierarchical topology, with control transactions relayed by the Master chips to Slave chips. The Control Management Unit block implements the control layer and provides full access to the control and status registers of the chip as well as to the multi-event memories in the Region Readout Units. The control bus is also used to distribute commands to the chips, most notably the trigger messages.

All the analog signals required by the front-ends are generated by a set of on-chip 8 bit DACs. Analog monitoring pads (DACMONV, DACMONI) are available to monitor the outputs of the internal DACs. The DACMONV pad can be used to override any of the voltage DACs. The DACMONI pad can be used to override any of the current DACs or to override the internal reference current used by the current DACs.

The analog section of the periphery also contains an ADC with 10-bit dynamic range, a bandgap voltage reference and a temperature sensing circuit. The ADC can be used to monitor several quasi-static internal analog signals: the outputs of the DACs, the analog and digital supply voltages, the bandgap voltage and the temperature sensor.

2 ALPIDE data sheet

2.1 Block diagrams

A general block diagram of the ALPIDE chip is given in Fig. 2.1.

A block diagram including details on the internal structures, module and peripheral hard blocks is given in Fig. 2.2.

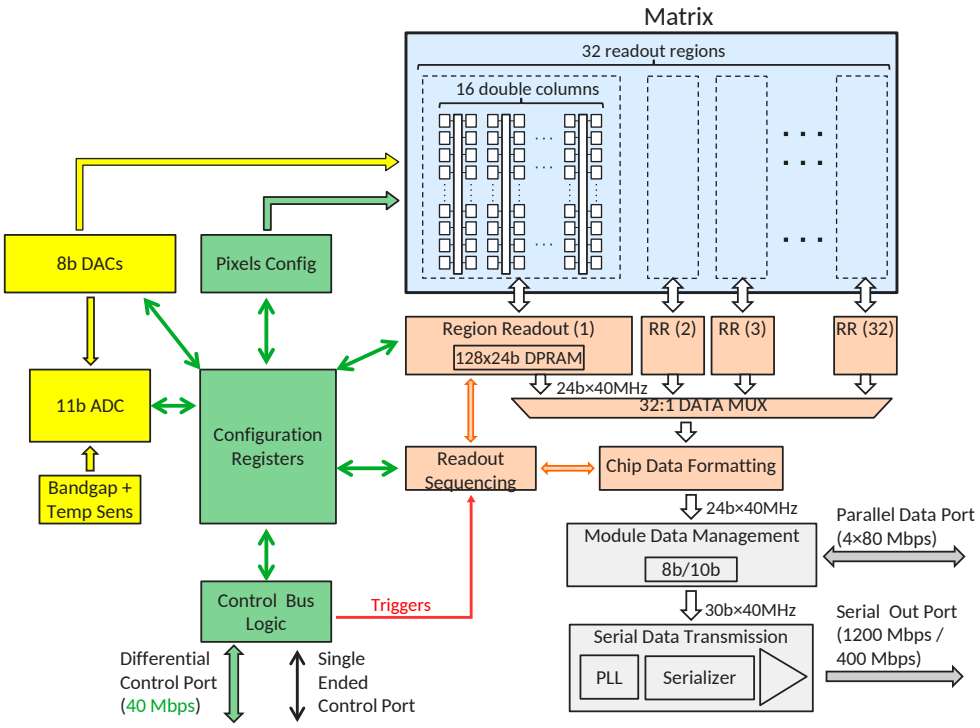

Figure 2.1: ALPIDE chip block diagram.

2.2 Interface signals

The main functional I/Os of the ALPIDE chip are listed in Table 2.1. Figure 2.3 shows the pinout of the chip with the locations of the pads.

The CMOS I/Os are 1.8 V compatible. Two types of CMOS I/O pad cells are used in ALPIDE: one has an internal pull-up resistor and one an internal pull-down resistor. The internal resistors are always connected to the pad. The driving strengths of the two cells are equal and fixed. The pad cells are tri-state capable and their drivers can be turned off and placed in a high-impedance mode depending on configuration and conditions. The internal pull-up or pull-down resistors remain connected and active in the high-impedance state.

The MCLK, DCTRL ans DCLK differential ports are implemented with a custom designed differential transceiver block. This has been designed with reference to standard TIA/EIA-899 Electrical Characteristics of Multipoint-Low-Voltage Differential Signaling (M-LVDS)[1].

[1] See Texas Instrument Application Report SLLA108A

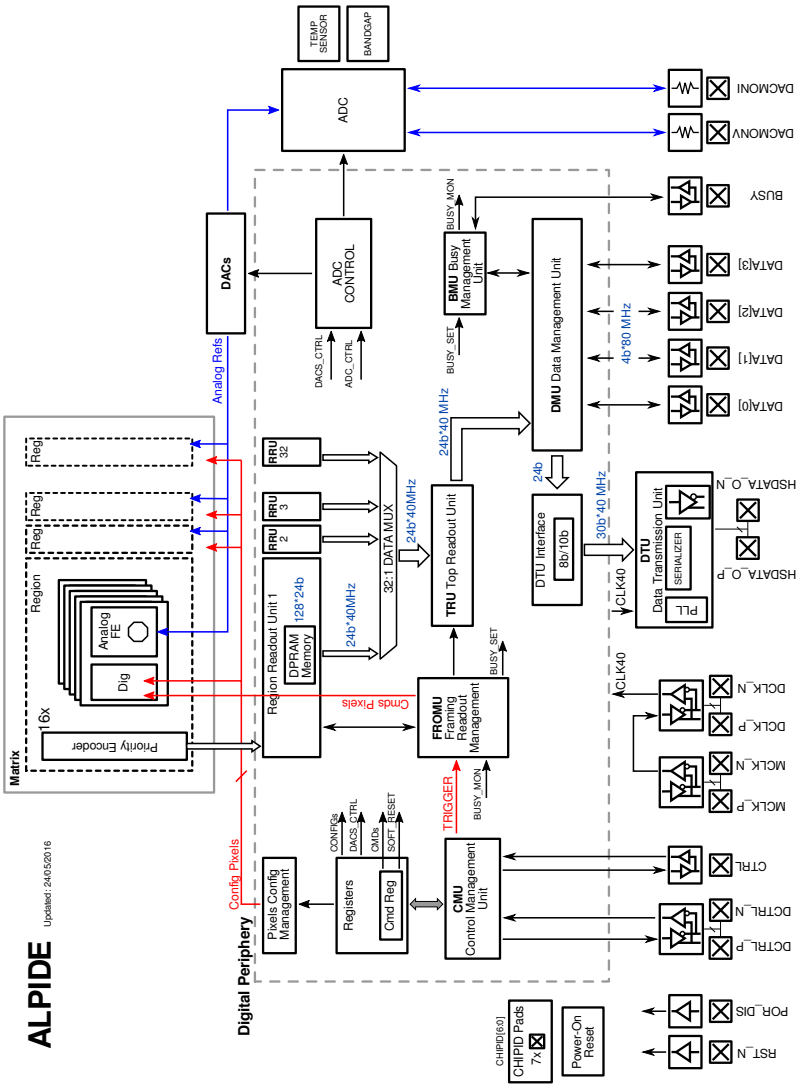

Figure 2.2: ALPIDE chip block diagram with modules and peripheral hard-blocks.

However the differential ports are not standard compliant in particular with respect to the allowed range of the input common voltage.

Tables 2.3, Table 2.4 and Table 2.5 summarize the recommended DC operating conditions and the electrical characteristics of the various interfaces.

The analog monitoring ports provide access to internal nodes through a series resistor.

Table 2.1: ALPIDE interface signals.

Signal	Type	Direction	Purpose
MCLK_P	Differential (MLVDS)	INPUT	Forwarded clock input
MCLK_N	Differential (MLVDS)	INPUT	Forwarded clock input
RST_N	CMOS, internal pull-up	INPUT	Global chip reset
POR_DIS_N	CMOS, internal pull-up	INPUT	Power On Reset Disable
DCTRL_P	Differential (MLVDS)	BIDIR	Differential Control port
DCTRL_N	Differential (MLVDS)	BIDIR	Differential Control port
DCLK_P	Differential (MLVDS)	BIDIR	Main clock input

			and clock forwarding output
DCLK_N	Differential (MLVDS)	BIDIR	Main clock input and clock forwarding output
HSDATA_P	Differential (LVDS)	OUTPUT	Serial Data Output
HSDATA_N	Differential (LVDS)	OUTPUT	Serial Data Output
CTRL	CMOS, internal pull-up	BIDIR	Control port (OB local bus)
DATA[7]	CMOS, internal pull-up	BIDIR	Data port (OB local bus)
DATA[6]	CMOS, internal pull-up	BIDIR	Data port (OB local bus)
DATA[5]	CMOS, internal pull-up	BIDIR	Data port (OB local bus)
DATA[4]	CMOS, internal pull-up	BIDIR	Data port (OB local bus)
DATA[3]	CMOS, internal pull-up	BIDIR	Data port (OB local bus)
DATA[2]	CMOS, internal pull-up	BIDIR	Data port (OB local bus)
DATA[1]	CMOS, internal pull-up	BIDIR	Data port (OB local bus)
DATA[0]	CMOS, internal pull-up	BIDIR	Data port (OB local bus)
BUSY	CMOS, internal pull-up	BIDIR	Busy flag
DACMONV	ANALOG	OUTPUT	Voltage Monitoring Output
DACMONI	ANALOG	OUTPUT	Current Monitoring Output
CHIPID[6]	CMOS, internal pull-down	INPUT	Topological chip address
CHIPID[5]	CMOS, internal pull-down	INPUT	Topological chip address
CHIPID[4]	CMOS, internal pull-down	INPUT	Topological chip address
CHIPID[3]	CMOS, internal pull-down	INPUT	Topological chip address
CHIPID[2]	CMOS, internal pull-down	INPUT	Topological chip address
CHIPID[1]	CMOS, internal pull-down	INPUT	Topological chip address
CHIPID[0]	CMOS, internal pull-down	INPUT	Topological chip address

MCLK_P, MCLK_N: Clock forwarding input port, used to implement the clock distribution in the Outer Barrel application scenario. This is a receiving only port, the driver behind it being disabled in all scenarios. The receiver is enabled when the chip is configured as Outer Barrel Master and the signal applied to this port is then forwarded to driver of the DCLK_P, DCLK_N port. A chip configured as Inner Chip or Outer Barrel slave chip keeps the receiver on this port disabled (refer to appendix A).

RST_N: Global active-low reset signal. This port can be left unconnected in applications not needing a dedicated reset pin. The ALPIDE chip includes a power-on-reset circuit. The chip can also be reset by commands issued by the control interface.

POR_DIS_N: Disabling of the power-on-reset circuit, active low. Driving low this input masks the output of the internal power-on reset circuitry. If the internal power-on-reset is used this pin can be left unconnected since it is internally pulled-up.

DCTRL_P, DCTRL_N: Differential bidirectional control port. Intended to implement the segments of the control bus between the Inner Barrel chips or the Outer Barrel Master chips and the off detector electronics. The DCTRL port is unused by a chip configured as Outer Barrel Slave Chip. The communication through this port is half-duplex. Signals are received or driven but not simultaneously.

DCLK_P, DCLK_N: Main clock input and forwarded clock output. The nominal clock frequency is the LHC bunch frequency, 40.08 MHz. This is the chip clock source regardless of the operating mode and configuration scenario. In all configurations the receiver circuit at this port provides the clock to the chip core. A chip configured as Outer Barrel Master has an active driver on this port and forwards on it the signal received on the MCLK_P, MCLK_N port. This port is also equipped with an on-chip termination resistor (100 Ω) that is enabled depending on the signals applied to the CHIPID configuration port. Additional details concerning the supported clocking schemes can be found in appendix A.

HSDATA_P, HSDATA_N: Differential data output port. This port is used for the high speed serial transmission of data between chips and the off-detector electronics. It is used by chips configured as Inner Barrel Chip or Outer Barrel Master. The signaling rate on this

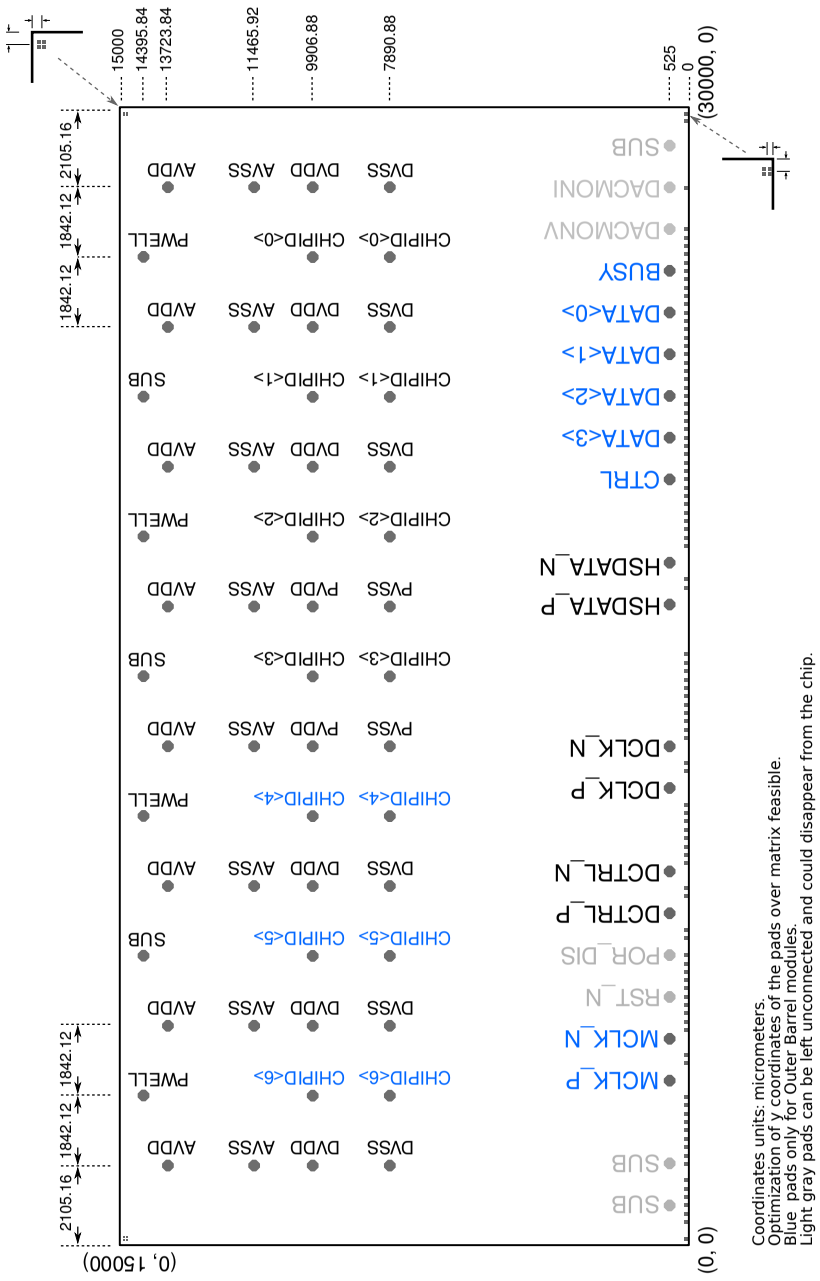

Figure 2.3: Pinout of the ALPIDE chip.

port is programmable in the Inner Barrel Chip operating mode, selecting between 1.2 Gb/s (default), 600 Mb/s or 400 Mb/s. The signaling rate is 400 Mb/s in the Outer Barrel Master configuration. The serial stream is 8b/10b encoded.

CTRL: Single ended, bidirectional control port. Intended to implement the local control bus segments between the Outer Barrel Master chip and the associated slaves. These chips shall have their CTRL ports directly connected by a single shared wire. The CTRL port is unused by a chip configured as Inner Barrel Chip. The communication through this port is half-duplex. Signals are received or driven but not simultaneously.

DATA[7:0]: CMOS bidirectional data port. Intended to implement a shared parallel data bus between the Outer Barrel Slave chips and the associated Master chip. By default, the 4 lowermost lines of this port operate in Double Data Rate mode, with bits launched or sampled at both clock edges and one complete byte transfer completed at every clock cycle. Thus the uppermost 4 bits can be left unconnected and the bus can be implemented using 4 parallel wires shared by the chips. Optionally, the chips can be configured to revert to Single Data Rate signaling also on the lowermost 4 bits. In this case one byte is launched or sampled at every rising edge of the clock. This operating mode can be used for readout of chips through a 8 bit Single Data Rate parallel bus.

BUSY: Single ended port. It is intended to implement the communication of the BUSY state between the Outer Barrel Slaves and the associated Master chip by wiring in parallel all their BUSY ports. This port is not used when the chip operates as an ITS Inner Barrel chip. This port can be in one of two states: actively driven low or high impedance, thus emulating an open-drain topology. The signaling is active low. The pad provides weak internal pull-up. An external strong pull-up resistor might be required to speed-up the rise-time of the de-assertion (rising) edge depending on the total capacitance of the line and the number of chips connected to it. The sampling of the input on this port is equipped with a sychronizer.

DACMONV: Analog pin with dual functionality. It can be used to monitor each of the voltages generated by the on-chip voltage DACs. It can also be used to override the internal voltage DACs. The overriding is possible for one user-selectable DAC at a given time.

DACMONI: Analog pin with triple functionality. (a) Monitoring of the currents generated by the on-chip current DACs. (b) Override of the internal current DACs. The overriding is possible for one user-selectable DAC at a given time. (c) Override of the chip internal current reference, thus changing the range of all current DACs simultaneously.

CHIPID[6:0]: Chip topological address and mode selection. This port is intended to assign a binary coded address to each chip depending on its position on the ALICE ITS Modules. The address is used in the transactions via the control interface. The address value also selects if the chip behaves as a Inner Barrel Chip, an Outer Barrel Master chip or an Outer Barrel Slave chip. These pads have been designed to be directly wired to digital supply in order to set a binary '1' on intended lines. Leaving a pad unconnected effectively sets to '0' the corresponding input by the internal pull-down.

2.3 Supply, ground and bias nets

Table 2.2: ALPIDE supply, ground and bias nets.

Net	Type	Purpose
AVDD	SUPPLY	Analog domain supply
AVSS	GROUND	Analog domain ground
DVDD	SUPPLY	Digital domain supply
DVSS	GROUND	Digital domain ground
PVDD	SUPPLY	Data Transmission Unit PLL supply
PVSS	GROUND	Data Transmission Unit PLL ground
PWELL	SUBSTRATE	Substrate bias

SUB	SUBSTRATE	Substrate bias

AVDD, AVSS: Supply and ground nets of the analog domain. This includes the pixel front-end circuits, the analog biasing circuits (DACs), the ADC block.

DVDD, DVSS: Supply and ground nets of the digital domain. This includes the in-pixel configuration registers, the matrix readout circuits, the peripheral readout circuits and the chip input and output buffers and transceivers.

PDVDD, PDVSS: Supply and ground nets exclusively dedicated to the Phase Locked Loop of the Data Transmission Unit.

PWELL: bias of the p-type wells in the pixel matrix region.

SUB: bias to the substrate contacts in the seal ring and in the periphery region.

All supply and ground nets must be connected to the recommended operating voltages. The pads of each supply or ground net are internally electrically connected by the on-chip supply and ground meshes. Therefore it is not mandatory to wire all the pads of a given supply or ground net. However, a reduction of the number of connected supply or ground pads can have detrimental effects on the circuit performance and operating capabilities or increase the risk of damaging the chip.

The PWELL and SUB bias nets cannot be left floating and must be strongly connected to appropriate bias voltages. The source impedance of the supply to these nets shall be kept as low as possible to limit the probability of latch-up.

The pads of the PWELL net are internally connected as well as those of the SUB net, therefore it is not mandatory to wire all the pads of a given net. The PWELL and SUB nets are electrically connected to each other through the conductance of the die substrate.

The purpose of the PWELL and SUB substrate biasing nets is to enable the increase of the reverse bias voltage on the charge collecting diodes. This is obtained by applying to these nets a negative voltage with respect to analog ground (AVSS). For applications not demanding enhanced sensor performance it is possible to short both the PWELL and SUB pads to the AVSS ground (0 V with respect to AVSS).

It is recommended to bond the PWELL and SUB pads to a grounding conductor before any other pad is bonded, to mitigate the risk of ESD damage.

2.4 Recommended operating conditions

Table 2.3: Recommended operating conditions.

		MIN	TYP	MAX	Unit	Condition
AVSS	Analog ground		0		V	
AVDD	Analog supply	1.62	1.8	1.98	V	
DVSS	Digital core ground		0		V	
DVDD	Digital core supply	1.62	1.8	1.98	V	
PVSS	PLL ground		0		V	
PVDD	PLL supply	1.62	1.8	1.98	V	
PWELL	Substrate bias	-6		0	V	
SUB	Substrate bias	-6		0	V	
V_I	Voltage at any CMOS input	0		DVDD	V	
V_{IL}	Low level digital input voltage			0.33*DVDD	V	
V_{IH}	High level digital input voltage	0.66*DVDD			V	
I_{OL}	Low level digital output current			13.7(?)	mA	Vo <0.45
$\|I_{OH}\|$	High level digital			13.6(?)	mA	Vo >DVDD-0.45

	output current					
V_P or V_N	Voltage at any differential bus terminal	0		DVDD	V	
$\|V_{ID}\|$	Magnitude of differential input voltage	50		DVDD	mV	
R_L	Differential load resistance	40 (?)	50(?)	60(?)	Ω	
T	Operating temperature	-25	25	85	℃	

2.5 Electrical characteristics

Table 2.4: CMOS IOs electrical characteristics over recommended operating conditions unless otherwise noted.

		MIN	TYP	MAX	Unit	Condition
V_{OH}	High level output voltage	DVDD-0.45			V	Io >-13.6 mA
V_{OL}	Low level output voltage			0.45	V	Io <13.7 mA
$\|I_{IL}\|$			44.4	62	μA	Pads with pull-up
I_{IH}			0.87	3.8	nA	Pads with pull-up
$\|I_{IL}\|$			3.3	11.5	nA	Pads with pull-down
I_{IH}			44.4	62	μA	Pads with pull-down
R_{Pullup}	Internal pull-up		40.6		kΩ	Pads with pull-up
R_{Pullup}	Internal pull-down		40.6		kΩ	Pads with pull-down
C_{PAD}	Input capacitance		0.98		pF	Inputs with A type pad only
C_{PAD}	Input capacitance		4.5		pF	Inputs with A and B type pads
R_s	Output series resistance		(?)		Ω	

Table 2.5: Electrical characteristics of MCLK, DCTRL and DCLK differential ports over recommended operating conditions unless otherwise noted.

		MIN	TYP	MAX	Unit	Condition
C_P or C_N	Input capacitance			3	pF	V_I, other pin at 1.1 V, driver disabled
C_{PN}	Differential input capacitance				pF	V_{ID} =, V_{IC} =, driver disabled
I_{OZ}	High-impedance state output current	-45		102	μA	Driver disabled
Driver related characteristics						
		MIN	TYP	MAX	Unit	Condition
$\|V_{OD}\|$	Output differential signal magnitude	80 (?)		480(?)	mV	
$\|I_{OD}\|$	Output differential current magnitude	2		8	mA	
$V_{OS(SS)}$	Steady-state output common signal	980		1210	mV	
$\Delta V_{OS(SS)}$	Change in steady-state output common signal between logic states	-20		+20	mV	
$V_{OS(PP)}$	Peak-to-peak output common signal			75	mV	
Receiver related characteristics						
		MIN	TYP	MAX	Unit	Condition
V_{IC}	Input common signal	DVSS+25	1.1	DVDD-25	mV	
V_{IT+}	Positive-going differential input voltage threshold			50	mV	
V_{IT-}	Negative-going differential	-50			mV	

	input voltage threshold			

2.6 Pad tables, geometrical data, alignment markers

A floorplan view with the name of the pad nets at the pads used for the connection to the ALICE ITS FPCs is given in Figure 2.3. The pad naming convention and the layout of the die with the position of the pads are illustrated in Figure 2.4.

Table 2.6 lists all the pads and interface nets of the chip. Table 2.7 gives the x and y coordinates of the *center* points of the chip pads.

Two types of pads with differing geometries are employed in the ALPIDE chip.

Type A pads are used for the pads in the pad ring along the chip south edge (from A00 to A94). All interface nets are available in this pad ring. Type A pads are standard size and intended to support wire bonding, probe testing or other applications. Figure 2.5 details the geometry of the opening of type A pads. The opening in the passivation layer of type A pads is square and 88 μm wide.

Type B are large pads over logic used above the periphery and the sensitive matrix of the chip (B00-B20, C00-C14, D00-D14, E00-E07, F00-F07, G00-G06). These pads enable the connection to Flexible Printed Circuits. The nets accessible through pads of type B are also found in the ring of pads of type A. Figure 2.6 details the geometry of the opening of pads of type B. The opening in the passivation layer of type B pads has rounded edges and it is 290 μm wide.

The chip includes four structures located close to the four corners of the die to facilitate geometrical alignment procedures, as shown in Figure 2.4 for the markers on the top-right and bottom-right corners. Figure 2.7 details the geometry of the alignment markers. The coordinates of the central point of the alignment markers are listed in table 2.8.

Table 2.6: ALPIDE pads and interface nets.

Pad Id	Net	Type	Direction	Purpose
A00	PWELL	SUBSTRATE		Substrate bias, Matrix
A01	AVSS	GROUND		Analog ground
A02	SUB	SUBSTRATE		Substrate bias, periphery
A03	AVDD	SUPPLY		Analog supply
A04	DVSS	GROUND		Digital ground
A05	DVDD	SUPPLY		Digital supply
A06	SUB	SUBSTRATE		Substrate bias, periphery
A07	AVSS	GROUND		Analog ground
A08	AVDD	SUPPLY		Analog supply
A09	DVSS	GROUND		Digital ground
A10	DVDD	SUPPLY		Digital supply
A11	SCI	CMOS		Unused
A12	AVSS	GROUND		Analog ground
A13	AVDD	SUPPLY		Analog supply
A14	CHIPID[6]	CMOS, pull-down	INPUT	Topological chip address
A15	MCLK_P	MLVDS	INPUT	Forwarded clock input
A16	MCLK_N	MLVDS	INPUT	Forwarded clock input
A17	DVSS	GROUND		Digital ground
A18	DVDD	SUPPLY		Digital supply
A19	PWELL	SUBSTRATE		Substrate bias, Matrix
A20	RESERVE_0	CMOS		Unused
A21	RST_N	CMOS, pull-up	INPUT	Global Hard Reset
A22	RESERVE_2	CMOS		Unused
A23	DVSS	GROUND		Digital ground
A24	DVDD	SUPPLY		Digital supply
A25	CHIPID[5]	CMOS, pull-down	INPUT	Topological chip address
A26	POR_DIS_N	CMOS, pull-up	INPUT	Disable Power-On Reset
A27	AVSS	GROUND		Analog ground
A28	AVDD	SUPPLY		Analog supply
A29	DCTRL_P	MLVDS	BIDIR	Differential Control port
A30	DCTRL_N	MLVDS	BIDIR	Differential Control port
A31	DVSS	GROUND		Digital ground
A32	DVDD	SUPPLY		Digital supply
A33	SUB	SUBSTRATE		Substrate bias, periphery
A34	AVSS	GROUND		Analog ground

A35	AVDD	SUPPLY		Analog supply
A36	SCO	CMOS		Unused
A37	CHIPID[4]	CMOS, pull-down	INPUT	Topological chip address
A38	DVSS	GROUND		Digital ground
A39	DCLK_P	MLVDS	BIDIR	Clock input. Clock forwarding output
A40	DCLK_N	MLVDS	BIDIR	Clock input. Clock forwarding output
A41	DVDD	SUPPLY		Digital supply
A42	PWELL	SUBSTRATE		Substrate bias, Matrix
A43	DVSS	GROUND		Digital ground
A44	DVDD	SUPPLY		Digital supply
A45	AVSS	GROUND		Analog ground
A46	AVDD	SUPPLY		Analog supply
A47	CHIPID[3]	CMOS, pull-down	INPUT	Topological chip address
A48	PVSS	GROUND		PLL ground
A49	PVDD	SUPPLY		PLL supply
A50	PVSS	GROUND		PLL ground
A51	PVDD	SUPPLY		PLL supply
A52	HSDATA_P	LVDS	OUTPUT	Serial Data Port
A53	HSDATA_N	LVDS	OUTPUT	Serial Data Port
A54	DVDD	SUPPLY		Digital supply
A55	CHIPID[2]	CMOS, pull-down	INPUT	Topological chip address
A56	DVSS	GROUND		Digital ground
A57	SCE	CMOS		Unused
A58	AVSS	GROUND		Analog ground
A59	AVDD	SUPPLY		Analog supply
A60	DVSS	GROUND		Digital ground
A61	DVDD	SUPPLY		Digital supply
A62	CTRL	CMOS, pull-up	BIDIR	Control Port (OB)
A63	DVSS	GROUND		Digital ground
A64	DVDD	SUPPLY		Digital supply
A65	DVSS	GROUND		Digital ground
A66	DVDD	SUPPLY		Digital supply
A67	DATA[3]	CMOS, pull-up	BIDIR	Data port
A68	DATA[7]	CMOS, pull-up	BIDIR	Data port
A69	DVSS	GROUND		Digital ground
A70	DVDD	SUPPLY		Digital supply
A71	CHIPID[1]	CMOS, pull-down	INPUT	Topological chip address
A72	DATA[2]	CMOS, pull-up	BIDIR	Data port
A73	DATA[6]	CMOS, pull-up	BIDIR	Data port
A74	DVSS	GROUND		Digital ground
A75	DVDD	SUPPLY		Digital supply
A76	DVSS	GROUND		Digital ground
A77	DATA[1]	CMOS, pull-up	BIDIR	Data port
A78	DATA[5]	CMOS, pull-up	BIDIR	Data port
A79	DVDD	SUPPLY		Digital supply
A80	AVSS	GROUND		Analog ground
A81	AVDD	SUPPLY		Analog supply
A82	DATA[0]	CMOS, pull-up	BIDIR	Data port
A83	DATA[4]	CMOS, pull-up	BIDIR	Data port
A84	SUB	SUBSTRATE		Substrate bias, periphery
A85	DVSS	GROUND		Digital ground
A86	DVDD	SUPPLY		Digital supply
A87	BUSY	CMOS, pull-up	BIDIR	Busy Flag
A88	CHIPID[0]	CMOS, pull-down	INPUT	Topological chip address
A89	AVSS	GROUND		Analog ground
A90	AVDD	SUPPLY		Analog supply
A91	PWELL	SUBSTRATE		Substrate bias, Matrix
A92	DACMONV	ANALOG		Voltage monitoring and overriding
A93	DACMONI	ANALOG		Current monitoring and overriding
A94	SUB	SUBSTRATE		Substrate bias, periphery

B00	SUB	SUBSTRATE		Substrate bias, periphery
B01	SUB	SUBSTRATE		Substrate bias, periphery
B02	MCLK_P	MLVDS	INPUT	Forwarded clock input
B03	MCLK_N	MLVDS	INPUT	Forwarded clock input
B04	RST_N	CMOS, pull-up	INPUT	Global Hard Reset
B05	POR_DIS_N	CMOS, pull-up	INPUT	Disable Power-On Reset
B06	DCTRL_P	MLVDS	BIDIR	Differential Control port
B07	DCTRL_N	MLVDS	BIDIR	Differential Control port
B08	DCLK_P	MLVDS	BIDIR	Clock input. Clock forwarding output
B09	DCLK_N	MLVDS	BIDIR	Clock input. Clock forwarding output
B10	HSDATA_P	LVDS	OUTPUT	Serial Data Port
B11	HSDATA_N	LVDS	OUTPUT	Serial Data Port
B12	CTRL	CMOS, pull-up	BIDIR	Control Port (OB)
B13	DATA[3]	CMOS, pull-up	BIDIR	Data port
B14	DATA[2]	CMOS, pull-up	BIDIR	Data port
B15	DATA[1]	CMOS, pull-up	BIDIR	Data port
B16	DATA[0]	CMOS, pull-up	BIDIR	Data port
B17	BUSY	CMOS, pull-up	BIDIR	Busy Flag
B18	DACMONI	ANALOG		Current monitoring and overriding
B19	DACMONV	ANALOG		Voltage monitoring and overriding
B20	SUB	SUBSTRATE		Substrate bias, periphery
C00	DVSS	GROUND		Digital ground
C01	CHIPID[6]	CMOS, pull-down	INPUT	Topological chip address
C02	DVSS	GROUND		Digital ground
C03	CHIPID[5]	CMOS, pull-down	INPUT	Topological chip address
C04	DVSS	GROUND		Digital ground
C05	CHIPID[4]	CMOS, pull-down	INPUT	Topological chip address
C06	PVSS	GROUND		PLL ground
C07	CHIPID[3]	CMOS, pull-down	INPUT	Topological chip address
C08	PVSS	GROUND		PLL ground
C09	CHIPID[2]	CMOS, pull-down	INPUT	Topological chip address
C10	DVSS	GROUND		Digital ground
C11	CHIPID[1]	CMOS, pull-down	INPUT	Topological chip address
C12	DVSS	GROUND		Digital ground
C13	CHIPID[0]	CMOS, pull-down	INPUT	Topological chip address
C14	DVSS	GROUND		Digital ground
D00	DVDD	SUPPLY		Digital supply
D01	CHIPID[6]	CMOS, pull-down	INPUT	Topological chip address
D02	DVDD	SUPPLY		Digital supply
D03	CHIPID[5]	CMOS, pull-down	INPUT	Topological chip address
D04	DVDD	SUPPLY		Digital supply
D05	CHIPID[4]	CMOS, pull-down	INPUT	Topological chip address
D06	PVDD	SUPPLY		PLL supply
D07	CHIPID[3]	CMOS, pull-down	INPUT	Topological chip address
D08	PVDD	SUPPLY		PLL supply
D09	CHIPID[2]	CMOS, pull-down	INPUT	Topological chip address
D10	DVDD	SUPPLY		Digital supply
D11	CHIPID[1]	CMOS, pull-down	INPUT	Topological chip address
D12	DVDD	SUPPLY		Digital supply
D13	CHIPID[0]	CMOS, pull-down	INPUT	Topological chip address
D14	DVDD	SUPPLY		Digital supply
E00	AVSS	GROUND		Analog ground
E01	AVSS	GROUND		Analog ground
E02	AVSS	GROUND		Analog ground
E03	AVSS	GROUND		Analog ground
E04	AVSS	GROUND		Analog ground
E05	AVSS	GROUND		Analog ground
E06	AVSS	GROUND		Analog ground

E07	AVSS	GROUND	Analog ground
F00	AVDD	SUPPLY	Analog supply
F01	AVDD	SUPPLY	Analog supply
F02	AVDD	SUPPLY	Analog supply
F03	AVDD	SUPPLY	Analog supply
F04	AVDD	SUPPLY	Analog supply
F05	AVDD	SUPPLY	Analog supply
F06	AVDD	SUPPLY	Analog supply
F07	AVDD	SUPPLY	Analog supply
G00	PWELL	SUBSTRATE	Substrate bias, Matrix
G01	SUB	SUBSTRATE	Substrate bias, periphery
G02	PWELL	SUBSTRATE	Substrate bias, Matrix
G03	SUB	SUBSTRATE	Substrate bias, periphery
G04	PWELL	SUBSTRATE	Substrate bias, Matrix
G05	SUB	SUBSTRATE	Substrate bias, periphery
G06	PWELL	SUBSTRATE	Substrate bias, Matrix

Table 2.7: Coordinates of the center points of the pads.

Pad Id	Net	Pad Geometry	x [μm]	y [μm]
A00	PWELL	A	607.62	66.8
A01	AVSS	A	827.62	66.8
A02	SUB	A	1047.62	66.8
A03	AVDD	A	1267.62	66.8
A04	DVSS	A	1487.62	66.8
A05	DVDD	A	1707.62	66.8
A06	SUB	A	2147.62	66.8
A07	AVSS	A	2367.62	66.8
A08	AVDD	A	2587.62	66.8
A09	DVSS	A	2807.62	66.8
A10	DVDD	A	3027.62	66.8
A11	SCI	A	3247.62	66.8
A12	AVSS	A	3467.62	66.8
A13	AVDD	A	3687.62	66.8
A14	CHIPID[6]	A	3907.62	66.8
A15	MCLK_P	A	4797.62	62.755
A16	MCLK_N	A	5017.62	62.755
A17	DVSS	A	5667.62	66.8
A18	DVDD	A	5887.62	66.8
A19	PWELL	A	6107.62	66.8
A20	RESERVE_0	A	6327.62	66.8
A21	RST_N	A	6547.62	66.8
A22	RESERVE_2	A	6767.62	66.8
A23	DVSS	A	6987.62	66.8
A24	DVDD	A	7207.62	66.8
A25	CHIPID[5]	A	7427.62	66.8
A26	POR_DIS_N	A	7647.62	66.8
A27	AVSS	A	8087.62	66.8
A28	AVDD	A	8307.62	66.8
A29	DCTRL_P	A	9197.62	62.755
A30	DCTRL_N	A	9417.62	62.755
A31	DVSS	A	10067.62	66.8
A32	DVDD	A	10287.62	66.8
A33	SUB	A	10507.62	66.8
A34	AVSS	A	10727.62	66.8
A35	AVDD	A	10947.62	66.8
A36	SCO	A	11167.62	66.8
A37	CHIPID[4]	A	11387.62	66.8

A38	DVSS	A	11607.62	66.8
A39	DCLK_P	A	12497.62	62.755
A40	DCLK_N	A	12717.62	62.755
A41	DVDD	A	13367.62	66.8
A42	PWELL	A	13587.62	66.8
A43	DVSS	A	13807.62	66.8
A44	DVDD	A	14027.62	66.8
A45	AVSS	A	14247.62	66.8
A46	AVDD	A	14467.62	66.8
A47	CHIPID[3]	A	14687.62	66.8
A48	PVSS	A	14907.62	66.8
A49	PVDD	A	15127.62	66.8
A50	PVSS	A	15347.62	66.8
A51	PVDD	A	15567.62	66.8
A52	HSDATA_P	A	17325.355	61.82
A53	HSDATA_N	A	17545.355	61.82
A54	DVDD	A	18427.62	66.8
A55	CHIPID[2]	A	18647.62	66.8
A56	DVSS	A	18867.62	66.8
A57	SCE	A	19087.62	66.8
A58	AVSS	A	19307.62	66.8
A59	AVDD	A	19527.62	66.8
A60	DVSS	A	19747.62	66.8
A61	DVDD	A	19967.62	66.8
A62	CTRL	A	20187.62	66.8
A63	DVSS	A	20407.62	66.8
A64	DVDD	A	20627.62	66.8
A65	DVSS	A	20847.62	66.8
A66	DVDD	A	21067.62	66.8
A67	DATA[3]	A	21287.62	66.8
A68	DATA[7]	A	21507.62	66.8
A69	DVSS	A	21727.62	66.8
A70	DVDD	A	21947.62	66.8
A71	CHIPID[1]	A	22167.62	66.8
A72	DATA[2]	A	22387.62	66.8
A73	DATA[6]	A	22607.62	66.8
A74	DVSS	A	22827.62	66.8
A75	DVDD	A	23047.62	66.8
A76	DVSS	A	23267.62	66.8
A77	DATA[1]	A	23487.62	66.8
A78	DATA[5]	A	23707.62	66.8
A79	DVDD	A	23927.62	66.8
A80	AVSS	A	24147.62	66.8
A81	AVDD	A	24367.62	66.8
A82	DATA[0]	A	24587.62	66.8
A83	DATA[4]	A	24807.62	66.8
A84	SUB	A	25027.62	66.8
A85	DVSS	A	25247.62	66.8
A86	DVDD	A	25467.62	66.8
A87	BUSY	A	25687.62	66.8
A88	CHIPID[0]	A	25907.62	66.8
A89	AVSS	A	26127.62	66.8
A90	AVDD	A	26347.62	66.8
A91	PWELL	A	26567.62	66.8
A92	DACMONV	A	26797.62	66.8
A93	DACMONI	A	27897.62	66.8
A94	SUB	A	29627.62	66.8
B00	SUB	B	1057.62	525
B01	SUB	B	2157.62	525
B02	MCLK_P	B	4357.62	525

B03	MCLK_N	B	5457.62	525
B04	RST_N	B	6557.62	525
B05	POR_DIS_N	B	7657.62	525
B06	DCTRL_P	B	8757.62	525
B07	DCTRL_N	B	9857.62	525
B08	DCLK_P	B	12057.62	525
B09	DCLK_N	B	13157.62	525
B10	HSDATA_P	B	16897.62	525
B11	HSDATA_N	B	17997.62	525
B12	CTRL	B	20197.62	525
B13	DATA[3]	B	21297.62	525
B14	DATA[2]	B	22397.62	525
B15	DATA[1]	B	23497.62	525
B16	DATA[0]	B	24597.62	525
B17	BUSY	B	25697.62	525
B18	DACMONI	B	26797.62	525
B19	DACMONV	B	27897.62	525
B20	SUB	B	28997.62	525
C00	DVSS	B	2105.16	7890.88
C01	CHIPID[6]	B	3947.28	7890.88
C02	DVSS	B	5789.4	7890.88
C03	CHIPID[5]	B	7631.52	7890.88
C04	DVSS	B	9473.64	7890.88
C05	CHIPID[4]	B	11315.76	7890.88
C06	PVSS	B	13157.88	7890.88
C07	CHIPID[3]	B	15000	7890.88
C08	PVSS	B	16842.12	7890.88
C09	CHIPID[2]	B	18684.24	7890.88
C10	DVSS	B	20526.36	7890.88
C11	CHIPID[1]	B	22368.48	7890.88
C12	DVSS	B	24210.6	7890.88
C13	CHIPID[0]	B	26052.72	7890.88
C14	DVSS	B	27894.84	7890.88
D00	DVDD	B	2105.16	9906.88
D01	CHIPID[6]	B	3947.28	9906.88
D02	DVDD	B	5789.4	9906.88
D03	CHIPID[5]	B	7631.52	9906.88
D04	DVDD	B	9473.64	9906.88
D05	CHIPID[4]	B	11315.76	9906.88
D06	PVDD	B	13157.88	9906.88
D07	CHIPID[3]	B	15000	9906.88
D08	PVDD	B	16842.12	9906.88
D09	CHIPID[2]	B	18684.24	9906.88
D10	DVDD	B	20526.36	9906.88
D11	CHIPID[1]	B	22368.48	9906.88
D12	DVDD	B	24210.6	9906.88
D13	CHIPID[0]	B	26052.72	9906.88
D14	DVDD	B	27894.84	9906.88
E00	AVSS	B	2105.16	11465.92
E01	AVSS	B	5789.4	11465.92
E02	AVSS	B	9473.64	11465.92
E03	AVSS	B	13157.88	11465.92
E04	AVSS	B	16842.12	11465.92
E05	AVSS	B	20526.36	11465.92
E06	AVSS	B	24210.6	11465.92
E07	AVSS	B	27894.84	11465.92
F00	AVDD	B	2105.16	13723.84

F01	AVDD	B	5789.4	13723.84
F02	AVDD	B	9473.64	13723.84
F03	AVDD	B	13157.88	13723.84
F04	AVDD	B	16842.12	13723.84
F05	AVDD	B	20526.36	13723.84
F06	AVDD	B	24210.6	13723.84
F07	AVDD	B	27894.84	13723.84
G00	PWELL	B	3947.28	14395.84
G01	SUB	B	7631.52	14395.84
G02	PWELL	B	11315.76	14395.84
G03	SUB	B	15000	14395.84
G04	PWELL	B	18684.24	14395.84
G05	SUB	B	22368.48	14395.84
G06	PWELL	B	26052.72	14395.84

Table 2.8: Coordinates of the center points of the alignment markers.

Marker	Location	x [μm]	y [μm]
AM-TL	Top Left	175.32	14852.8
AM-TR	Top Right	29824.68	14852.8
AM-BL	Bottom Left	175.32	76.8
AM-BR	Bottom Right	29824.68	76.8

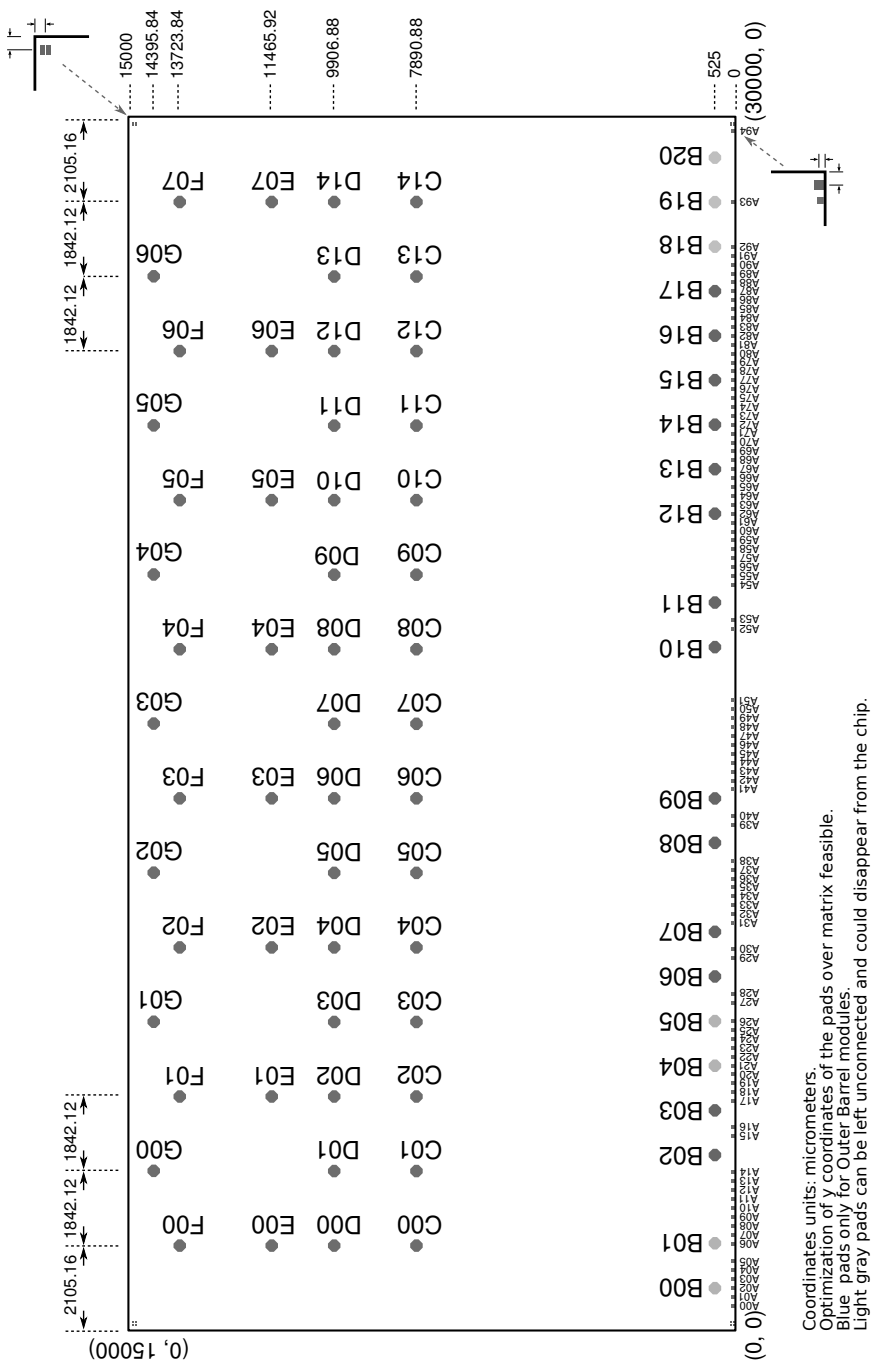

Figure 2.4: ALPIDE pad naming convention.

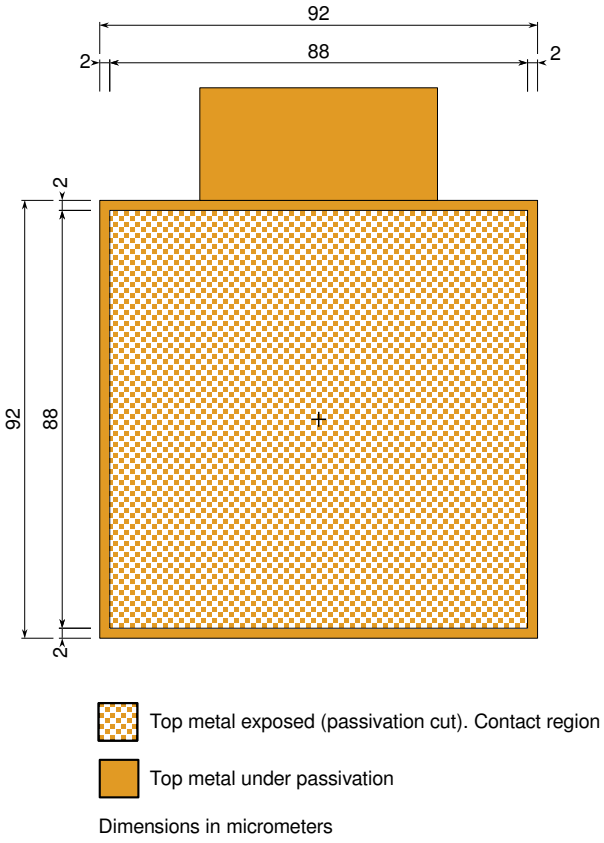

Figure 2.5: Geometry of type A pad.

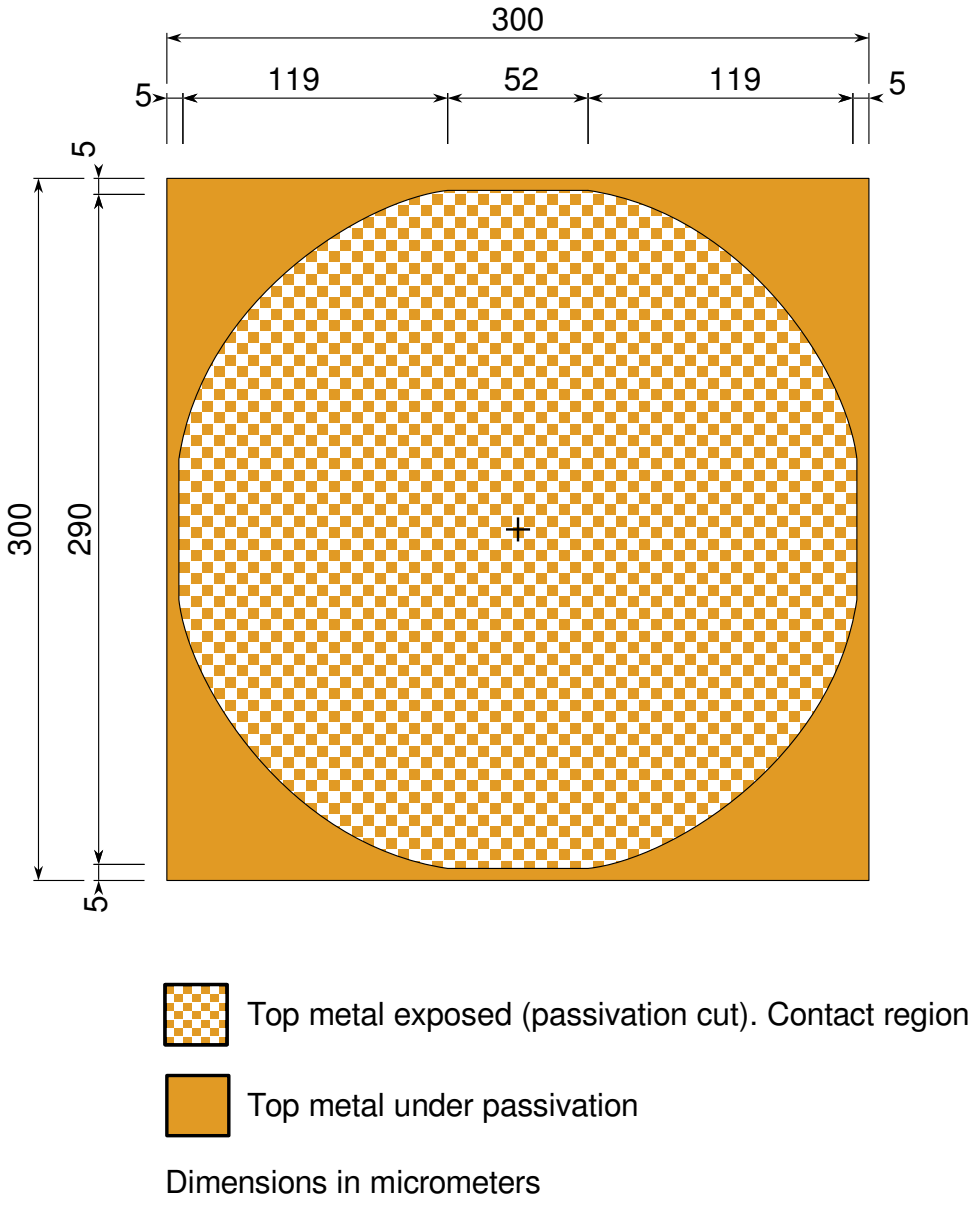

Figure 2.6: Geometry of type B pad.

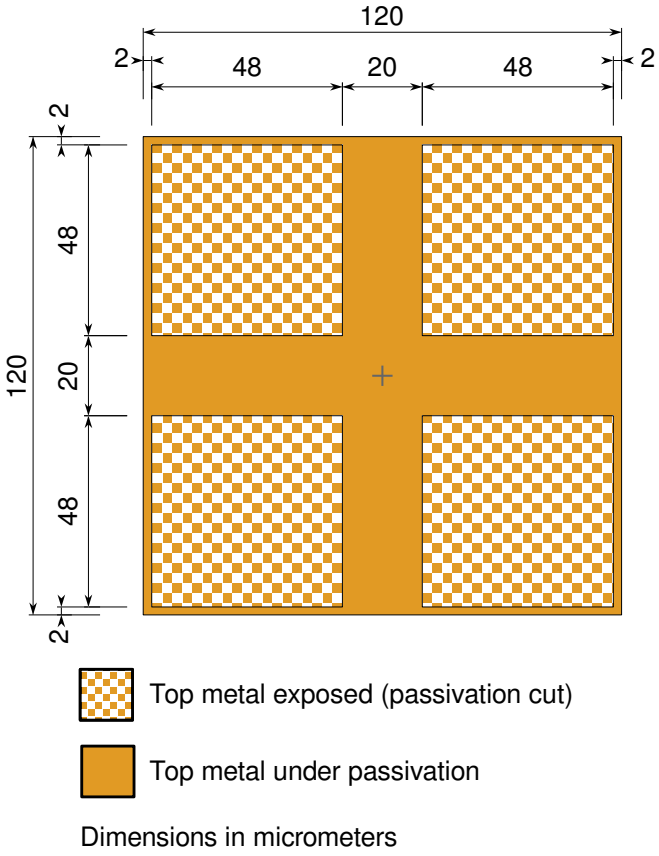

Figure 2.7: Geometry of the alignment markers.

3 User manual

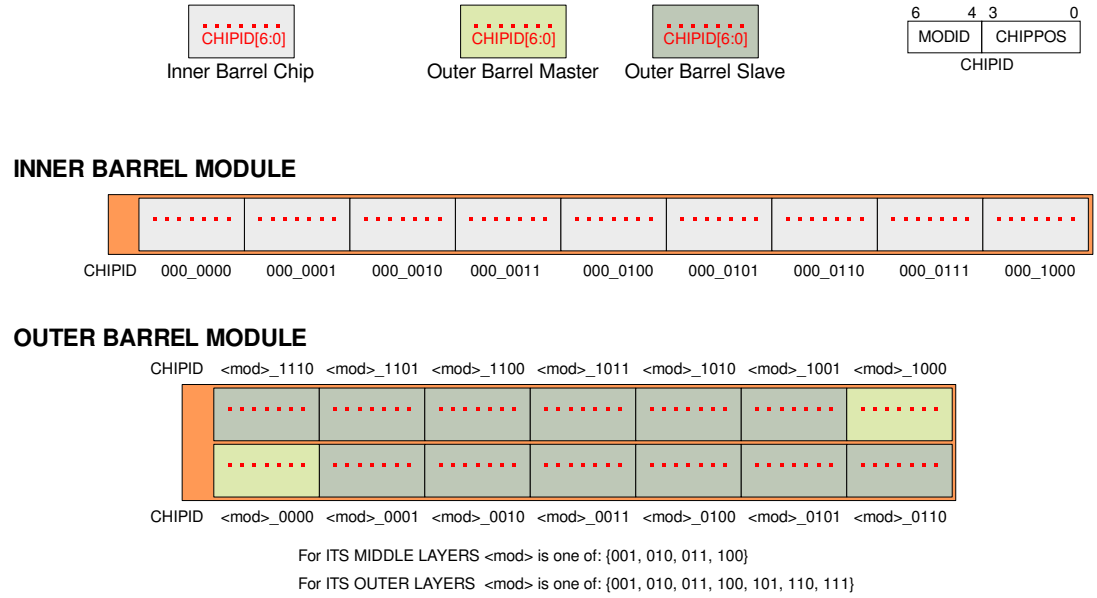

Figure 3.1: Illustration of chip identification and geographical address allocation. Default assignments of CHIPID values on one Inner Barrel Module and on a generic Outer Barrel Module.

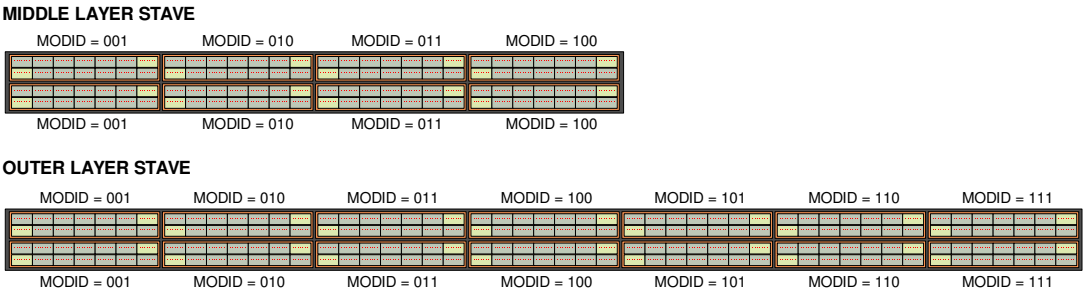

Figure 3.2: Illustration of chip identification and geographical address allocation. Default assignments of Module Identifier fields for the Middle Layer Stave and Outer Layer Stave.

3.1 Control interface and protocol

3.1.1 Chip identification and geographical address allocation

The chip can operate in three different roles in the ALICE ITS Upgrade application: Inner Barrel Chip, Outer Barrel Master and Outer Barrel Slave. The selection of the operating mode is based on the input applied to the **CHIPID[6:0]** chip port. This port selects the operating mode and also provides an address to the chip for the slow control transactions. The pads of this port shall be tied to digital supply (DVDD) to set the corresponding bit to 1. The pads have internal pull-down resistors. Leaving them unconnected is equivalent to shorting them to digital ground (DVSS) thus establishing a value of 0 for the corresponding bit.

The three bits **CHIPID[6:4]** constitute a **Module Identifier** field. The remaining bits **CHIPID[3:0]** act as identifiers of the position and role inside a module. The reference specification for the allocation of the values of CHIPID to the chips on the modules is illustrated in Fig. 3.1 and Fig. 3.2.

The *Module Identifier* field shall be **all zeros** for Inner Barrel chips. Chips with the three bits CHIPID[6:4] all set to zero identify and configure themselves as Inner Barrel Chips. The

remaining bits **CHIPID[3:0]** shall be a position dependent binary identifier and can have any of the values from 0 up to 14 (binary *b1110*). The binary code *b1111* **shall not be used** for CHIPID[3:0], since it is reserved for broadcast addressing.

The *Module Identifier* field shall contain at least one non-zero bit to configure the chip to operate in one of the Outer Barrel roles, i.e. CHIPID[6:4] *must not* be *b000*. CHIPID[6:4] is intended to be a module index, a unique value for all the fourteen chips of a specific module on a half-stave. CHIPID[6:4] can be one of *b001*, *b010*, *b011*, *b100* for modules on Middle Layers staves. CHIPID[6:4] can be one of *b001*, *b010*, *b011*, *b100*, *b101*, *b110*, *b111* for modules on Outer Layers staves. The four bits **CHIPID[3:0]** are also meant to specify the geographical position of the chip on the Outer Barrel Module and its role. There are two rows of seven chips on an Outer Barrel Module. Bit CHIPID[3] identifies in which of the two rows the chip is located. The remaining three bits **CHIPID[2:0]** select the operating mode of Outer Barrel Master if they are all set to zero, *b000*. Otherwise the chip behaves as an Outer Barrel Slave. Bits CHIPID[2:0] **must not be** binary *b111*, since this is reserved for broadcast addressing.

3.1.2 Control interfaces

The slow control interface serves two purposes:

1. provide write and read access to internal registers, commands, configuration and memories
2. distribute trigger commands or other broadcast synchronous signals

The ALPIDE chip has two ports to implement the slow control functionalities: a differential DCTRL port and a single-ended CTRL port. The port that is actually functional depends on the operating scenario. In Inner Barrel Chip role only the differential DCTRL port is used. In Outer Barrel Master role both ports are operated. In Outer Barrel Slave role only the single ended CTRL port is used. The slow control interface and the ports have been designed to implement a hierarchical control bus topology with multi-point connections of chips on the same electrical line. The reader can refer to appendices B and C for further details and illustrations on the items of this section.

The nine (9) chips on an Inner Barrel module are directly connected to a shared control differential line using the DCTRL port. The Inner Barrel control bus is entirely based on differential signaling and it has multipoint topology.

On Outer Barrel Staves, the control bus is implemented with a hierarchical structure. Every Module Master chip is connected with other Master chips on the same half-stave by a differential shared bus with multi-point topology. The differential line crosses the module boundaries and can connect 4 (Middle Layer Stave) or 7 (Outer Layer Stave) Module Master on the same row along the z axis. Each Outer Barrel Master chip acts as a slow control *hub* and relays the control transactions to six Outer Barrel Slave chips that are connected in a multi-point shared line topology with the Master. The bus segment local to the Outer Barrel Module operates with single-ended signaling through the CTRL port.

Inner Barrel modules and Outer Barrel modules present to the off-detector hardware fully equivalent control interfaces, physically appearing as a bi-directional differential port. The control interface supports *bi-directional, half-duplex* communication: data are exchanged in both directions but not simultaneously.

The signaling on the control buses is serial and synchronous with the system clock (nominal 40.08 MHz, LHC clock) that is distributed through a hierarchical clock tree. The slow control *transactions* are governed by the off-detector hardware initiating all type of messaging on the control bus. All chips have clocks derived from the same system clock and continuosly sample the incoming serial control stream, decoding the transactions on the bus. The deserialization and the decoding of the control messages are executed at corresponding clock edges in all chips.

The ALPIDE control interface has been designed with support of DC balanced signaling on the DCTRL port for applications that require or can benefit of AC coupling of the DCTRL line. This is obtained using Manchester encoding for the serial characters transmitted by the chips, following the IEEE8 802.3 convention for the bi-phase symbols. The transmission of control responses on th DCTRL line using Manchester coding is enabled by deafult (post reset value) but can be disabled in the chip configuration space. The off-detector electronics can also signal using Manchester encoding, this being transparent to on-chip circuits. The chips sample the control bus on the clock rising edges, therefore it is the electrical value seen on the bus at those sampling edges that is used by the logic of the chip control module.

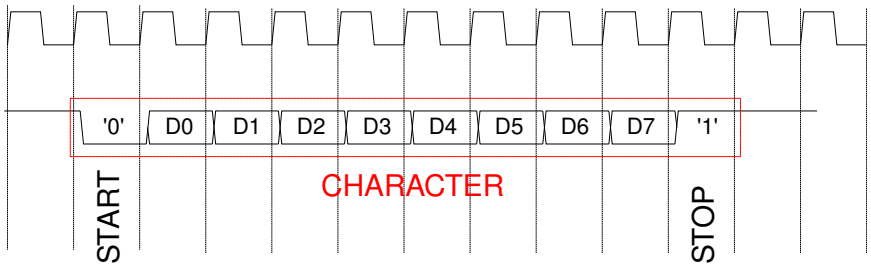

Figure 3.3: Format of a single character exchanged on the control bus.

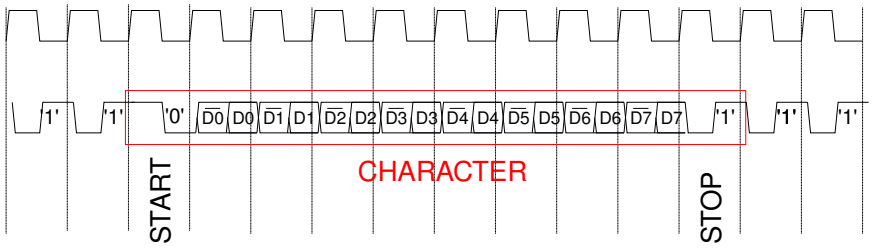

Figure 3.4: Format of a single character exchanged on the DCTRL bus with Manchester coding enabled (default).

3.1.3 Control transactions format

The transactions on the Control Bus are constituted of sequences of 10-bit wide *characters*. A character (Fig. 3.3) corresponds to the exchange of a single byte and it is made by a leading start bit (logic 0), 8-bit wide payload and a trailing stop bit (logic 1). The serial transmission convention is Less Significant Bit first. Fig. 3.4 illustrates the transmission of reply characters on the DCTRL line when Manchester signaling is enabled (default setting).

The idle state of the physical lines between the characters is logic 1. Bus idle gaps are allowed between characters transmitted to the chip. The minimum length of the idle gap between characters is 0 clock cycles (start bit immediately follows end bit of previous characters), the maximum length is 42 cycles.

The valid control transactions begin with pre-defined **OPCODE** characters, listed in Table 3.1. There is a Hamming distance of 4 bits between any two opcodes. This is meant to prevent the internal execution of transactions in case of bit errors on the transmission. There are broadcast type opcodes, to which all chips react executing an internal action. Four specific opcodes are reserved for the triggering commands. These are similar to broadcast opcodes but the internal decoding latency is reduced. The early decoding of the trigger commands is based on the two less significant bits of the reserved OPCODEs. This is the reason of having four redundant TRIGGER opcodes. Longer write and read transactions are prefix with dedicated opcodes.

There are five types of valid transactions, illustrated in Fig. 3.5:

- **BROADCAST COMMAND**: a single 10-bit character message (one of GRST, PRST, PULSE, BCRST, RORST, DEBUG); all chips react (e.g. Resets).
- **TRIGGER COMMAND**: a single character message cotaining a TRIGGER opcode, all chips react. The decoding of the trigger commands is internally executed at the deserializing stage to minimize trigger latency.
- **UNICAST WRITE**: six (6) characters message. A 16 bit data word is written to an internal register of one specific chip. It begins with a WRITE OPCODE character, followed by a CHIPID character identifying the target chip on the bus. The internal chip address is specified with the subsequent two characters (REG_ADDR) and finally the DATA payload is transferred with the last two characters.
- **MULTICAST WRITE**: six (6) characters message. A 16 bit data word is written siultaneously to an internal register of a set chips. It has the same format of a UNICAST WRITE with the expection that a MULTICAST ID chip identifier is used for the CHIPID character.
- **READ**: four (4) characters message to the chips followed by three (3) characters response from a chip. The first four characters are driven by the bus master to initiate the transaction (READ OPCODE), target the chip (CHIPID) and specify the chip internal address (REG_ADDR). Then a bus turnaround phase is initiated and the target chip (if present on the bus) transmits a response with its own CHIPID (acknowledge) followed by two DATA characters payload. The response characters are transmitted by the chip one after the other without any idle gap between the stop and start bits of consecutive characters.

The BROADCAST COMMAND transactions are shortcuts for MULTICAST WRITE transactions targeting the chip command register (see also section 3.2.2). The effects of the broadcast commands can be achieved individually on a specific chip using the command register feature.

The MULTICAST WRITE transactions are based on reserved values for the chip identification characters. These are identified as **MULTICAST ID** control addresses:

- **GLOBAL BROADCAST** is the binary string **b00001111**. Any chip listening to a write transaction with this byte on the chip identification character reacts and internally executes the write transaction, regardless of its own CHIPID and role.
- **Outer Barrel MULTICASTs**: binary code {*b0*, *modid[2:0]*, *master_id*, *b111*}. This enables addressing 7 chips on a specific outer barrel module identified by the 3 bits *modid* and connected to the OB master with CHIPID[3] equal to bit *master_id*.

Opcode	Hex value	Purpose
TRIGGER	8'hB1	Trigger command
TRIGGER	8'h55	Trigger command
TRIGGER	8'hC9	Trigger command
TRIGGER	8'h2D	Trigger command
GRST	8'hD2	Chip global reset
PRST	8'hE4	Pixel matrix reset
PULSE	8'h78	Pixel matrix pulse
BCRST	8'h36	Bunch Counter reset
DEBUG	8'hAA	Sample state in shadow registers
RORST	8'h63	Readout (RRU/TRU/DMU) reset
WROP	8'h9C	Start Unicast or Multicast Write
RDOP	8'h4E	Start Unicast Read

Table 3.1: Valid opcodes of control transactions

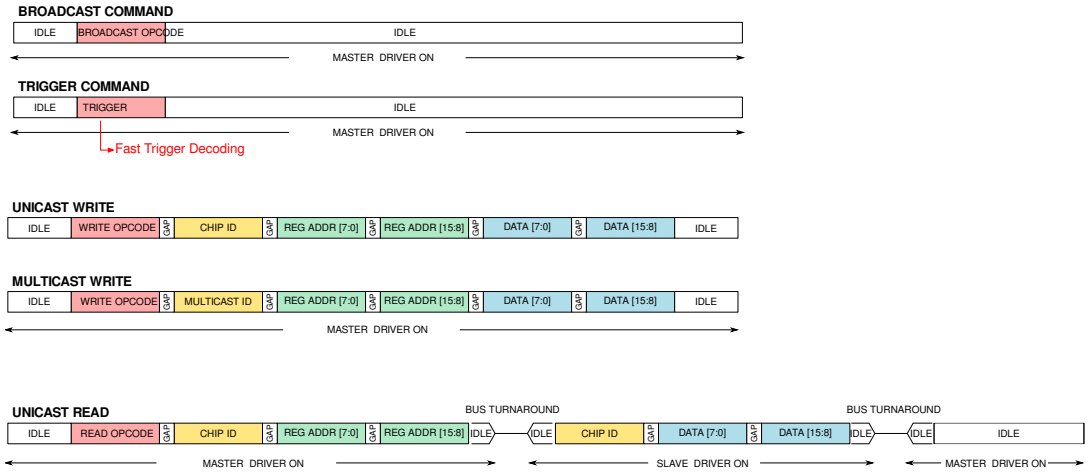

Figure 3.5: Format of valid transactions on the control bus.

3.1.4 Bus turnaround and reply phase of read control transactions

The response of the chips in read transactions is strictly specified and complies with the timing specification illustrated in Fig. 3.6.

The bus master must release the electrical drive of the bus line for a predefined number of clock cycles (50). This interval is used by the addressed chip (if any) to reply with its own CHIPID followed by two consecutive DATA characters. The same specification of bus turnaround timing applies to: (a) the differential bus connecting external electronics and Inner Chips, (b) the differential bus connecting external electronics and OB Masters and (c) the single ended shared bus connecting one Outer Barrel Master chip with its Slave chips.

The bus turnaround and the reply phase can be divided in distinct sub-phases separated by notable clock edges.

1. Clock edge 0, reference clock edge. The bus master completes the transmission of stop (mark) bit of ADDRH character. Beginning of *Master Idle Phase* with duration of 5 (five) clock periods. The bus master keeps driving idle the line during this phase. Purpose: allow bus slaves to complete sampling of the delayed message. Bus slaves stop sampling the line during this phase ignoring the state of the line onwards.

2. 5^{th} clock edge, the bus master disables the line driver. Beginning of the *First Turnaround Phase*: duration 5 cycles. Purpose: allow margin to prevent line contention and a introduce an interval in which the line electrical state is ignored by both bus masters and bus slaves. The target bus slave shall start actively driving the line with Idle during this phase.

3. 10^{th} clock edge, the bus master starts sampling the line at this edge (included). The target shall already be driving actively (with Idle) the line at this time. It is the beginning of the *Slave Idle Phase*: duration 5 cycles. Purpose: allow the line receiver to sense the Idle and signaling phase (if Manchester coding is used by the Target).

4. 15^{th} clock edge, the responding slave shall keep driving idle up to this edge. This edge can be the launching edge of the first start bit. Beginning of the *Reply Phase*: duration 35 cycles. The responding slave transmits the three reply characters (min 30 cycles needed) foreseen for the Read Transaction.

5. 50^{th} clock edge, the bus master samples the line for the last edge. End of Reply Phase, beginning of second *Slave Idle Phase*: duration 5 cycles. The responding slave shall

drive the line Idle during the Slave Idle Phase. The bus master stops sampling the line during the Slave Idle Phase.

6. 55^{th} clock edge, beginning of the *Second Turnaround Phase*, duration 5 cycles. The responding slave stops driving actively the line after this edge. The bus master starts driving the line actively (with Idle) from during the Second Turnaround phase.

7. 60^{th} clock edge, the bus slaves shall start sampling the line at this edge (included). The line shall be driven Idle by the bus master. Marks the beginning of the *Second Master Idle phase*: duration 5 cycles. The bus master keeps driving Idle during the Second Master Idle phase.

8. 65^{th} clock edge, completion of Read Transaction

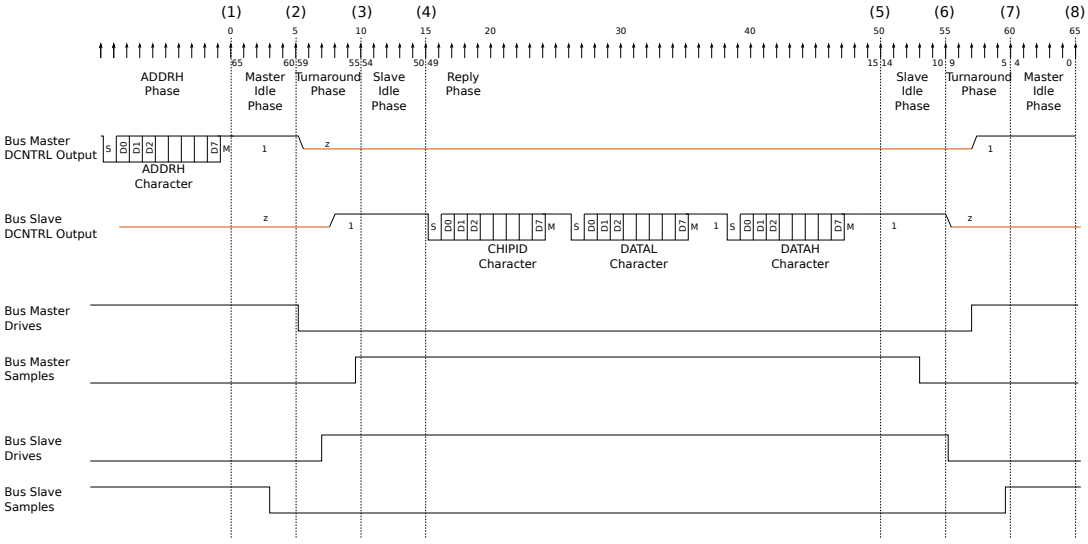

Figure 3.6: Timing diagram of the reply phase of a Read transaction including turnaroud phases. Signaling is represented as seen on the differential line at the output of the bus master (off-detector electronics).

3.2 Control registers and addressing space

The ALPIDE control interface provides access to the chip internal registers and data path memories. The write and read transactions via the control interface have 16 bit wide address field and 16 bit wide data payload.

3.2.1 Addressing space

The registers and physical memory locations are mapped to the addressing space according to the scheme described in the following tables. A read access to an unused address returns hexadecimal 0xFFFF, while a write access has no effects.

Several addresses point to registers or physical memory locations that are less than 16 bit wide. In these cases the physical bits are accessible as *least significant* bits of the transaction data field. The unused most significant bits are read as constant binary 0s.

3.2.2 Periphery Control Registers

Address	Mode	Reset	Register or memory
0x0000	R/W	0x0000	Command Register
0x0001	R/W	0x01FC	Mode Control register
0x0002	R/W	0x0000	Disable of regions 0-15
0x0003	R/W	0x0000	Disable of regions 16-31
0x0004	R/W	0x0010	FROMU Configuration Register 1
0x0005	R/W	0x0014	FROMU Configuration Register 2
0x0006	R/W	0x0000	FROMU Configuration Register 3
0x0007	R/W	0x0000	FROMU Pulsing Register1
0x0008	R/W	0x0000	FROMU Pulsing Register 2
0x0009	R	-	FROMU Status Register 1
0x000A	R	-	FROMU Status Register 2
0x000B	R	-	FROMU Status Register 3
0x000C	R	-	FROMU Status Register 4
0x000D	R	-	FROMU Status Register 5
0x000E	R/W	0x0AAA	DAC settings for DCLK and MCLK I/O buffers
0x000F	R/W	0x00AA	DAC settings for CMU I/O buffers
0x0010	R/W	0x004F	CMU and DMU Configuration Register
0x0011	R	-	CMU and DMU Status Register
0x0012	R	-	DMU Data FIFO [15:0]
0x0013	R	-	DMU Data FIFO [23:16]
0x0014	R/W	0x008D	DTU Configuration Register
0x0015	R/W	0x0088	DTU DACs Register
0x0016	R	-	DTU PLL Lock Register 1
0x0017	R/W	0x0000	DTU PLL Lock Register 2
0x0018	R/W	0x0000	DTU Test Register 1
0x0019	R/W	0x0000	DTU Test Register 2
0x001A	R/W	0x0000	DTU Test Register 3
0x001B	R/W	0x0008	BUSY min width

Table 3.2: Periphery Control Registers.

0x0000 - Command register

The command register is a special register allowing the execution of internal operations or sequences. Writing of specific codes to this register generates internal pulses that are typically distributed to other chip blocks. The valid command codes are listed in table 3.3.

The byte code of *BROADCAST COMMAND* control transactions or the byte code of *TRIGGER COMMAND* transactions are always written to the command register (padding the most significant byte with zeros). Therefore, the *BROADCAST COMMAND* control transactions are effectively shortcuts for a broadcast write transaction to the command register of all receiving chips. Notice however that the commands *CMU CLEAR ERR*, *FIFOTEST*, *LOADOBDEFCFG*, *XOFF*, *XON*, *ADCMEASURE* are accessible only by normal write transactions and there are no shortcuts for these. This reflects on the command code having the most significant byte differing from binary zero.

The *TRIGGER COMMANDs* are an exception to the mechanism described before. These are decoded directly in the CMU during de-serialization, not in the command register. This is done to minimize the latency of decoding trigger commands. However the *TRIGGER* opcode is still written to the command register on a valid *TRIGGER COMMAND* transaction and a read back from the command register after the transmission of a trigger command will return the trigger opcode value that was utilised.

OPCODE	Hex value	Purpose
GRST	0x00D2	Chip global reset
PRST	0x00E4	Pixel matrix **RESET**
PULSE	0x0078	Pixel matrix **PULSE**
BCRST	0x0036	Bunch Counter reset
RORST	0x0063	Readout (RRU/TRU/DMU) reset
DEBUG	0x00AA	Store snapshot into debug registers
TRIGGER	0x00B1,0055,00C9,002D	Trigger command
WROP	0x009C	Start Unicast or Multicast Write
RDOP	0x004E	Start Unicast Read
CMU CLEAR ERR	0xFF00	Clear CMU error flags
FIFOTEST	0xFF01	Starts regions memory test
LOADOBDEFCFG	0xFF02	Loads default configuration for the OB Module. Initial Token is set in the MASTER chip; Previous Chip ID is set to 0x6 in the MASTER, and to Chip ID minus 1 in the SLAVEs
XOFF	0xFF10	Stops sending data off-chip
XON	0xFF11	Resume data sending
ADCMEASURE	0xFF20	Start ADC measure

Table 3.3: Command codes recognized by the command register

0x0001 - Mode Control Register

Bits	Field Name	Read/Write	Initial Value
1:0	Chip Mode selector	R/W	0x0
2	Clustering enable	R/W	0x1
3	Matrix Readout Speed	R/W	0x1
5:4	IB Serial Link Speed	R/W	0x3
6	Enable Skewing of Global Signals	R/W	0x1
7	Enable Skewing of Start Readout	R/W	0x1
8	Enable Readout Clock Gating	R/W	0x1
9	Enable Readout From CMU	R/W	0x0
15:10	Not used	-	-

Table 3.4: Mode Control Register Field Description

- **Bits 1:0 - Chip Mode Selector**
  0: Configuration Mode (readout disabled)
  1: Readout enabled, Triggered Mode
  2: Readout enabled, Continuous Mode
- **Bit 2 - Clustering Enable**
  0: Disable clustering of nearby hits
  1: Enable clustering of nearby hits
- **Bit 3 - Matrix Readout Speed**
  0: Readout of one pixel hit in each region every four clock cycles (10 MHz)
  1: Readout of one pixel hit in each region every two clock cycles (20 MHz)
- **Bits 5:4 - IB Serial Link Speed**
  Sets the serial link line rate in the Inner Barrel mode of operation:
  0: 400 Mb/s
  1: 600 Mb/s
  2,3: 1200 Mb/s

- **Bit 6 - Enable Skewing of Global Signals**
  0: Disable the skewing of global signals to the matrix
  1: Enable the skewing of global signals to the matrix

- **Bit 7 - Enable Skewing of Start of Readout**
  Controls the skewing of the start of the readout processes in the regions.
  0: Disabled
  1: Enabled.
  When the skewing of the start of readout is disabled, the readout processes in the 32 regions are initiated at the same clock cycle. When it is enabled, the 32 readout processes are started gradually. On the first cycle the 8 central regions from 12 to 19 are enabled. One cycle later other 8 regions are enabled (4 on the left and 4 on the right) and so on. On the fourth clock cycle all regions are active reading the matrix.

- **Bit 8 - Enable Readout Clock Gating**
  Controls the clock gating for the readout modules (RRU and TRU).
  0: Clock gating disabled. The clock of RRU and TRU modules is always active.
  1: Clock gating enabled. The clock of most of RRU and TRU circuits is disabled when they are idle.

- **Bit 9 - Enable Readout From CMU**
  When this bit is set, frame data are not transmitted off-chip, but remain in the internal memory until they are fetched by the user with control read transactions from registers 0x0012 and 0x0013.

0x0002 - Region Disable Register 1

16 bit register used to disable the readout of individual regions. When a bit is set to 1, the corresponding region is disabled. The LSB corresponds to Region 0 and MSB to Region 15.

0x0003 - Region Disable Register 2

16 bit register used to disable the readout of individual regions. When a bit is set to 1, the corresponding region is disabled. The LSB corresponds to Region 16 and MSB to Region 31.

0x0004 - FROMU Configuration Register 1

Bits	Field Name	Read/Write	Initial Value
2:0	Pixel MEB Mask	R/W	0x0
3	Internal **STROBE** generation	R/W	0x0
4	Enable **BUSY** Monitoring	R/W	0x1
5	Test Pulse Mode	R/W	0x0
6	Enable Test **STROBE**	R/W	0x0
7	Enable Rotate Pulse Lines	R/W	0x0
10:8	Trigger Delay	R/W	0x0
15:11	Not used	-	-

Table 3.5: FROMU Configuration Register 1 Register Field Description

- **Bits 2:0 - Pixel MEB Mask**
  Setting one of these three bits to 1 disables the usage of a corresponding bank of the in-pixel multi event buffers (the default value b000 enables all three MEB slices). Setting

bit 0 disables MEB Slice 1, bit 1 disables Slice 2 and bit 2 disables Slice 3. **Masking all slices with b111 is ignored and equivalent to enabling all.**

- **Bit 3 - Internal Strobe Generation**
  Controls the sequencer to generate a periodically repeating internal trigger command (in readout mode). 0: disabled, 1: enabled.
- **Bit 4 - Enable Busy Monitoring**
  Controls if the FROMU monitors the BUSY input and rejects triggers when the BUSY is asserted (input low). 0: disabled (BUSY line ignored), 1: enabled (BUSY line monitored, default).
- **Bit 5 - Test Pulse Mode**
  Controls the type of pulsing of the pixels that is triggered by the PUlSE command. 0: digital pulsing (override of the discriminator output, default), 1: analog pulsing (injection of test charge in the front-end)
- **Bit 6 - Enable Test STROBE**
  1: enables the automatic generation of an internal TRIGGER pulse after a **PULSE** command
- **Bit 7 - Enable Rotate Pulse Lines**
  1: enable automatic shift and rotate of selected pulse line after each **PULSE** signal
- **Bits 10:9 - Trigger Delay**
  Delay applied to the trigger command (measured in periods of clock), increasing the latency between the command and the internal generation of **STROBE** signal.

0x0005 - FROMU Configuration Register 2

Controls the *duration* of the **STROBE** pulses to the pixels, in units of clock cycles. Value n gives a duration of $n+1$ cycles (minimum 1 cycle and maximum 65536 cycles i.e. from 25 ns to 1638.4 us assuming a clock period of 25 ns).

0x0006 - FROMU Configuration Register 3

Controls the duration of the *gap* between subsequent **STROBE** pulses to the pixels when the **internal sequencer is activated**. Value n gives a gap of $n+1$ clock cycles (minimum 1 cycle and maximum 65536 cycles i.e. from 25 ns to 1638.4 us assuming a clock period of 25 ns).

0x0007 - FROMU Pulsing Register 1

Controls the delay from the **PULSE** signal to the **STROBE** signal when the automatic generation of STROBE following PULSE is enabled. Value n gives a delay of $n+1$ cycles (minimum 1 cycle and maximum 65536 cycles i.e. from 25 ns to 1638.4 us assuming a clock period of 25 ns).

0x0008 - FROMU Pulsing Register 2

Controls the duration of the **PULSE** signal. Value n gives a duration of n cycles (maximum 65535 cycles i.e. from 25 ns to 1638.375 us assuming a clock period of 25 ns). If this register is set to 0, the APULSE or DPULSE signals are *not* activated.

0x0009 - FROMU Status Register 1

Current value of the FROMU trigger counter. This counter is incremented on any trigger, external (TRIGGER command) and internal (from the internal sequencer).

0x000A - FROMU Status Register 2

Current value of the **STROBE** counter. This counter is incremented every time a pulse on any of the global STROBE lines is sent to the pixels.

0x000B - FROMU Status Register 3

Current value of the Matrix Readout counter. This counter is incremented every time the completion of the readout of a matrix frame is acknowledged to the FROMU by the Region Readout Units.

0x000C - FROMU Status Register 4

Current value of the Frame counter. This counter is incremented every time the FROMU sends a Chip Header message to the Top Readout Unit (TRU).

0x00D - FROMU Status Register 5

Bits	Field Name	Read/Write	Initial Value
11:0	Bunch Counter	R	-
14:12	Events In MEB	R	-
15	Frame Extended	R	-

Table 3.6: FROMU Status Register 5 Field Description

- **Bits 11:0 - Bunch Counter**
  This field contains the last sample of the Bunch Crossing counter. The FROMU samples the Bunch Crossing counter shortly after it receives a trigger pulse. The sample of the Bunch Crossing counter is also copied into the CHIP HEADER data word prefix to the chip frame data packet corresponding to the trigger pulse (without the two less significant bits).
- **Bits 14:12 - Events In MEB**
  Each of the 3 bits is set if the corresponding bank of the in-pixel Multi-Event Buffers has a frame stored the readout of which is not yet completed.
- **Bits 15 - Frame Extended**
  This bit is set if a new trigger arrives (or a new **STROBE** is generated by the internal sequencer) before the end of the current **STROBE** window. The current strobing window is extended accordingly to the value set in FROMU Configuration Register 2.

DAC settings for MLVDS I/O buffers

The **DCLK**, **MCLK** and **DCTRL** differential ports are implemented with two instances of the same pseudo-MLVDS transceiver block. The transceiver block has tunable currents for the receiver and for the driver circuits. The following two registers control the currents of the MLVDS circuits:

0x000E - DAC settings for DCLK and MCLK I/O buffers

Bits	Field Name	Read/Write	Initial Value
3:0	DCLK Receiver current	R/W	0xA
7:4	DCLK Driver current	R/W	0xA
11:8	MCLK Receiver current	R/W	0xA
15:12	Not used	-	-

Table 3.7: CLK buffer DAC Settings Register Field Description

0x000F - DAC settings for CMU I/O buffers

Bits	Field Name	Read/Write	Initial Value
3:0	DCTRL receiver current	R/W	0xA
7:4	DCTRL driver current	R/W	0xA
15:8	Not used	-	-

Table 3.8: CMU Buffer DAC Settings Register Field Description

0x0010 - CMU and DMU configuration register

This register is used to configure the Control Management Unit (CMU) and Data Management Unit (DMU) modules.

Bits	Field Name	Read/Write	Initial Value
3:0	Previous Chip ID	R/W	0xF
4	Initial Token	R/W	0x0
5	Disable Manchester encoding	R/W	0x0
6	Enable Double Data Rate	R/W	0x1
15:7	Not used	-	-

Table 3.9: CMU and CMU Config Register Field Description

- **Bits 3:0 - Previous Chip ID**
  Value of the chip Index of the chip accessing the OB Module local bus just before the chip on which the register resides

- **Bit 4 - Initial Token**
  Must be set to 1 to the chip that has the token by default (normally should be the MASTER chip of the module)

- **Bit 5 - Disable Manchester Encoding**
  0: CMU Manchester Encoding enabled (default), 1: Encoding disabled

- **Bit 6 - Enable Double Data Rate**
  0: Disable DDR on Local Bus, 1: Enable DDR on Local Bus (default)

0x0011 - CMU Errors Counter Register

Bits	Field Name	Read/Write	Initial Value
3:0	Deserializer error counter	R	-
7:4	Time-out error counter	R	-
11:8	Unknown OPCODE error counter	R	-

Table 3.10: CMU Errors Counter Register Field Description

The CMU has three 4 bit wide counters of control protocol error conditions. This is a read only register to retrieve the values of the counters and to detect control protocol violation errors. Under normal conditions these counters should remain at zero. These counters can be reset by a dedicated command written to the command register.

The deserializer errors are of two types. A *framing error* is induced by the incoming serial stream if the stop bit is not received at the expected cycle after the start bit. An *overrun error* is induced by a missing acknowledge of the retrieval of a deserialized character from the output port.

Readout from CMU

When bit 9 of Mode Control Register is set, data is not sent automatically off-chip but remains in the memories until it is read through the DMU Data FIFO registers. DMU Data FIFO LSB register should be read first. Reading of MSB register will also pop the word from the internal DMU FIFO. Note that if BUSY words are present in the DMU Busy FIFO, they have priority and are read out before any data word.

0x0012 - DMU Data FIFO LSB

Readout of bits 15:0 from DMU Data or Busy FIFO

0x0013 - DMU Data FIFO MSB

Readout of bits 23:16 from DMU Data or Busy FIFO

0x0014 - DTU Configuration Register

Bits	Field Name	Read/Write	Initial Value
1:0	PLL VCO Delay Stages control	R/W	0x1
2	PLL Bandwidth control	R/W	0x1
3	PLL off signal	R/W	0x1
7:4	Serializer Phase	R/W	0x8
8	PLL Reset	R/W	0x0
11:9	Not used	-	-
12	Load Enable Status	R	-
15:13	Not used	-	-

Table 3.11: DTU Configuration Register Field Description

- **Bits 1:0 - PLL VCO Delay Stages control**
  these two bits control the number of delay stages used in the VCO. *b00*: VCO with 3

stages (slow case). *b01*: VCO with 4 stages (typical case). *b11*: VCO with 5 stages (fast case).

- **Bit 2 - PLL bandwidth control**
  b0 selects a wide bandwidth response, *b1* selects a narrow bandwidth response.
- **Bit 3 - PLL off signal**
  b1 shuts down the PLL.
- **Bits 7:4 - Serializer Phase**
  control of the timing of the parallel load of data into the shift registers of the serializer.
- **Bit 8 - PLL Reset**
  Asynchronous reset of the PLL.
- **Bit 12 - Load Enable Status**
  read only monitor of the enable input of the DTU Serializer generated by the DTU logic.

0x0015 - DTU DACs Register

Bits	Field Name	Read/Write	Initial Value
3:0	PLL Charge Pump current setting	R/W	0x8
7:4	High Speed Line Driver current setting	R/W	0x8
11:8	Pre-emphasis driver current setting	R/W	0x0
15:12	Not used	-	-

Table 3.12: DTU DACs Register Field Description

0x0016 - DTU PLL Lock Register 1

This read only register gives access to output signals and flags of PLL related logic.

Bits	Field Name	Read/Write	Initial Value
7:0	Lock Counter	R	-
8	Lock Flag	R	-
9	Lock Status	R	-
15:10	Not used	-	-

Table 3.13: DTU PLL Lock Register 1 Field Description

- **Lock Counter**
  Internal counter incremented every time the PLL monitoring state machine enters the *PLL LOCKED* state.
- **Lock Flag**
  Direct access to the unfiltered PLL Lock flag.
- **Lock Status**
  Asserted while the PLL monitoring state machine is in the *PLL LOCKED* state. This signal is the result of a moving average of the PLL Lock Flag executed by the PLL monitoring state machine.

0x0017 - DTU PLL Lock Register 2

This register provides control of the behavior of the PLL monitoring state machine.

Bits	Field Name	Read/Write	Initial Value
7:0	Lock Wait Cycles	R/W	0x0
15:8	Unlock Wait Cycles	R/W	0x0

Table 3.14: DTU PLL Lock Register 2 Fields Description

- **Lock Wait Cycles**
  The PLL monitoring state machine enters the *PLL LOCKED*
  state after sampling a continuous assertion of the PLL Lock flag for a number of clock cycles set with this register.
- **Unlock Wait Cycles**
  The PLL monitoring state machine enters the *PLL UNLOCKED*
  state after sampling the PLL Lock flag continuously de-asserted for a number of clock cycles set with this register.

0x0018 - DTU Test Register 1

This register provides access to the built-in DTU test facilities.

Bits	Field Name	Read/Write	Initial Value
0	Test Enable	R/W	0x0
1	Internal Pattern Enable	R/W	0x0
2	Test Single Mode	R/W	0x0
4:3	PRBS Rate	R/W	0x0
5	Bypass 8b10b	R/W	0x0
7:6	BDIN8b10b0	R/W	0x0
9:8	BDIN8b10b1	R/W	0x0
11:10	BDIN8b10b2	R/W	0x0
12	K0	R/W	0x0
13	K1	R/W	0x0
14	K2	R/W	0x0
15	Not Used	-	-

Table 3.15: DTU Test Register 1 Field Description

- **Bit 0 - Test Enable**
  Activates the DTU test mode. When this bit is set, the parallel data bus to the encoder and then to the DTU is driven by the test logic and effectively disconnected from the chip Data Management Unit.
- **Bit 1 - Internal Pattern Enable**
  Selects between constant test patterns (1'b0) or test patterns generated by an internal pseudo-random pattern generator (1'b1). The internal pattern generator produces PRBS-7 bit sequences.
- **Bit 2 - Test Single Mode**
  Selects if the encoding and line rate control circuits of the DTU Logic operate in single lane mode, producing a 400 Mb/s output stream. This bit is effective only while the DTU test mode is enabled (bit 0 set). The single lane mode is internally enabled when the chip is configured as Outer Barrel Master.
- **Bits 4:3 - PRBS-7 pattern rate**
  Control of the output bit rate when using the internal PRBS-7 pattern generator. *2'b00*: 1200 Mb/s. *2'b01*: 400 Mb/s. *2'b10*: 600 Mb/s. *2'b11*: synchronous reset of the PRBS-7 pattern generator. These bits are effective in test mode and with the internal pseudo-random pattern generator enabled.

- **Bit 5 - Bypass 8b10b**
  Forces bypassing the 8b10b encoding stage. The parallel data (either from the DMU either from the test logic) are applied directly to the DTU Serializer parallel inputs when this signal is asserted. Note: this is effective independently from the enabling of the test mode.
- **Bits 7:6 - BDIN8b10b0**
  Padding bits for the DIN0 Test Code, used in configurations when the 8b10b encoder is bypassed and the internal pattern generator is not used.
- **Bits 9:8 - BDIN8b10b1**
  Padding bits for the DIN1 Test Code, used in configurations when the 8b10b encoder is bypassed and the internal pattern generator is not used.
- **Bits 11:10 - BDIN8b10b2**
  Padding bits for the DIN2 Test Code, used in configurations when the 8b10b encoder is bypassed and the internal pattern generator is not used.
- **Bit 12 - K0**
  Direct drive of the K-code selection bit of lane 0 of the 8b10b encoder. Useful in test configurations with the encoder enabled and test patterns applied to the encoder inputs.
- **Bit 13 - K1**
  Direct drive of the K-code selection bit of lane 1 of the 8b10b encoder. Useful in test configurations with the encoder enabled and test patterns applied to the encoder inputs.
- **Bit 14 - K2**
  Direct drive of the K-code selection bit of lane 2 of the 8b10b encoder. Useful in test configurations with the encoder enabled and test patterns applied to the encoder inputs.

0x0019 - DTU Test Register 2

Bits	Field Name	Read/Write	Initial Value
7:0	DIN0 Test Code	R/W	0x0
15:8	DIN1 Test Code	R/W	0x0

Table 3.16: DTU Test Register 2 Field Description

- **Bits 7:0 - DIN0**
  Constant programmable test code applied to the lane 0 (bits 7:0) of the encoder when Test Mode is selected and the internal pattern generator is disabled.
- **Bits 15:8 - DIN1**
  Constant programmable test code applied to the lane 1 (bits 15:8) of the encoder when Test Mode is selected and the internal pattern generator is disabled.

0x001A - DTU Test Register 3

Bits	Field Name	Read/Write	Initial Value
7:0	DIN2 Test Code	R/W	0x0
8	Force Load Enable High	R/W	0x0
9	Force Load Enable Low	R/W	0x0
15:10	Not used	-	-

Table 3.17: DTU Test Register 3 Field Description

- **Bits 7:0 - DIN2**
  Constant programmable test code applied to the lane 2 (bits 23:16) of the encoder when Test Mode is selected and the internal pattern generator is disabled.

- **Bit 8 - Force Load Enable High**
  Forces to logic high the Load Enable input of the Serializer. This overrides the normal behavior consisting of having the Load Enable equal to the Locked Status of the PLL monitoring state machine. This control is always effective, independently of the enabling of the DTU test logic (bit 0).

- **Bit 9 - Force Load Enable Low**
  Forces to logic low (resetting level) the Load Enable input of the Serializer. This overrides the normal behavior consisting of having the Load Enable equal to the Locked Status of the PLL monitoring state machine. This control is always effective, independently of the enabling of the DTU test logic (bit 0) and has priority over the Force Load Enable High control bit. When the DTU Serializer receives a low level on the Load Enable input it remains in a reset condition and its two shift registers keep constant values. When this bit is set the high speed output consists of a replica of the PLL output clock propagated through the Serializer and the DTU Driver input multiplexer.

0x001B - BUSY Minimum Width

Bits	Field Name	Read/Write	Initial Value
4:0	**BUSY** Minimum Length	R/W	0x8
15:5	Not used	-	-

Table 3.18: BUSY Minimum Width Field Description

- **Bits 4:0 - BUSY** Minimum Length
  Set the minimum length of **BUSY** signal, in step of 25ns

3.2.3 Region Control Registers and direct access to the DPRAM memories

The thirty-two Region Readout Unit modules contain registers to exclude from the readout specific double-columns (Priority Encoders). They also feature a Region Status Register. Each RRU contains a 124 words deep, 24b wide DPRAM hard macro used to implement readout FIFOs. The control logic supports direct read/write access to the storage elements of the DPRAM macros when the chip is in *Configuration Mode*.

The read and write accesses to the Region Control Registers and the DPRAM locations are implemented through the following address decoding scheme. The address field of the control transactions is internally treated as composed of sub-fields:

- ADDRESS[15:11] = Region selection field
- ADDRESS[10:8] = Register or memory selection field
- ADDRESS[7] = Broadcast write bit
- ADDRESS[6:0] = Register selector or DPRAM word offset

Values of the sub-fields of the transaction address are detailed in table 3.19.

Setting the *Broadcast write bit* in a write transaction has the effect of writing to the selected locations in all regions, ignoring the *Region selection field*.

Addr[15 : 11]	**Addr**[10 : 8]	**Addr**[6 : 0]	**Mode**	**Reset**	**Register or memory**
0x0-0x1F	0x1	0x0-0x7F	R/W	-	RRU MEB LSB
0x0-0x1F	0x2	0x0-0x7F	R/W	-	RRU MEB MSB
0x0-0x1F	0x3	0x0	R/W	0x0000	Double Column Disable Register
0x0-0x1F	0x3	0x1	R/W	-	Region Status Register

Table 3.19: Addressing of Region Control Registers and DPRAM memory words.

RRU Memory random access

The locations of the DPRAM used for the RRU FIFOs can be accessed by the user when the chip is in Configuration Mode. The write and read control transactions have a 16 bit wide data payload. The DPRAM blocks have 24 bit wide words. To complete a write operation into a memory location, the 16 Less Significant Bits [15:0] must be written first, followed by the 8 Most Significant Bits [23:16].

The Less Significant Bits of DPRAM locations are accessible with the **RRU MEB LSB** addresses. On a write transaction, the 16 bit data word is stored in a temporary register which will be written to the RAM location when the MSBs are written. Read transactions from these addresses retrieve bits [15:0] of the selected RAM word.

The Most Significant Bits of DPRAM locations are accessible with the **RRU MEB MSB** addresses. On a write transaction to these addresses, bits [7:0] of the data word are used to pad the 16 LSBs present in the temporary register and the resulting 24 bits word is written to the addressed RAM location. Read transactions from these addresses retrieve bits [23:16] of the selected RAM word.

Double Column Disable Registers

Set of 32 registers used to disable the readout of double columns in a specific region. When a bit is set the readout of the corresponding double column (Priority Encoder) is disabled.

Region Status Registers

Bits	**Field Name**	**Read/Write**	**Initial Value**
0	Pixel Status	R/W	-
1	Memory Status	R/W	-
2	All Column Disabled	R	-
15:3	Not used	-	-

Table 3.20: Region Status Registers, Field Description

- **Bit 0 - Pixel Status** This bit is set to 1 by the internal readout logic when a faulty pixel is detected during readout. A pixel is considered faulty when it can not be reset by the readout logic. When a faulty pixel is detected, the Priority Encoder containing the pixel is disabled by the internal readout logic, setting the corresponding bit of the Column Disable Register. The Pixel Status bits can be reset by writing a binary 0 to them.
- **Bit 1 - Memory Status** This bit is set to 1 by the internal logic when a malfunctioning DPRAM location is detected during a *Memory Self Test*. Memory Status bits can be reset by writing a binary 0 to them.

- **Bit 2 - All Columns Disabled** This read-only bit is set to 1 by the internal logic when all columns in the region are disabled by the user, via the **Double Column Disable Register**

3.2.4 Pixel Control Registers

In each pixel there are two write only, single bit registers: the Mask register and the Pulse Enable register. Writing to these registers is achieved by an addressing scheme driven by the periphery and based on a grid of orthogonal row and columns selection signals. There are 1024 row selection lines, two for each row of pixels since there are two registers in each pixel. There are 1024 column selection lines. A register in the pixel is enabled if the corresponding row selection signal and column selection signal are simultaneously asserted. When pixel registers are enabled, they latch the value of a global signal (PIXCNFG_DATA) distributed to all pixels. Read back of the registers inside the pixels is not possible. Further details on the configuration of the in-pixel latches are given in Section 3.6.

Each of the registers selection lines can be controlled individually through dedicated registers in the periphery. These peripheral registers are part of a set of registers called *Pixel Control Registers*. Another group of registers belonging to this set is used to gate the global pulse triggering signals that are distributed to the pixels.

There are 128 *Pixel Control Registers* registers (16 bit wide) in total. These are distributed in the thirty-two region modules of the periphery. Each region module contains:

- *two* 16 bits wide registers to steer 32 column selection lines
- *one* 16 bits wide register to control 16 Mask register row selection lines or 16 Pulse register row selection lines
- *one* 16 bits wide register to gate the global pulsing signal transmitted to the column pairs of the region.

The column selection registers in a region control the 32 column selection lines of the 32 pixel columns that are read out by the region. The top 16 pixel rows are configured from the row selection register in region 0, the bottom 16 pixel rows are configured from region 31. An additional global configuration bit (PIXCNFG_REGSEL) controls the forwarding of the values of the row selection registers in the regions to either the Mask registers row selection wires or to the Pulse registers row selection wires.

The read and write accesses to the peripheral Pixel Control Registers are implemented with the following address decoding scheme. The control transaction address is internally treated as composed of sub-fields:

- ADDRESS[15:11] = Region selection field
- ADDRESS[10:8] = Register group selection field
- ADDRESS[7] = Broadcast write bit
- ADDRESS[4] = Data toggle bit
- ADDRESS[3:0] = Register selection field.

Values of the sub-fields of the transaction address are detailed in table 3.21.

The four pixel configuration registers in each region are selected through the bits of the *Register selection field*. Therefore more than one register can be written at once with a single write transaction, by setting to 1'b1 more than one bit of this field.

A write transaction to any of the row or column selection registers with the *Data toggle bit* set to 1'b1 causes the toggling (change of the value to its ones' binary complement) of the

current content of the register. In this case, the data word of the write transaction is ignored. The content toggling feature does not apply to the Pulse Selection register.

Setting the *Broadcast write bit* in a write transaction has the effect of writing to the selected registers of all regions, ignoring the *Region selection field*.

For read back transactions, only one of the registers must be addressed.

Addr[15 : 11]	**Addr**[10 : 8]	**Addr**[6 : 0]	**Mode**	**Reset**	**Register or memory**
0x0-0x1F	0x4	0x01	R/W	0xFFFF	Column Selection Slice 1
0x0-0x1F	0x4	0x02	R/W	0xFFFF	Column Selection Slice 2
0x0-0x1F	0x4	0x04	R/W	0xFFFF	Row Selection Slice
0x0-0x1F	0x4	0x08	R/W	0xFFFF	Pulse Selection Slice

Table 3.21: Addressing of Pixel Control Registers driving the column selection, row selection and pulsing lines to the matrix.

0x0500 - Pixel Configuration Register (global)

Bits	**Field Name**	**Read/Write**	**Initial Value**
0	PIXCNFG_REGSEL	R/W	0x0
1	PIXCNFG_DATA	R/W	0x0
15:2	Not used	-	-

Table 3.22: Pixel Configuration Register Field Description

- **Bit 0 - PIXCNFG_REGSEL**
  Selection of the in-pixel register to be addressed: 0: for the Mask Enable register, 1: for the Pulse register. This bit is controlling the forwarding of the values of the row selection registers in the regions to either the Mask registers row selection wires or to the Pulse register row selection wires. The row selection lines of the pixel registers that are *not* selected are driven with 1'b0 and the corresponding pixel registers are disabled and maintain their values.

- **Bit 1 - PIXCNFG_DATA**
  Bit to be written in the target register. This bit is buffered and applied to the inputs of all the Mask and Pulse registers in the pixels.

3.2.5 DACs and Monitoring Control Registers

Address	Mode	Reset	Register or memory
0x0600	R/W	0x0400	Analog Monitor and Override Register
0x0601	R/W	0x0075	VRESETP
0x0602	R/W	0x0000	VRESETD
0x0603	R/W	0x0056	VCASP
0x0604	R/W	0x0039	VCASN
0x0605	R/W	0x00FF	VPULSEH
0x0606	R/W	0x0000	VPULSEL
0x0607	R/W	0x0040	VCASN2
0x0608	R/W	0x0000	VCLIP
0x0609	R/W	0x0000	VTEMP
0x060A	R/W	0x0000	IAUX2
0x060B	R/W	0x0032	IRESET
0x060C	R/W	0x0040	IDB
0x060D	R/W	0x0040	IBIAS
0x060E	R/W	0x0033	ITHR
0x060F	R/W	0x0000	Buffer Bypass Register
0x0610	R/W	0x0400	ADC Control Register
0x0611	R/W	0x0000	ADC DAC input value
0x0612	R	-	ADC Calibration Value Register
0x0613	R	-	ADC AVSS/Manual Value Register
0x0614	R	-	ADC DVSS Value Register
0x0615	R	-	ADC AVDD Value Register
0x0616	R	-	ADC AVDD Value Register
0x0617	R	-	ADC VCASN Value Register
0x0618	R	-	ADC VCASP Value Register
0x0619	R	-	ADC VPULSEH Value Register
0x061A	R	-	ADC VPULSEL Value Register
0x061B	R	-	ADC VRESETP Value Register
0x061C	R	-	ADC VRESETD Value Register
0x061D	R	-	ADC VCASN2 Value Register
0x061E	R	-	ADC VCLIP Value Register
0x061F	R	-	ADC VTEMP Value Register
0x0620	R	-	ADC ITHR Value Register
0x0621	R	-	ADC IREF Value Register
0x0622	R	-	ADC IDB Value Register
0x0623	R	-	ADC IBIAS Value Register
0x0624	R	-	ADC IAUX2 Value Register
0x0625	R	-	ADC IRESET Value Register
0x0626	R	-	ADC BG2V Value Register
0x0627	R	-	ADC T2V Value Register

Table 3.23: DACs and Monitoring Control Registers.

0x0600 - Analog Monitor and Override Register

Bits	Field Name	Read/Write	Initial Value
3:0	Voltage DAC selection	R/W	0x0
6:4	Current DAC selection	R/W	0x0
7	SWCNTL_DACMONI	R/W	0x0
8	SWCNTL_DACMONV	R/W	0x0
10:9	IRef Buffer current	R/W	0x4
15:11	Not Used	-	-

Table 3.24: Analog Monitor and Override Register Field Description

- **Bits 3:0 - Voltage DAC Selection**
  Voltage selected to DACMONV pin as per values in Table 3.25
- **Bits 6:4 - Current DAC Selection**
  Current selected to DACMONI pin as per values in Table 3.26
- **Bit 7 - SWCNTL_DACMONI**
  Configures the DAC block to enable the overriding of the selected current DAC. 0: for monitoring, 1: for overriding
- **Bit 8 - SWCNTL_DACMONV**
  Configures the DAC block to enable the overriding of the selected voltage DAC. 0: for monitoring, 1: for overriding
- **Bits 10:9 - IREF Buffer Current**
  IRef buffer current as per values in Table 3.27

Value	Selected voltage
0	VCASN
1	VCASP
2	VPULSEH
3	VPULSEL
4	VRESETP
5	VRESETD
6	VCASN2
7	VCLIP
8	VTEMP
9	ADCDAC

Table 3.25: Voltage DAC selection field values

Value	Selected current
0	IRESET
1	IAUX2
2	IBIAS
3	IDB
4	IREF
5	ITHR
6	IREFBuffer

Table 3.26: Current DAC selection field values

Value	Current
0	0,25 uA
1	0,75 uA (default)
2	1,00 uA
3	1,25 uA

Table 3.27: IREF Buffer current field values

0x601:0x060E - DACs Setting Registers

For each of the DACs, there is an 8-bit setting field. The default values for each of those are indicated in Table 3.23.

0x060F - Buffer Bypass Register

A value of 1 in the corresponding bit will bypass the buffer cell for that particular voltage or current.

Bits	Field Name	Read/Write	Initial Value
0	VCASN	R/W	0x0
1	VCASN2	R/W	0x0
2	VCASP	R/W	0x0
3	VCLIP	R/W	0x0
4	IRESET	R/W	0x0
5	IBIAS	R/W	0x0
6	ITHR	R/W	0x0
7	IDB	R/W	0x0
15:8	Not Used	-	-

Table 3.28: Buffer Bypass Field Description

0x0610 - ADC Control Register

Bits	Field Name	Read/Write	Initial Value
1:0	Mode	R/W	0x0
5:2	Select Input	R/W	0x0
7:6	Set Comparator Current	R/W	0x0
8	Discriminator Sign	R/W	0x0
10:9	Ramp Speed	R/W	0x2
11	Half LSB Trim	R/W	0x0
14:12	Not Used	-	-
15	Comparator Output	R	-

Table 3.29: ADC Control Field Description

- **Bits 1:0 - Mode**
  Mode of operation as per values in Table 3.30
- **Bits 5:2 - Select Input**
  Input selection as per values in Table 3.31

- **Bits 7:6 - Set Comparator Current**
  Comparator current as per values in Table 3.32
- **Bit 8 - Discriminator Sign**
  Control of the polarity of the ADC comparator
- **Bits 10:9 - Ramp Speed**
  Ramp speed as per values in Table 3.33
- **Bit 11 - Half LSB Trim**
  Fine control of the offset of the DAC inside the ADC. Used for ADC calibration procedures
- **Bit 15 - Comparator Output**
  Value of comparator output (read only)

Value	Mode of Operation
0	Manual
1	Calibration
2	Auto
3	Super-manual

Table 3.30: ADC mode of operation field values

Value	Selected Input
0	AVSS
1	DVSS
2	AVDD
3	DVDD
4	V band-gap through voltage scaling
5	DACMONV
6	DACMONI
7	Bandgap (direct measurement)
8	Temperature (direct measurement)

Table 3.31: ADC input selection field values

Value	ADC comparator current
0	163 μA
1	190 μA
2	296 μA (nominal)
3	410 μA

Table 3.32: ADC comparator current control values

Value	Ramp Speed
0	500ns
1	1us (default)
2	2us
3	4us

Table 3.33: ADC ramp speed field values

0x0611 - ADC DAC Input Value Register

In Super-Manual mode sets the input reference value for the DAC

0x0612 - ADC Calibration Value Register

Sampled Value when ADC is in Calibration mode

0x6013:0x0627 - ADC Value Registers

For each of the ADC input, there is an 11-bit register with the corresponding sampled value. The values for each of those are indicated in Table 3.23.

3.2.6 Test and Debug Control Registers

Address	Mode	Reset	Register or memory
0x0700	R	-	SEU Error Counter
0x0701	R/W	0x0000	Test Control Register
0x0702	R	-	BMU Debug Stream
0x0703	R	-	DMU Debug Stream
0x0704	R	-	TRU Debug Stream
0x0705	R	-	RRU Debug Stream
0x0706	R	-	FROMU Debug Stream
0x0707	R	-	ADC Debug Stream

Table 3.34: Test And Debug Control registers.

0x0700 - SEU Error Counter Register

Incremented every clock cycle while in any of the sets of triplicated flip-flops (TMR) there is a mismatch between the values of the three flip-flops. Note that since the auto correction of the mismatch requires normally 2-3 clock cycles, the counter can get incremented of few units for every SEU event.

0x0701 - Test Control Register

Bits	Field Name	Read/Write	Initial Value
0	Force Local Bus	R/W	0x0
8:1	Force Local Bus Value	R/W	0x0
9	Force FROMU Busy	R/W	0x0
10	Force Disable Busy Line	R/W	0x0
15:11	Not Used	-	-

Table 3.35: Test Control Register

- **Bit 0 - Force Local Bus**
  0: normal operation of Local Bus, 1: Local Bus is forced to the value specified in bits

8:1.
Note that ALPIDE must be in Outer Barrel configuration mode and have the token (Initial Token set), to access the Local Bus for writing

- **Bits 8:1 - Force Local Bus Value**
  Value forced in the Local Bus
- **Bit 9 - Force FROMU Busy**
  When set, forces FROMU to enter BUSY condition
- **Bit 10 - Force Disable Busy Line**
  When set, input value of Busy Line is forced to be considered de-asserted

0x0702 - BMU Debug Stream

Address to read the chain of shadow registers of the Busy Management Unit (ref. section 4.4).

0x0703 - DMU Debug Stream

Address to read the chain of shadow registers of the Data Management Unit (ref. section 4.4).

0x0704 - TRU Debug Stream

Address to read the chain of shadow registers of the Top Readout Unit (ref. section 4.4).

0x0705 - RRU Debug Stream

Address to read the chain of shadow registers of the Region Readout Unit (ref. section 4.4).

0x0706 - FROMU Debug Stream

Address to read the chain of shadow registers of the FROMU Unit (ref. section 4.4).

0x0707 - ADC Debug Stream

Address to read the chain of shadow registers of the ADC Logic (ref. section 4.4).

3.3 Triggering and Framing

The core functionality of the ALPIDE is to store the internal state of the front–end discriminators within every pixel and transmit this information off chip. The collection of pixel states at a particular time is called a frame or a snapshot. A frame will be generated and trasmitted off chip following the reception of a TRIGGER. The use of this functionality is described in detail within the current section.

Each pixel in the ALPIDE Matrix has 3 in-pixel data storage elements, called buffers. The three buffers together form what's referred to as a Multi event buffer (MEB). This enables the storage of 3 complete frames without the completion of a matrix readout or any data loss.

Thus, the writing and reading of frames is based on the management of the pixel MEBs. The management scheme is implemented by the Framing and Management Unit (FROMU). This is illustrated in Figure 3.7.

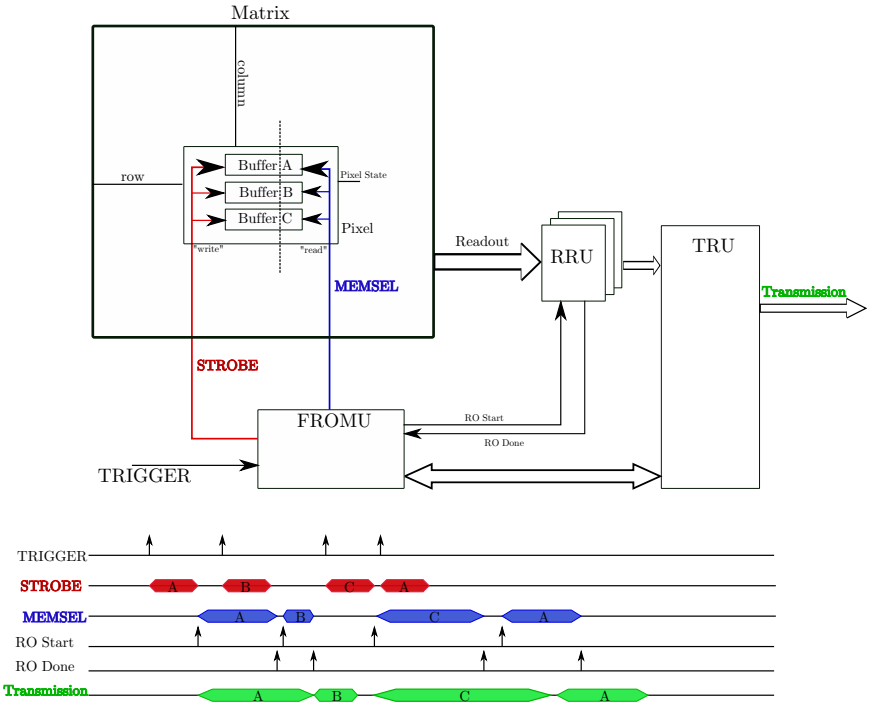

Figure 3.7: Pixel MEB Management Scheme

The recording of the pixel discriminator outputs in the storage registers is controlled by three global signals (STROBE[2:0]) distributed from the periphery to all pixels. A TRIGGER, which can be generated internally or sent externally via the control interface, will initiate the timed assertions of the STROBE signals. This timed STROBE assertion is called framing interval or window. The outputs of the latches in the pixels can be selectively connected to the inputs of the Priority Encoders. This is governed by three global signals (MEMSEL[2:0]). Only one bank of registers can be selected at any time for readout. The STROBE and MEMSEL signals act as write and read pointers to the matrix event buffers. The orderly generation of the STROBE signals in response to trigger commands and the updating of the MEMSEL signals on completion of frame readout is the main functionality of the FROMU module.

The assertion of the STROBE ad MEMSEL signals to the matrix buffers is sequenced such that a circular buffer is effectively implemented. By design, only one of the three STROBE signals and one of the three MEMSEL signals can be asserted during normal operation but never pointing to the same in-pixel latch.

The readout of a pixel buffer slice starts at the end of the corresponding STROBE assertion. The 32 matrix regions are read out concurrently by 32 region readout unit (RRU). Each RRU stores the data in local memories. The Top Readout Unit (TRU) fetches the information from each RRU and transmits it off chip.

The ALPIDE provides for two MEB management schemes, called readout modes.

3.3.1 Readout Modes

The readout logic support two operating modes: *Triggered* and *Continuous*. The operating mode can be selected by **Mode Control Register (0x0001)** (see section 3.2.2).

The **triggered** mode is intended to sample over a relatively short interval of time the status of the discriminated outputs of the pixels. The coincidence between a discriminated signal and the assertion of a strobe signal determines the latching of the pixel hit. All pixel firing during the assertion of the strobe are latched as hits. The duration of the strobing interval is expected to be set to a few hundredths of nanoseconds. The start of the strobing intervals is tpically controlled by an external trigger source.

The **continuous** mode is intended to provide the readout of pixel hits sampled during periodically repeating strobing intervals, with a duration equal to the interval between two consecutive ones. The framing intervals would typically be longer (order of a few μs) compared to those used in triggered mode. The gap between framing intervals is maintained as short as possible (order of 100 ns).

The framing intervals can be controlled by external commands (triggers) or by an internal sequencer, providing flexible control of strobe duration and repetition rate (see section 3.3.6).

While the two modes are expected to operated with different framing configurations, the only difference between them is related to the logic handling the condition of in-pixel multi-event buffers becoming full and no storage space is left for subsequent frames.

Triggered Mode

In **triggered** mode, the chip prioritises events that are already stored in the matrix over new incoming triggers. The ALPIDE readout logic priorities the completion of the readout of the frames pending in the matrix. A trigger command received when the matrix buffers are all full will not generate a new strobing interval. The trigger command will still be acknowledged in the data stream with an empty chip data packet. Additionally, the occurrence of such an event is indicated by one of the feedback flags in the chip data packet trailer (refer to Section 3.4.1).

Continuous Mode

In **continuous** mode, the chip prioritises newly received frame requests over data that are already stored within the matrix. The ALPIDE needs to ensure that there is always at least one available storage element in the matrix such that forthcoming frames can be stored. This is guaranteed by forcedly deleting the pixel hit data of the frame that is being read out, to free space for a future one. Thus, on reception of a framing request to the last free matrix buffer, the chip will interrupt the presently ongoing frame readout in order to free up one of the matrix MEB slices. The data packet associated with the interrupted readout will contain any data that was readout until the interruption. Additionally, the activation of this mechanism and the partial data loss are reported by one of the feedback flags in the chip data packet trailer (refer to Section 3.4.1).

3.3.2 Frame Timestamping using Bunch Crossing Counter

The ALPIDE implements an internal time reference. There is a 12 bit wide continuosly running up-counter wrapping around after reaching the maximum value of 12'd3563. This

maximum value is matching the duration of one LHC orbit. If the chip is operated with a clock synchronous with the LHC clock, this provides an on-chip bunch crossing (BC) counter.

The ALPIDE BC Counter has a synchronous reset that can be issued via a BCRST command. The BCRST command allows to synchronize the counters across chips and align them with a time reference in the experiment.

The beginning of a framing interval causes the latching of bits [10:3] of the BC counter, providing a frame time stamp. The time stamp is transmitted within the Chip Data Packet Header (see Section 3.4.1).

3.3.3 Matrix Memory Bank Masking

The ALPIDE provides the functionality to mask any of the Matrix memory slices to prevent their usage for hit data storage. This setting affects the order of assertions of STROBE and MEMSEL signals. The remaining active memory buffers are still accessed in a rotating fashion. Figure 3.8 illustrates the rotating pattern obtained if a MEB slice C is masked.

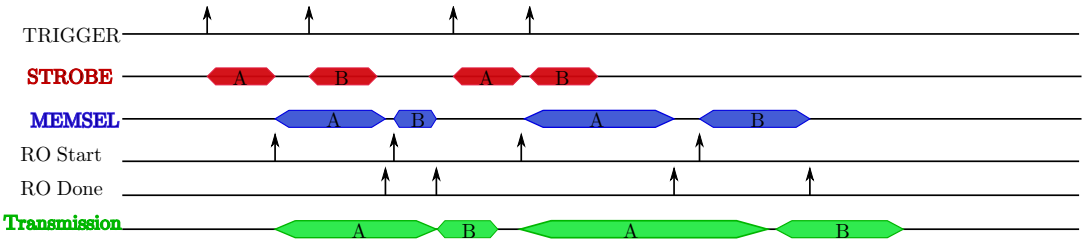

Figure 3.8: Pixel MEB Management with Slice C Masked

This functionality is accessed via the Pixel MEB Mask Field of FROMU Configuration Register 1 (see Table 3.5).

Note that while this feature is available in both readout modes, it does not make sense to be used in Continuous mode. If even a single matrix slice is masked, the readout associated with the first frame will start briefly after the de-assertion of the STROBE signal. The second trigger is likely to arrive immediately after, thus causing the first frame to be discarded almost immediately.

Further, if all 3 slices are masked, the chip will recognise the likely programming error and treat the condition as if no slices are masked.

3.3.4 STROBE Window Duration

The duration of the STROBE signals applied to the pixels can be programmed using **FROMU Configuration Register 2 (0x0005)**. A minimum duration of 2 CLK cycles (50 ns) is recommended for this setting.

The STROBE is asserted two cycles after the FROMU receives a trigger command. An internal counter then counts up to the preprogrammed duration before the STROBE is de-asserted.

3.3.5 STROBE Window Extendability

If a trigger is received whilst a framing window is still active (i.e. the STROBE signal is asserted to one of the 3 latches), the window duration is extended as if it has just started. Thus, the overall framing window duration becomes whatever time had elapsed before the reception of the second TRIGGER plus the full programmed duration. The occurrence of this event is marked by the setting of STROBE EXTENDED bit in the trailer flags for the data

frame in question (refer to Section 3.4.1). Figure illustrates the occurrence of an extended STROBE.

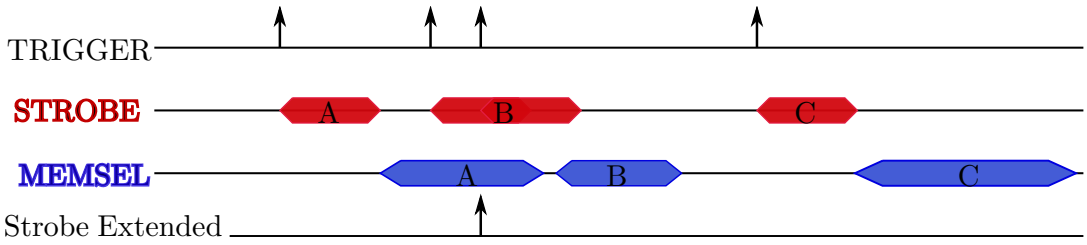

Figure 3.9: An illustration of an extended STROBE window followin the reception of a second TRIGGER

3.3.6 Internal Sequencer

The FROMU includes the functionality to internally generate sequences of periodic triggers. The internally generated triggers are processed in the same way as external ones. Once a framing window is complete, i.e. the STROBE is de–asserted, the internal sequencer does not assert the next STROBE signal until a time duration set by **FROMU Configuration Registers 3** has elapsed. Thus the rate of generation of internal triggers is given by the sum of the settings for the Strobe Duration and Strobe Gap.

In order to use the internal sequencer one needs first to enable it via **FROMU Configuration Register 1** The repeating sequence requires a single initial external trigger. Once started, the sequencer will generate internal triggers until the aforementined configuration bit is deasserted.

The internally generated pattern of triggers can be re–timed via another external trigger. Note that this behaviour is compatible with the option to extend a STROBE window (see Section 3.3.5). A timing diagram of the generated sequence and the retiming behaviour can be seen in Figure 3.10.

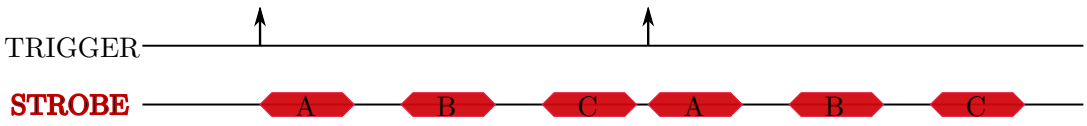

Figure 3.10: Internal TRIGGER Sequencer Initiation and Re-Timing

3.3.7 Triggering and Waveform Diagrams

3.4 Data Transmission

The ALPIDE has two means of data transmission - a serial transmission port (HSDATA_P / HSDATA_N) and a parallel data port (DATA[7:0]). Both of these utilise the same byte-oriented data protocol, described in Section 3.4.1. The two ports are intended to implement the data interfaces on the ITS Modules as described in the Appendices B and C.

A summary of how the two interfaces are utilised can be seen in Table 3.36.

ALPIDE Configuration	Serial Link Utilised	Parallel Port Utilised	Description
Inner Barrel	Yes	No	Programmable Serial Link Speed. Default is 1200 Mb/s, but can optionally use 400 Mb/s or 600 Mb/s
Outer Barrel Master	Yes	Yes	Samples the parallel port and forwards stream to Serial Link Transmission. Serial Link transmission at 400 Mb/s
Outer Barrel Slave	No	Yes	Transmits data over parallel port. Data is then to be serialized and transmitted on Serial Link by Master

Table 3.36: Summary of Transmission Interface Utilisation Depending on Chip Configuration

On the serial port data is 8b/10b encoded with the LSB launched first. The COMMA control word used is K28.5. Thus, the decoded COMMA value of 8'hBC is transmitted as either 10'b001111_1010 or 10'b110000_0101 depending on the running disparity.

The COMMA transmission can be used for clock recovery and synchronisation to the data stream. The COMMA word can be observed in the following scenarios:

- When there is no readout in progress and the chip is completely idle.
- When there is a readout in progress but the chip hasn't had enough time to process data and/or pack it.

The parallel data port, also referred to as the local bus, is clock synchronous (40 MHz) and has two modes of operation. In the default mode, the full byte is transmitted over pads DATA[3:0] using Double Data Rate transmission. Each byte is sent in two consecutive nibbles with the most significant nibble (bits 7–4) first. Nibbles are launched on both rising and falling edges of the chip clock. Therefore, a 320 Mb/s stream is produced.

The chip can be optionally configured to utilize Single Data Rate transmission on the full bus DATA[7:0], with a byte being launched at every clock rising edge (see CMU and DMU config register description).

The uppermost DATA[7:4] bits are always clocked on a positive clock edge i.e. they always operate in Single Data Rate mode.

There is no data encoding on the local bus and the protocol described in Section 3.4.1 is used directly. A sample local bus stream utilising DDR transmission and the corresponding serial transmission can be seen in Figure 3.12.

3.4.1 Data format and valid data words

Table 3.37 lists the valid data words. The valid data words are identified by predefined prefix bit strings.

Data Word	Length (Bits)	Value (binary)
IDLE	8	1111_1111
CHIP HEADER	16	1010<chip_id[3:0]><BUNCH_COUNTER_FOR_FRAME[10:3] >
CHIP TRAILER	8	1011<readout_flags[3:0]>
CHIP EMPTY FRAME	16	1110<chip_id[3:0]><BUNCH_COUNTER_FOR_FRAME[10:3] >
REGION HEADER	8	110<region_id[4:0]>
DATA SHORT	16	01<encoder_id[3:0]><addr[9:0]>
DATA LONG	24	00<encoder_id[3:0]><addr[9:0]>_0_ <hit_map[6:0]>
BUSY ON	8	1111_0001
BUSY OFF	8	1111_0000

Table 3.37: Data Format adopted in ALPIDE chip.

IDLE - 1111_1111 The IDLE word is used as a filler whenever data is not ready to be transmitted.

CHIP HEADER Data word transmitted at the beginning of each data packet. Bits CHIPID[3:0], which are sent as the least significant nibble of the first CHIP HEADER byte, identify the geographical location of a given chip on a ITS module. Additionally, the second byte of the CHIP HEADER word contains BUNCH_COUNTER_FOR_FRAME[10:3], the latched value of the internal Bunch Crossing counter at the time of reception of the trigger corresponding to the data packet (See Section 3.3.2).

CHIP TRAILER Byte transmitted at the end of each data frame.

The <readout_flags[3:0]> data field consists of 4 flags, which transmit information about the event in question and the state of the chip. The data composition is as follows:

$$< readout\_flags[3:0] >= \{< BUSY\_VIOLATION >< FLUSHED\_INCOMPLETE > \\ < STROBE\_EXTENDED >< BUSY\_TRANSITION >\}$$

The flags outlined above are generated as follows:

- BUSY_VIOLATION – indication that the chip is replying with an empty data packet due to saturation of data processing capabilities.
- FLUSHED_INCOMPLETE – indication that a MEB slice was flushed in order to ensure that the MATRIX always has a free memory bank for storing new events. Observed in Continuous mode only.
- STROBE_EXTENDED – indication that the framing window for the event of question was extended due to the reception of an external trigger.
- BUSY_TRANSITION – indication that the BUSY was asserted during the **readout** of the frame in question.

Note that the flags descirbed above can be overridden in special circumstances:

1. If the FATAL bit was asserted, the <readout_flags[3:0]> field is set to a constant 4'b1110, regardless of whether any of the standard flags were asserted for the event in question.
2. If FATAL is not asserted but the event is transmitted when the chip is in DATA OVERRUN MODE, the standard flags are overridden by a constant 4'b1100 pattern.

A summary of the readout flag configurations that can be observed is shown in Table

Readout Flags Configuration	Description of event
4'b1000	Busy Violation Event - Observable in a data packet consisting of solely a CHIP HEADER + CHIP TRAILER only.
4'b0xxx	Readout event. Meaning of flags is as described previously. Flags are ortogonal and can be set independently of one another provided that the event they describe has occurred.
4'b1100	Indicates that the event is transmitted in DATA OVERRUN mode and any event flags in the Start FIFO are overridden. The flag override will be in place until the Start FIFO is fully emptied.
4'b1110	Indicates that the event is transmitted whilst the FATAL condition is asserted and any event flags in the Start FIFO are overridden. This means that the identifier information for at least one event has been lost due to FIFO overflow. The readout flag configuration will be maintained until a GRST/RORST is issued to clear the FATAL condition even if the chip recovers as a result of DATA OVERRUN mode and is back in sync.

Table 3.38: A summary of observable readout flag scenarios

CHIP EMPTY FRAME The CHIP EMPTY FRAME is used when a fully readout event is actually completely empty. This data word forms a data packet on its own and therefore contains CHIPID and BUNCH_COUNTER_FOR_FRAME[10:3] fields as described for the CHIP HEADER word.

REGION HEADER Data word used to indicate the beginning of transmission of data for a particular region. The variable part *region_id[4:0]* is the index of the region. Region data frames are sent sequentially in ascending order. The region header is only sent for regions with pixel hit information. If a region has no hits, its header is omitted and the transmission moves on to the next one that does have hits.

DATA SHORT Data word containing the geographical location of a single pixel. The *encoder_id* is the index of the priority encoder inside a region and *addr* is the pixel hit index generated by the priority encoder.

DATA LONG Data word used to enable the compression of up to 8 pixels via a bit map. This word is only ever transmitted if clustering is enabled. The *encoder_id* and *addr* fields, indentically to the DATA SHORT word, contain the geographical information of the first pixel, i.e. the one with the lowest address, in a group. The *hit_map[6:0]* contains the cluster shape information in the form of a bit map. A bit in the hit_map is set for any active pixel among the 7 immediately after (based on PE pixel addreses) the one indicated by the addr[9:0] field.

The LSB of hit map corresponds to first subsequent pixel and bit 6 to the 7th. Figure 3.11 illustrates this for a sample cluster.

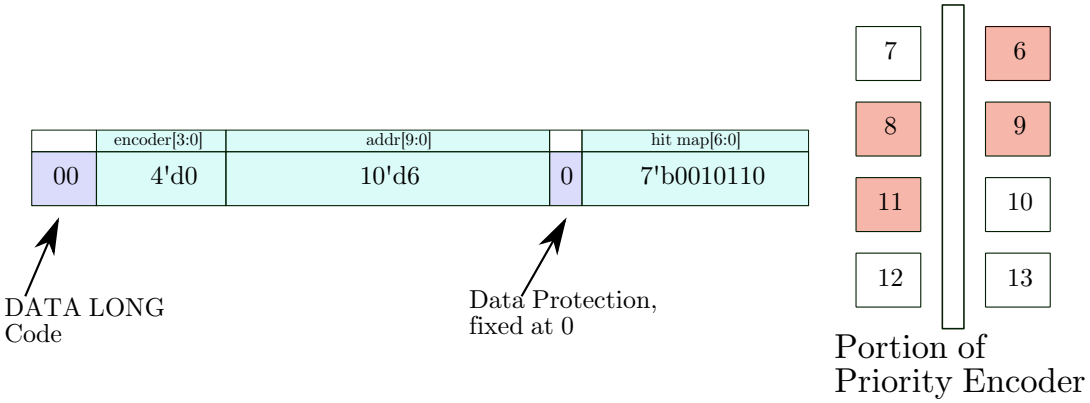

Figure 3.11: A depiction of a cluster hit and its DATA LONG representation

BUSY ON Code word transmitted on assertion of the BUSY status. Transmitted on both the serial link and on the parallel bus. See Section 3.5 for further notes on the BUSY signaling mechanism.

BUSY OFF Code word transmitted on the serial port on de-assertion of the BUSY status. Transmitted on both the serial link and on the parallel bus.

3.4.2 Data format rules

- Data words are transmitted byte by byte, most significant byte first.
- IDLE, BUSY_ON and BUSY_OFF words can be arbitrarily inserted betweeen other code words.
- Words that are 16 or 24 bits (DATA SHORT, DATA LONG, CHIP HEADER, CHIP EMPTY FRAME) cannot be split by a IDLE or a BUSY.
- The BUSY words are transmitted as soon as possible without violating data integrity.

3.4.3 Local Data Bus Sharing Mechanism

The OB Slave chips send their data to the OB Master using a shared Local Data Bus implemented using the DATA parallel port. Time division multiplexing of the write access to the bus is governed by a *virtual token* exchange mechanism. Only one chip at a time gets the right to write to the Local Bus. The chip with the token is allowed to enable the drivers onto the DATA ports. The DATA I/Os are equipped with internal pull-ups and the inactive state of the Local Bus is **xFF** (IDLE code word). This is described in further detail in Appendix C.

The OB Module chip continously sample the bus and write to it in turn. The token is effectively constituted by the CHIP TRAILER data word of a Chip Data Frame and by the CHIP EMPTY FRAME in case of an empty frame. The chips monitor the Local Bus waiting for the previous chip in the sequence to complete the transmission of its own data frame. Once a chip obtains the right to write onto the bus, it transmits one complete Chip Data Frame and releases the bus.

Following a reset, the token is assigned to the chip that has the initial_token field set to 1 (see CMU and DMU Config Register). It is the user's responsibility to ensure that only one chip in a module has this bit set to 1 to ensure that the local bus is not driven by two devices.

The token passing order is established through the Previous Chip ID field (see CMU and DMU Config Register) of each chip. It is the user's responsibility that the full loop integrity is ensured i.e. that the chip with the initial_token field set to 1 has its Previous Chip ID pointing to the last device in the sequence.

It is possible to bypass selected chips (both Master and Slaves) in the readout sequence. Should the Master device be bypassed, it continues to monitor the local bus and transmit data via the serial link.

Should there be no pending frames, the bus is left undriven. In this case the bus is maintained to the idle state (IDLE code) by the pull-ups.

3.4.4 Sample Data Streams

Outlined in Figure 3.12 are two sample data streams that would be observed on the Serial and Local Bus respectively for an identical frame. The Master samples the local bus irrespective of which device is driving it and transmits the data on the serial link.

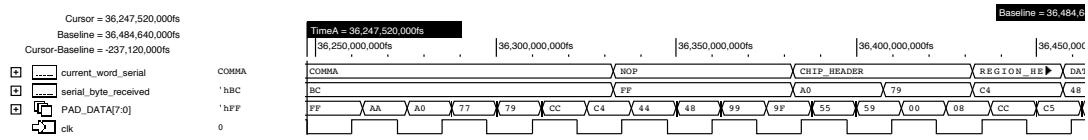

Figure 3.12: A Sample Data Stream on the Local Bus in DDR Mode and the Corresponding Transmission on the Serial Bus

3.5 Busy State and Conditions

The ALPIDE's functionality includes the monitoring of the internal state of circuitry. Should the chip be near saturation of its data processing capabilities, it will communicate this by asserting the BUSY state to the outside world. The BUSY data word and the data protocol is discussed in detail in Section 3.4.

The specific bottleneck that is causing the chip to assert the BUSY can be determined based on the readout mode and the observed data stream. The user can make the decision whether to reduce the trigger rate at the cost of some data loss or proceed without change.

3.5.1 Signaling of the Busy State

Once the chip becomes Busy, it will, by default, utilise the BUSY protocol word (see Section 3.4.1 for on transmission protocol) to transmit this information to the outside world in one of two ways:

- In an OB configuration, the Slave chips and the Master chip share a wire connected to their BUSY pins. Once a chip on the half–module becomes busy, the shared BUSY line will be asserted. The chip which has the transmission token will then transmit the BUSY word as soon as it observes it without breaking the local bus data transmission rules (see Section 3.4.2). The master chip will then sample the local bus as per usual and transmit the BUSY on the serial link.
- In an IB configuration, the BUSY word will directly be transmitted on the serial link.

Additionally, the user can recognise the transition to a BUSY state via the BUSY TRANSITION trailer flag. The flag will be set in the data packet for the event, whose readout is completed immediately following the busy transition in question.

3.5.2 Busy in TRIGGERED Mode

In **triggered mode**, the chip will become Busy following the assertion of a framing window to the last available matrix slice such that there are no in-pixel buffers left available. The busy will be de–asserted once there available buffers i.e. once the ongoing matrix frame readout is completed.

3.5.3 Busy in CONTINUOUS Mode

In **continuous mode** the chip gets BUSY at the beginning of the strobing interval of the second available matrix slice. At this time only one additional matrix MEB slice remains free for storing a next frame. The busy is deasserted when two free buffers are available.

If a further TRIGGER is received while the BUSY is asserted, the FROMU will assess the state of the pixel MEBs. If at least one buffer is available, the strobing window is asserted to the pixels. In parallel, the currently ongoing readout of the matrix frame will be interrupted, pending hit data flushed and the associated chip data packet qualified with the FLUSHED_INCOMPLETE flag to indicate the data loss (see Section 3.4.1 on Data Protocol).

Under specific circumstances, the data flushing mechanism can be insufficient to guarantee the availability of MEB slices. In these cases the logic does not assert the strobe to the matrix and an empty chip data packet with BUSY VIOLATION flag set is scheduled for transmission.

3.5.4 Busy Monitoring

In an OB configuration, the ALPIDE can be configured to take into account the state of the shared BUSY line for the purposes of its individual MEB management. Thus, if the shared BUSY line is asserted and this functionality is enabled, the ALPIDE will behave as if it is itself BUSY even if the shared line was asserted by another device.

3.5.5 Busy Testing Functionalities

For testing or operation purposes, the BUSY State can be overridden by the user. This functionality is accessed via the Force Busy Field of the *Test Control Register (0x0701)*. The chip will then behave as if the BUSY has been asserted due to filling of the matrix slices and according the the programmed readout mode. The BUSY will then be de–asserted once the user releases the force bit.

Further, the user can disable the communication of the BUSY condition via the Force Busy Val field of the*Test Control Register (0x0701)*. The ALPIDE will not assert the BUSY line nor transmit the BUSY word on the serial link The chip will still use the internal busy state, whether that is genuine or forced, in the way described previously and according the the programmed readout mode. An internal transition of BUSY of a chip will still be detectable via the BUSY TRANSITION flag of the Chip Data Packet trailer.

3.5.6 Busy associated with internal FIFOs and measures to prevent overflow

As previously stated, the ALPIDE acknowledges every TRIGGER with a data packet, which includes a corresponding time stamp (See Section 3.3.2). A **Start FIFO** guarantees the orderly transmission of this information. It is written at the time the readout is initiated. The FIFO is popped when the TRU is ready to begin constructing a new data packet for transmission.

The FIFO is 64 deep. When the FIFO has 48 events, referred to as the ALMUST FULL 1 marker, in the pipeline for transmission, the chip indicates that it is close to saturation by asserting the BUSY. The intention is that user would acknowledge this and reduce the trigger rate. Any further TRIGGERs at that point are still written to the FIFO but marked as BUSY VIOLATIONs (regardless of the readout mode) and are thus transmitted as empty packets. The BUSY state is de-asserted once the number of events drops below 48.

Should the FIFO reach 56 events, or the ALMOST FULL 2 marker, the chip enters DATA OVERRUN mode. The intention of the DATA OVERRUN mode is to empty the FIFO and prevent loss of synchronism. In DATA OVERRUN mode, all frame information in the RRU memories is discarded. All events are transmitted as empty data packets, maintaining the correct time stamp and indicating that they are empty due to the occurrence of DATA OVERRUN mode via specific readout flags (see Section 3.4.1). DATA OVERRUN mode is exited once there are no events in FIFO pipeline, at which point the BUSY is de-asserted as well. Note that the chip will still acknowledge any TRIGGERs received whilst in DATA OVERRUN mode.

Finally, if the trigger rate is so high that even DATA OVERRUN mode cannot prevent the FIFO from reaching the full count of 64 events, the FATAL bit is set. Every subsequently transmitted event, even if the chip does recover and leaves panic mode, has the FATAL trailer flag set indicating that there has been FIFO overflow and an event has been lost (See Description of Trailer Flags in Section 3.4.1). The FATAL bit is cleared via a RORST/GRST command.

3.6 Configuring pixels for masking and pulsing

Every pixel contains a Mask register and a Pulse Enable register within it.

The Mask register can be used to force low the digital output of the pixel. A masked pixel is skipped by the readout as any other pixel without an active hit. Therefore it will not appear in the event frames.

The pixels contain built-in testing features. They can be forced to produce a hit both using a test charge injection capacitor (*analog pulsing*) or directly setting the pixel state register (*digital pulsing*). The pulsing functionalities are enabled with the Pulse Enable register. Additional global pulsing signals are used to trigger the injection of the charge in the front-end or to set the state register once the pulsing is enabled in the desired pixels. Details on the pixel digital section including masking and pulsing are given in 4.1.2.

This section describes the writing of the Mask and Pulse Enable pixel registers using specific registers implemented in the periphery and accessed through the control interface. The triggering of the pulsing functionality and other related features are described in Section 3.7.

3.6.1 Description of PULSE_EN and MASK_EN latches and of the matrix configuration management scheme

The in-pixel Mask register (MASK_EN) and Pulse Enable (PULSE_EN) register are implemented with latches, as illustrated in Fig. 3.13. Each register is individually addressable, allowing the writing of arbitrary masking or pulsing patterns. The Mask and Pulse Enable registers are write-only. They can not be read back directly.

Each pixel configuration circuit receives 4 signals driven from the periphery, namely DATA, COLSEL, ROWREGM_SEL and ROWREGP_SEL. The two latches are enabled by the simultaneous assertion of a column select wire (COLSEL) and a dedicated row select line (ROWREGM_SEL or ROWREGP_SEL). When a latch is enabled, its content is updated with the value of the global DATA signal that is propagated to all the latches in the matrix. When the selection signals are de-asserted, the latches maintain their content until it is updated by another configuration cycle. The PULSE_EN and MASK_EN latches are not affected by the Pixel Reset (PRST) command. The pixel latches do not provide a reset mechanism,

the value after power-on is unknown. **The user shall initialize the in-pixel registers**. It is also noted that the pixel latches are not protected against SEU effects, thus a periodic refresh of the mask pattern might be required when the chip is operating in ionizing radiation.

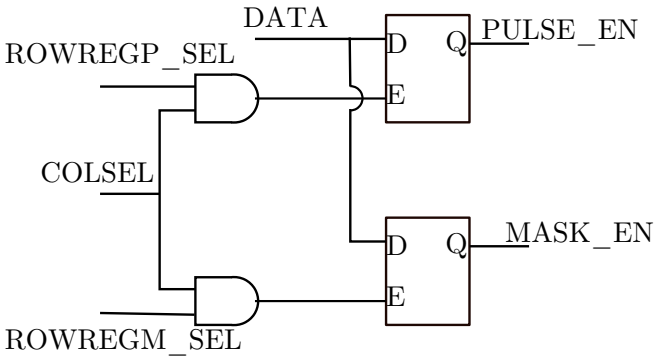

Figure 3.13: Schematic of in-pixel latch circuitry

Writing to the pixel registers is achieved with an addressing scheme based on a grid of orthogonal row and columns selection lines illustrated in Fig. 3.14. There are 1024 column selection lines (COLSEL), 512 lines to select the Mask latches in rows (ROWREGM_SEL) and 512 lines to select the Pulse Enable latches (ROWREGP_SEL). The row and column selection lines are driven by registers distributed in the periphery across the 32 region modules (Fig. 3.15).

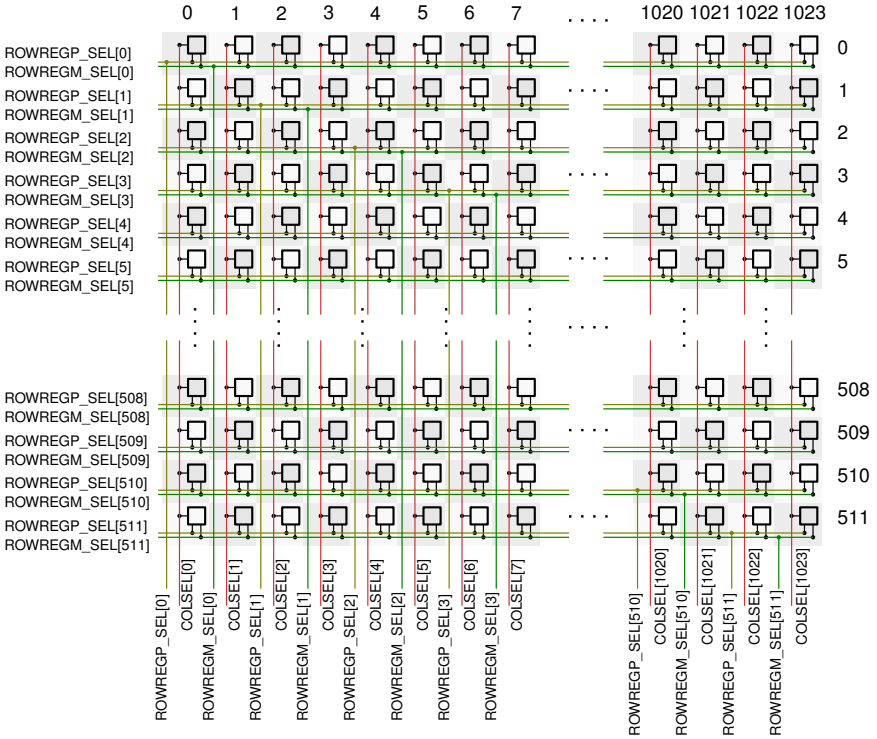

Figure 3.14: Topology of the row and column selection lines to configure the in-pixel Mask and Pulse Enable latches.

Each region module contains specific registers controlling signals related to the configuration or pulsing of the pixels:

- *two* 16 bits wide registers to steer 32 column selection lines (*Column Selection Slice 1* and *Column Selection Slice 2*)

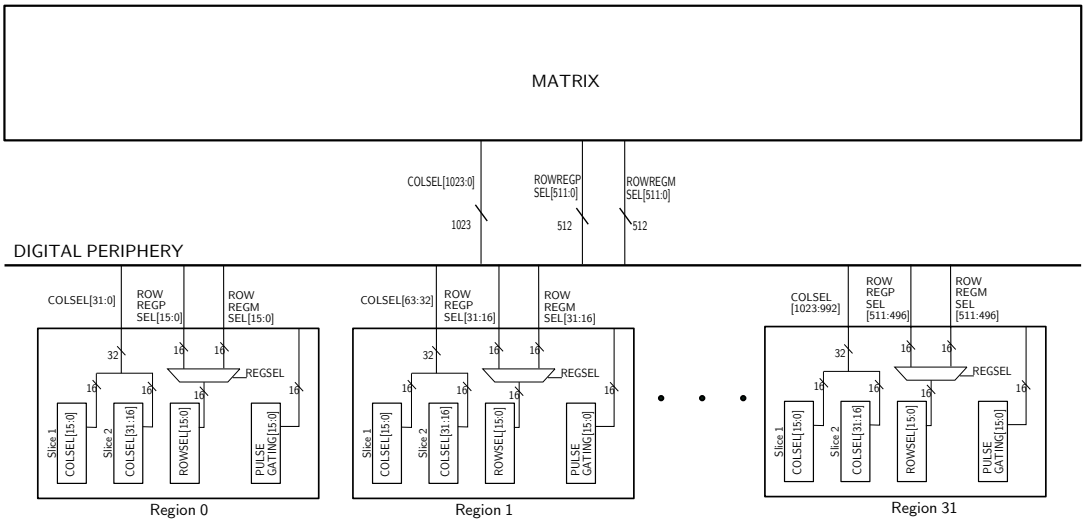

Figure 3.15: Scheme of the distribution of the peripheral registers controlling the writing to in-pixel registers and the gating of pulsing signals.

- *one* 16 bits wide register (*Row Selection Slice*) to control 16 Mask register row selection lines or 16 Pulse register row selection lines
- *one* 16 bits wide register (*Pulse Gating Slice*) to gate the global pulsing signal propagated to the column pairs of the region.

These constitute a set of 128 *Pixel Control Registers* registers (16 bit wide) called *Pixel Control Registers*.

The bits of the column select registers are buffered and applied to the COLSEL lines. Region 0 controls the slice of the 32 leftmost columns (0-31), while region 31 controls the slice of 32 rightmost columns (992-1023). Register *Column Selection Slice 1* drives the 16 leftmost column selection wires within the region, register *Column Selection Slice 2* controls the 16 rightmost ones. The less significant bits of the registers (bit index 0) drive the leftmost columns of each group.

The *Row Selection Slice* register (16 bit wide) of a given region is used to control 16 row selection lines of Mask registers (ROWREGM_SEL) and 16 lines of Pulse Enable registers (ROWREGP_SEL). The row selection register of region 0 controls the row selection lines of pixel rows 0-15, the one in region 1 is dedicated to pixel rows 16-31 and so on up to the register in region 31 controlling pixel rows 496-511. The less significant bit (index 0) drives the topmost line of a slice. Each bit of the row selection registers can control *either* a ROWREGM_SEL wire *or* a ROWREGP_SEL wire. A digital 1-to-2 demultiplexer drives one or the other line with the value stored in the register bit.

The PIXCNFG_REGSEL bit of the **Pixel Configuration Register (0x0500)** controls the array of demultiplexers and is used to select between the two configurations. Binary value 0 has the effect of driving of the masking lines (ROWREGM_SEL) with the value stored in the *Row Selection Slice* register. Binary value 1 causes the driving of the pulsing lines (ROWREGP_SEL) with the same register value. The inactive set of row selection wire is driven with a low value (latches disabled). Thus, only a subset of one of the two banks of configuration latches of the pixels can be enabled at any given time and the user can write to either Mask Registers or to Pulse Enable Registers.

The DATA inputs of the in-pixel latches are driven by a global signal distributed (with buffering) to all pixels. This signal is controlled via the PIXCNFG_DATA bit of the **Pixel Configuration Register (0x0500)**.

In addition to the column selection and row selection registers, each region contains a 16 bit wide Pulse Gating register. These registers determine if a Test Pulse command, be it ANALOG or DIGITAL, is propagated to the pixels, on a double-column basis (refer to section 3.7 for more details on pulsing).

3.6.2 Control of Column Select and Row Select lines. Control of Pulse gating and propagation of pulsing signals

The peripheral registers described in the previous section, controlling the configuration of the pixels and the propagation of Test Pulse commands, are accessible via the control interface. The addressing of these registers was designed with the intent of providing flexibility and efficiency in the management of the matrix configuration, aiming at minimizing the number of control transactions needed to update the Mask or Pulse Enable patterns stored in the pixel configuration latches. This is especially true for mask patterns with a small number of pixels to be masked over the full matrix. Section 3.2.4 gives a shorter description of the addressing scheme detailed in the following.

For write and read accesses to the peripheral Pixel Control Registers, the address of the control transaction is internally treated as composed of sub-fields. Figure 3.16 illustrates the sub-fields of the chip address of a control transaction targeting the peripheral registers related to the configuration of the matrix.

Configuration Address Bus

Region Selector Field <4:0>	3'b100	Region Broad-cast	Not Used	Not Used	Toggle Bit	Pulse <15:0> Select	Row <15:0> Select	Column <31:15> Select	Column <15:0> Select
Bit <15:11>	Bit <10:8>	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figure 3.16: Illustration of address bus bits used to access pixel configuration registers

- Region Selector Field (bits <15:11>) – Determines the region containing the addressed peripheral Pixel Control Registers. This is ignored in write transactions when the Region Broadcast bit (bit 7) is asserted.
- Base address (bits <10:8>) – Constant value 3'b100, indicating that an access to Pixel Control registers is taking place.
- Region Broadcast bit (bit 7) – Overrides the Region Selector field in write transactions. If asserted, the selected Pixel Configuration registers of all regions are written simultaneously.
- Toggle bit (bit 4) – If asserted the selected registers are bit-wise toggled, i.e. they are loaded with the binary ones' complement of the currently stored value. The 16 bit data word of the control transaction is ignored for the Column Selection and Row Selection registers. **Note: the toggle functionality applies only to the Column Selection and Row Selection registers. It does not work for the Pulse Gating registers.**
- Pulse Gating Selector (bit 3) – Indicates that bits <15:0> of the Pulse Gating Slice register for the selected region are addressed.
- Row Selection Slice selector (bit2) – Indicates that bits <15:0> of the Row Selection Slice register are addressed.
- Column Selection Slice 2 selector (bit 1)– Indicates that bits <31:16> of the Column Select register are addressed.
- Column Selection Slice 1 selector (bit 0) – Indicates that bits <15:0> of the Column Select register are addressed.

Read back of the peripheral Pixel Configuration registers is possible. The bit map scheme previously described is still utilized to select the desired register to be read. The same base address field is to be used for address bits 10:8 (3'b100). However the broadcasting options of

the addressing scheme are obviously not working for read transaction. The Region Selector Field must indicate the region containing the addressed register within the periphery and bits 3:0 must be one-hot to specify only one of the Pixel Configuration registers in the region for the read back operation.

Figure 3.17 illustrates an example of the address and data fields of a single control write transaction that can be utilized to initialize all of the Column Selection, Row selection and Pulse Gating registers at once, writing a binary zero to all their bits. In this example, the Region Broadcast function is used thus addressing all the regions. The region selector field is not used by the logic in this case. The toggle bit is de-asserted, so all selected 16–bit Pixel Configuration registers are assigned the value of the transaction Data field (all zeros in the example).

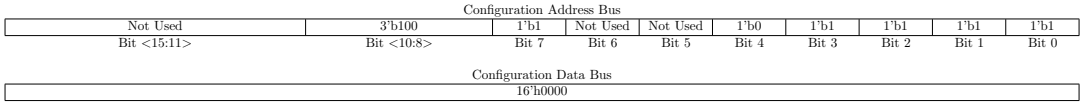

Figure 3.17: Address and Data fields of a sample chip control transaction to initialize all Pulse Propagation, Row and Column Select lines to binary 0

3.6.3 Recommended masking and pulsing configuration procedures

The in-pixel latches do not have a direct reset signal and the value after powering the chip is unknown. **It is recommended that all PULSE_EN and MASK_EN latches are properly configured in the chip's initial configuration phase**.

It is also recommended that **all column and row select bits are cleared in-between setting of patterns**. Recommended sequences of transactions for common configuration tasks are given as examples below.

Loading Masking Pattern

- Clear PIXCNFG_DATA bit and clear PIXCNFG_REGSEL bit (select driving of Mask register row selection lines)
- Set all columns and rows selection bits thus clearing (unmasking) all the pixels
- Clear all columns and rows selection bits
- Set PIXCNFG_DATA bit
- Looping over all rows
  - Set the column selection bits of pixels to mask in the given row
  - Set the given row selection bit thus *setting* the mask bit of the selected pixels
  - Clear all columns and rows selection bits

Refreshing Masking Pattern (without temporary clearing)

- Clear PIXCNFG_DATA bit and clear PIXCNFG_REGSEL bit (select driving of Mask register row selection lines)
- Set all column select bit
- Set the row selection bits for all rows *without* pixels to be masked, thus clearing (unmasking) all the pixels in all the rows without pixels to be masked
- Clear all row and column selection bits
- Looping over the rows *with* pixels to be masked

- Set the column selection bits corresponding to the pixels that do not need to be masked in the given row
- Set the corresponding row selection bit thus *clearing* the mask bit of the selected pixels in the row
- Clear the row selection bit
- *Toggle* the column selection bits
- Set PIXCNFG_DATA bit
- Set the corresponding row selection bit thus *setting* the mask bit of the selected pixels in the row
- Clear all row and column selection bits
- Clear PIXCNFG_DATA bit

Note: the sequence above is different than the previous one. This one refreshes the mask bit of all pixels, those that are supposed to be masked and those that are supposed to remain unmasked. Also, it does so avoiding clearing of the currently stored mask pattern, contrary to the previous one. It is a possible sequence that can be used for a periodic refreshing of the full mask pattern also in the background of data taking operations, minimizing the number of transactions and of the possible side effects on data frames. This can be useful to refresh a mask pattern that gets progressively altered due to the accumulation of SEUs of the in-pixel latches.

Setting a Pulsing pattern and masking all remaining pixels

- Clear PIXCNFG_DATA bit and select Masking
- Set all row and column select bit thus writing '0' to all MASK_EN latches (this *clears* the full mask pattern)
- Clear all row and column select bits
- Set PIXCNFG_DATA bit
- On a row by row basis:
  - Set COLSEL bits of pixels to be masked in a given row
  - Set ROWSEL bit thus writing '1' to selected MASK_EN latches
  - Clear ROWSEL bit
  - Toggle COLSEL bits thus selecting the pattern to pulse
  - Select PULSING
  - Set ROWSEL bit thus writing '1' to selected PULSE_EN latches
  - Clear all row and column select bits
  - Select MASKING

3.6.4 Notes on Signal Skewing in the context of Pixel Configuration Global Signals

The ALPIDE chip supports the skewing of the assertion of signals to the matrix to avoid surges of current with the consequent voltage drops on the on-chip power distribution mesh. This functionality is enabled via the *Enable Skewing of Global Signals* field in the Mode Control Register. When skewing is enabled, the update of the slice registers in the regions is executed over 16 clock cycles. The update of the registers in regions 15 and 16 (middle ones) happens in the first cycle, the update of regions 14 and 17 in the second cycle and so on up to regions 0 and 31 whose registers get updated 16 cycles later. Additionally, the propagation of

the PIXCNFG_REGSEL and PIXCNFG_DATA to the regions is skewed in the same fashion. These skewed updates of the peripheral registers reflect in the driving of the select wires to column and rows and of the global data signal.

The usage of the skewing functionality together with advanced addressing features needs caution. It might lead to unexpected results if the user does not ensure that the column or row selection lines are all de-asserted in-between transactions, so that all pixel latches are disabled. For example the PIXCNFG_REGSEL and PIXCNFG_DATA are contained within one register and thus can be modified with a single control transaction. Flipping both bits in conjunction with skewing and without de-asserting all column and row select wires beforehand can lead to unexpected Mask and Pulse Enable patterns.

3.7 Analog test pulse injection and pixel digital testing

The pixels of the ALPIDE chip contain built-in testing features. They can be used to force the generation of a hit using a test charge injection capacitor (*analog pulsing*) or directly setting the pixel state register (*digital pulsing*). Details on the pixel digital section including pulsing are given in 4.1.2.

The pulsing functionalities of each pixel are enabled with its Pulse Enable (PULSE_EN) register. Once the pulsing has been enabled in the desired pixels, two pulsing signals (APULSE and DPULSE) are used to trigger the injection of the charge in the front-end or to set the state register. The APULSE and DPULSE signals are distributed globally to all pixels from the periphery. Their fan-out trees have individual buffers for each of the double-columns. The APULSE and DPULSE signals propagate vertically along the double-columns, feeding all pixels of each double column. They are also re-generated by buffers placed at regular distances along the double-column.

The edges of the APULSE signal cause the injection in the front-end input node of current pulses. The polarity of the pulse depends on the direction of the edge (rising or falling). The rising edge simulates the effect of release of charge in the collection diode by a particle hitting the pixel. The total charge of the current pulse is controllable by dedicated DACs. The DPULSE signal can be used to set the pixel state latches that constitute the Multi-Event buffer.

Both pulsing operations require that the PULSE_EN latches of the target pixels are set and that one of the three global STROBE signals is asserted, in order to store the hit in the corresponding MEB location. The masking of the pixels has priority over the pulsing, therefore a masked pixel does not produce a hit when it is pulsed.

3.7.1 Generation and timing of pulsing signals

This section describes the generation of the global APULSE and DPULSE signals and the associated features. The APULSE or DPULSE signals are asserted by the FROMU following the reception of a PULSE command (see Table 3.1 on page 33 and Table 3.3 on page 37). The user needs to select if the PULSE command shall activate either the APULSE signal or the DPULSE signal. The selection is made via the *Test Pulse Mode* bit of **FROMU Configuration Register 1 (0x0004)** (Table 3.5 on page 38). If it is set to '0', a PULSE command will generate a pulse on the DPULSE nets, if it is set to '1', the APULSE nets will be activated.

The duration of the APULSE or DPULSE signals is programmable with a granularity of 25 ns using the **FROMU Pulsing Register 2**. The maximum duration that can be set is 1638.4 μs (16 bits register).

The delay between the assertion of the first bit of the PULSE command on the control interface (PAD_DCTRL_I) and the assertion of the DPULSE at the input of the matrix is of 19 clock cycles (see Figure 3.18). The delay is identical in the case of an APULSE. Ten clock cycles of the total delay are needed for the transmission of the PULSE command on the control line.

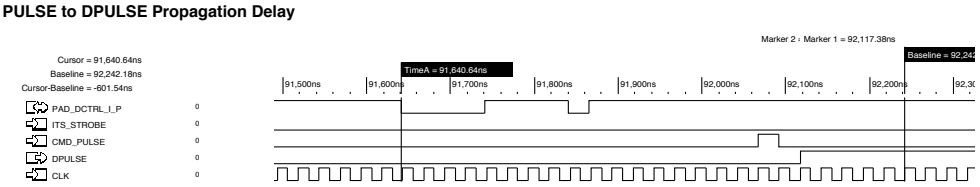

Figure 3.18: Delay between the assertion of the first bit of the PULSE command and the assertion of the DPULSE as seen by the Matrix.

3.7.2 Automatic assertion of an internally generated STROBE following a PULSE command

It is possible to configure the circuit to generate automatically an internal TRIGGER command following a PULSE command (be it a DPULSE or an APULSE). This functionality is enabled via the setting of the *Enable Test STROBE* bit in the **FROMU Configuration Register 1 (0x0004)**.

The delay between the *positive edge* of the internal PULSE signal and the assertion of the automatically generated trigger is controlled via the **FROMU Pulsing Register 1 (0x0007)**. The value of the register incremented by one gives the delay in units of clock cycles. The duration of the assertion of the STROBE signal is still controllable with the **FROMU Configuration Register 2 (0x0005)**.

3.7.3 Fan-out of pulse signals to the Matrix

The fan-out of the DPULSE/APULSE signals to the double columns is implemented in the region modules logic as illustrated in Fig. 3.19. Each region drives the DPULSE and APULSE inputs of 16 double columns, propagating the DPULSE or APULSE signals generated by the FROMU following a Pulse command.

A 16 bit Pulse Gating register (Fig. 3.19 and Fig. 3.15) controls the propagation of the DPULSE or APULSE signals to each double column. The propagation of a DPULSE or APULSE to a double column is enabled if the corresponding bit of the Pulse Gating register is set. The control of the fan-out of the APULSE/DPULSE signals aims at minimizing the current pulses needed to charge and discharge the large capacitance of the global APULSE and DPULSE nets, resulting from the sum of wiring capacitance and total gate-capacitance at the inputs of the pixels.

Additionally, the Pulse Gating registers are connected in a shift-and-rotate topology. The pulse gating pattern stored in the Pulse Gating registers can automatically shift and rotate after the de-assertion of the APULSE/DPULSE signal received from the FROMU. This functionality is optionally enabled using the *Enable Rotate Pulse Lines* bit of **FROMU Configuration Register 1 (0x0004)**. The automatic rotation of the pulse gating pattern is intended to facilitate testing by reducing the number of control transactions needed to change the set of pixels that get pulsed.

The Pulse Gating registers are chained in two groups (region 0 to 15 and regions 16 to 31). The shift output of the register in region n connects to the shift input of the register in region $n+1$. The shift output of the 16-th register of the group is connected back to the shift input of the first register of the group. Thus, the full set of 512 bits of the Pulse Gating registers are combined into two shift and rotate registers of 256 bits. This is illustrated in Figure 3.20. The shift is triggered on the de-assertion of the DPULSE/APULSE input.

The content of the Pulse Gating registers can be read back through the scheme described in Section 3.6. The content of the registers after the automatic shift and rotate can therefore be directly inspected and the pulsing pattern always retrieved.

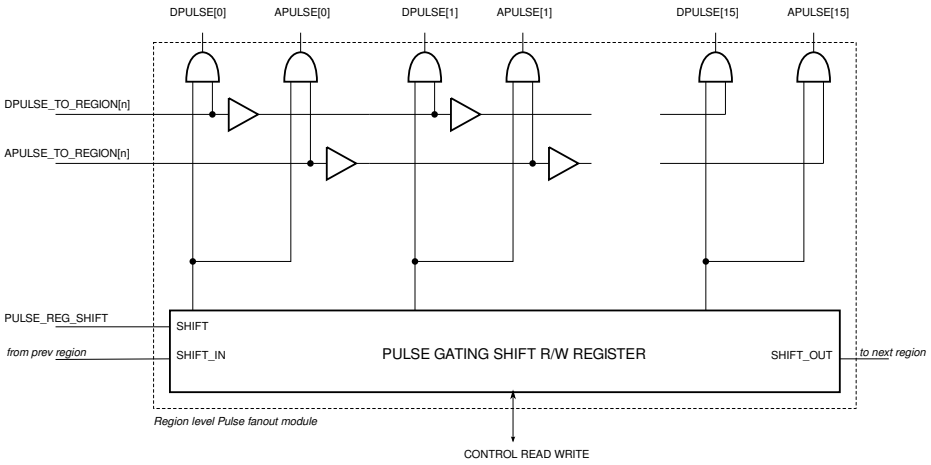

Figure 3.19: Fan-out of the Pulsing signals to the Double Columns. Schematic diagram of the circuits in one of the region modules in the periphery, forwarding APULSE and DPULSE to 16 double columns.

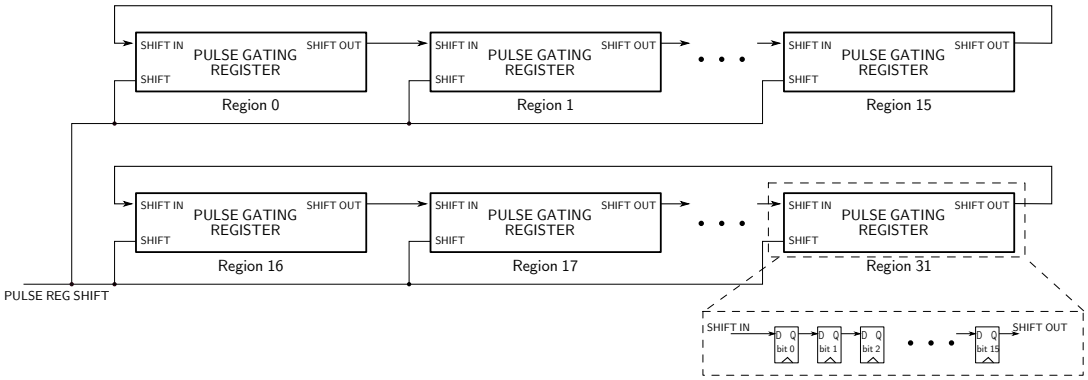

Figure 3.20: Pulse shift registers scheme where the rotation is triggered on the negative edge of a Test Pulse

3.8 Chip initialization

This section outlines sample procedures to initialise the chip and prepare it for data taking. It is assumed that the control interface(s) are operational and well-configured. It is recommended that an initial **global reset** is applied to the chip in order to load all configuration registers with recommended default values.

3.8.1 Configuration of in-pixel logic

Even if the user does not intend to use the Test Pulse functionality, it is recommended that the PULSE_EN and MASK_EN latches in every pixel are initialised. This is done via the procedure outlined in Section 3.6.

3.8.2 Configuration and start-up of the Data Transmission Unit

The following procedure configures and starts up the high speed data transmission port. This is to be executed for Inner Barrel Chips or Outer Barrel Master chips in the ITS Upgrade application.

Writes to **DTU Configuration Register** and **DTU DACs Register**:

1. Set the PLL VCO stages and bandwidth configuration fields to desired values. Nominal design value are 4 stages and narrow bandwidth and should provide adequate behavior. However tuning of these settings might be needed depending on the incoming clock jitter and operating conditions.
2. Set the *PLL charge pump current* control to valid value. The nominal design value of 4'b1000 is intended to provide adequate behavior.
3. Set the *Line Driver current* and *Pre-emphasis Driver* current controls to values optimized for the line characteristics.
4. Clear *PLL off signal* bit to **start the PLL up**.
5. Force a **reset of the PLL**. This is done by **first setting and then clearing the PLL reset bit** with two subsequent transactions, leaving all the other bits of the DTU Configuration Register unchanged.

3.8.3 Setting up of readout

Setting CMU and DMU Configuration Register in the Outer Barrel scenario

This register should be written to individually for every register in an OB.

- Set the *Previous Chip ID* field as required. Every chip monitors the parallel bus. Upon detecting in a CHIP HEADER the ID value programmed, the chip waits for the previous one to transmit the CHIP TRAILER word and thus yield the token.
- Set the *Initial Token* as required to specify the chip in question is the first in the readout chain. Care should be taken to ensure that only one chip in an module has this bit set to '1'.

NOTE: the setting of Initial Token is not committed to the logic until a RORST command is executed. It is recommended to send this command at the end of the configuration sequence, as a last action immediately before sending triggers.

Setting CMU and DMU Configuration Register in the Inner Barrel scenario

- The *Previous Chip ID* field is redundant - can be set to the default value of 4'b1111. Note that is a sanity check as 4'b1111 is not a legal CHIP ID.
- The *Initial Token* field is redundant - can be set to '0'.

Setting FROMU Configuration Registers and enabling readout mode

Items outlined below apply to both IB and OB configuration. It is recommended that a MULTICAST to target all chips in a module is used to ensure consistency.

1. Write to **FROMU Configuration Register 1**
   - Set *Pixel MEB Mask* field as required. Mask LSB to mask Matrix Memory Slice 0.
   - Set *Internal STROBE generation* field if repetitive auto-generate STROBEs in CONTINUOUS mode is a desired. An initial TRIGGER will then be required to start the strobing. Following that any further TRIGGERS will re–time the stream.
   - RECOMMENDED : Set *Enable BUSY Monitoring* to '1'.
   - Set *Test Pulse Mode* field as required - '0' for DPULSE and '1' for APULSE.
   - Set *Enable Test STROBE* field as required - '1' to auto-generate STROBE windows following a PULSE. Note that if this feature is to be used the user must also set the **FROMU Pulsing Register 1** and **FROMU Pulsing Register 2**. The duration of the STROBE window is set by the **FROMU Configuration Register 2**.
   - Leave remaining bits as '0' - they are either Not used or read-only.
2. Write to **FROMU Configuration Register 2**
   - Set STROBE duration as required (25 ns step).
3. If PULSE is to be used the user should also program the **FROMU Pulsing Register 1** and **FROMU Pulsing Register 2**.
4. Write to **Periphery Control Register**. This should be the last register write before TRIGGER/PULSE commands start being sent as it would put the chip in a readout mode:
   - Set *Chip Mode selector* field as required - 2'b01 for TRIGGERED and 2'b10 for CONTINUOUS.
   - Set *Clustering enable* field as required - '1' is the recommended value.
   - Set *Matrix Readout Speed* as required - '1' is the default (20 kHz). Setting '0' would imply readout at 10 kHz.
   - Set *IB Serial Link Speed* as required - '0' sets transmission at 400 Mbps, '1' at 600 Mbps, and '2' and '3' at 1200 Mbps, which is also the default.
   - Note that it is recommended that *Force Busy*, *Force Busy Value* and *Xoff* fields would typically be modified while in readout so the values for the remaining fields should be stored.
5. Send a **RORST Readout Reset command**. Ideally this should be sent as a broadcast command to all the chips on the control link. This command is mandatory to commit the Initial Token bit and activates the driving of the local bus by the chip with the token.

Use of the Load Outer Barrel Default Configuration command (LOADOBDEFCFG)

The ALPIDE chip includes a specific command to facilitate the set up procedure in OB mode. Upon issue, the LOADOBDEFCFG (Load Outer Barrel Default Configuration) command will set the CMU and DMU Configuration registers of the chip with valid default values for the correct operation of a OB Master and six active slave chips. This command is intended to be issued as a MULTICAST.

In this automated write, the MASTER chip has the *initial_token* field set while all the SLAVEs have it cleared. Additionally, a sequential readout chain is implemented via the

prev_chip_id fields such that the master is first, the SLAVE with CHIPID 7'bxxx_x001 second, and the SLAVE with CHIPID 7'xxx_x110 last. The *DisableManchester* and *EnableDDR* fields within the same register get assigned their default values.

Finally, following the writing of the register, an automated RORST is issued in order to commit the new configuration to logic.

3.9 Data Transmission Unit and Test Logic

The high speed serial data interface of the chip is implemented by the Data Transmission Unit (DTU), a full custom block containing a PLL, a Serializer and a differential line driver with pre-emphasis. The digital periphery contains a Data Transmission Unit Logic (DTU Logic) module implementing functionalities closely related to the serial data port and the DTU. Figure 3.21 illustrates the interfaces between these functional blocks and other modules in the chip, as well as a simplified representation of their internal structure. The DTU and the DTU Logic have user programmable configuration and status flags that are accessible through the control registers **0x0014** to **0x001A** (refer to section 3.2.2, table 3.2 and subsequent descriptions).

The control inputs of the PLL are accessible by **DTU Configuration Register (0x0014)**, bits 3:0 and 8. The current of the charge pump circuit of the PLL can be controlled with **DTU DACs Register (0x0015)**, bits 3:0. The current of the main line driver of the DTU can be programmed using **DTU DACs Register (0x0015)**, bits 7:4. The current of the pre-emphasis line driver of the DTU can be programmed using **DTU DACs Register (0x0015)**, bits 11:8.

The DTU Logic module is located between the module scheduling the transmission of Chip Data Frames (DMU) and the DTU block. The main functionalities provided by the DTU Logic module are:

1. 8b/10b encoding of the data produced by the Data Management Unit and to be transmitted by the DTU
2. Programming of the serial port line rate (1200 Mb/s, 600 Mb/s, 400 Mb/s) according to operating mode and configuration
3. Monitoring of the PLL lock status and re-synchronization of the serializer
4. Test features for the Data Transmission Unit.

If the chip is configured to operate as an Inner Chip, its CHIPID input being a valid Inner Chip identifier, the serial output bit rate can be programmed. The available options are 1200 Mb/s (default), 600 Mb/s and 400 Mb/s selectable using the **Mode Control Register (0x0001)**, bits 5 and 4. When the chip is configured to operate as an Outer Barrel Master chip, through a proper CHIPID identifier, the output line rate is 400 Mb/s (provided test modes are not enabled). When the chip is configured as an Outer Barrel Slave the Data Transmission Unit is kept in a low-power off state and no output is generated.

The Data Transmission Unit and its Serializer are agnostic of the programmed output line rate. They always operate in the same fashion, shifting out a 30 bits data word loaded into the serializer at every cycle of the main digital clock (40 MHz). It is the DTU LOGIC that generates the 30 bits of the DTU DATA bus to obtain the desired output rate, effectively generating slower bit serial stream at the DTU output. For the operation at 1200 Mb/s, all bits of the DTU DATA bus can be different. When transmitting at 600 Mb/s, every bit is replicated and transmitted twice, or equivalently for two consecutive bit Unit Intervals. Finally, every bit is replicated three times when operating at 400 Mb/s.

The input data to the DTU Logic are provided on three input data buses with a width of one byte (DIN0, DIN1, DIN2). There is also a valid data qualifier flag and data or control (k)

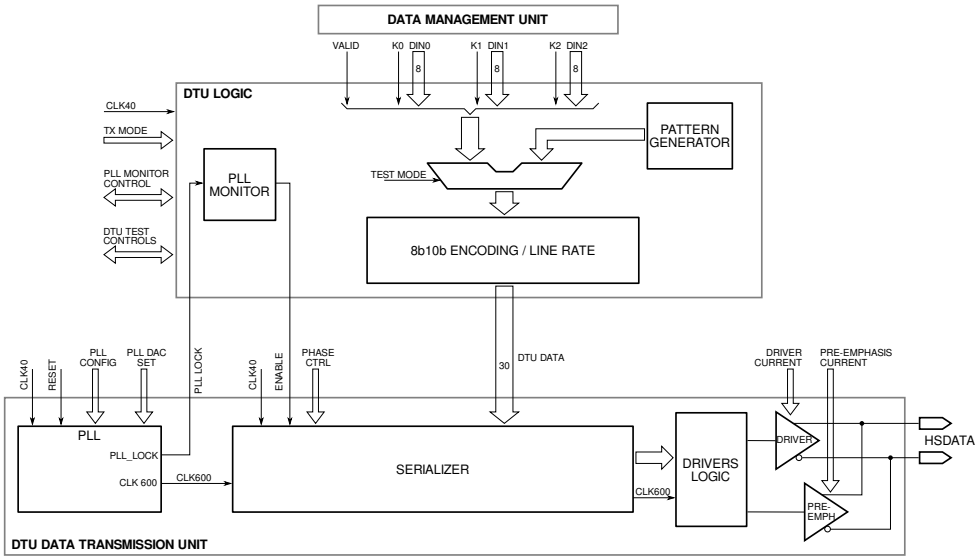

Figure 3.21: Functional block diagram of the Data Transmission Unit and the related digital module DTU LOGIC.

selectors. The DMU is supposed to request the transmission of K28.5 words if there are no data to be transmitted.

The 8b/10b encoder operates on the convention of transmitting the encoded version of the DIN0 byte first, then DIN1, then DIN2. When selecting the 600 Mb/s and 400 Mb/s rates for an Inner Chip, the Data Management Unit validates the data on the input bus once every two cycles or once every three cycles. In Outer Barrel Master mode (400 Mb/s output rate) only inputs DIN0 and K0 are used and inputs DIN1, K1, DIN2, K2 are ignored.

The DTU LOGIC contains a PLL MONITOR State Machine (SM) intended to filter the LOCK flag of the PLL during locking and un-locking transients. During the locking phase or as a consequence of noise the PLL LOCK flag could transiently toggle. These transients of the LOCK flag do not necessarily imply that the quality of the fast clock is insufficient for reliable transmission. If the PLL LOCK flag is kept asserted over a sufficiently long interval, the SM transitions into its LOCKED state. Similarly, if the LOCK flag is de-asserted for sufficient time, it enters the UNLOCKED state.

The PLL LOCK output flag is directly readable (**DTU PLL Lock Register 1, 0x0016**, bit 8). A flag indicating that the SM is in the LOCKED state is accessible by **DTU PLL Lock Register 1 (0x0016)**, bit 9. Transitions of the PLL MONITOR State Machine from the UNLOCKED state into the LOCKED state are counted. The counter value is readable at **DTU PLL Lock Register 1 (0x0016)**, bits 7:0. The counter is reset by the PLL reset command (**DTU Configuration Register, 0x0014**, bit 8). The minimum duration of the assertion of the flag to bring the SM to the LOCKED state is programmed by **DTU PLL Lock Register 2 (0x0017)**, bits 7:0. Similarly, the minimum duration of the de-assertion of the flag to bring the SM to the UNLOCKED state is programmed by **DTU PLL Lock Register 2 (0x0017)**, bits 15:8.

In addition to the monitoring of the PLL LOCK flag, the PLL MONITOR drives the ENABLE signal of the SERIALIZER. The ENABLE of the Serializer is de-asserted when the SM declares that the PLL is not locked and asserted when the SM declares a PLL LOCKED condition. The assertion of the ENABLE input of the Serializer initializes an internal synchronizer that generates the timing signal for the sampling of the DTU DATA from the main clock domain to the fast clock domain of the Serializer. The timing of the parallel load pulse with respect to the main clock cycle can be programmed using **DTU Configuration Register (0x0014)**, bits 7:4. The ENABLE signal to the Serializer can be over-driven using **DTU Test Register**

3 (0x001A), bits 9 and 8. These features allow a fully automatic or alternative fully *manual* control of the re-initialization of the synchronizer in the event of a loss of lock of the PLL followed by re-locking. If the Serializer is maintained disabled, the DTU DATA bus is ignored, there is no parallel load in the internal shift registers and these maintain constant values. The output of the DTU in this condition is a replica of the PLL clock output forwarded through buffers in the Serializer and a multiplexer in the Drivers Logic module.

The DTU LOGIC module contains circuits for the testing of the serial link. The activation of the Test Mode effectively detaches the DTU LOGIC from the input data bus. Constant arbitrary patterns and pseudo-random bit patterns are possible options.

The DTU Test mode is enabled with **DTU Test Register 1 (0x0018)**, bit 0. Bit 1 of the same registers selects between constant bit patterns or an internal pseudo-random pattern.

When constant patterns are used, the K0, K1, K2, DIN0, DIN1, DIN2 inputs of the Encoder are driven with the values stored in the fields of **DTU Test Register 1 (0x0018)**, **DTU Test Register 2 (0x0019)**, **DTU Test Register 3 (0x001A)**.

The 8b/10b encoding can be disabled setting bit 5 of **DTU Test Register 1 (0x0018)**. In this configuration, the K0, K1, K2 bits have no function, while six additional bits are padded to DIN0, DIN1, DIN2 to produce the 30 bit constant pattern sent to the Serializer. The padding bits are programmed with register **DTU Test Register 1 (0x0018)**, bits 11:6.

When the internal pseudo-random pattern is enabled, the input bus to the Encoding module is driven with outputs provided by an internal pseudo-random bit stream generator (PRBS-7). It is recommended to disable the 8b/10b encoding when using the internal pattern generator. It is possible to program the output bit rate on the serial output using **DTU Test Register 1 (0x0018)**, bits 4:3. The same bits can be used to reset the PRBS logic ensuring that the internal shift register is loaded with a valid pattern (all ones).

The supported DTU testing configurations are summarized in the following list. **It is recommended to configure bits 5 and 4 of Mode Control Register (0x0001) to their default value, corresponding to 1200 Mb/s, when using any of the supported DTU Test Modes described above.**

1. Constant 30 bit pattern bypassing encoding at 1200 Mb/s. Cyclic transmission of a programmable, arbitrary 30 bit pattern. This can be used to generate square waves of programmable frequencies amongst many other arbitrary repeating patterns. This configuration is obtained setting *high* bits 0, 5 of **DTU Test Register 1 (0x0018)** and setting *low* bits 1, 2. The 30 bit pattern is programmed with fields DIN0, DIN1, DIN2, BDIN8b100, BDIN8b101, BDIN8b102 of the DTU Test registers.

2. Constant 10 bit pattern bypassing encoding at 400 Mb/s. Cyclic transmission of a programmable, arbitrary 10 bit pattern using 400 Mb/s line rate. This uses the same configuration of the Outer Barrel Master Chip. This configuration is obtained setting *high* bits 0, 2, 5 of **DTU Test Register 1 (0x0018)** and setting *low* bit 1. The 10 bit pattern is programmed with fields DIN0, BDIN8b100 of the DTU Test registers.

3. Pseudo-random bit pattern (PRBS-7) bypassing encoding at 1200 Mb/s, 600 Mb/s or 400 Mb/s. This configuration is obtained setting *high* bits 0, 1, 5 of **DTU Test Register 1 (0x0018)** and setting *low* bit 2. The lane rate is controlled with bits 4:3 of the same register.

4. Constant bit pattern with 8b/10b encoding at 1200 Mb/s. Cyclic transmission of three programmable data or control words using 1200 Mb/s line rate. This mode uses the same configuration of the Inner Barrel Chip, including the encoding logic. This configuration is obtained setting *high* bit 0 of **DTU Test Register 1 (0x0018)** and setting *low* bits 1, 2, 5. The data or control words are programmed with fields DIN0, DIN1, DIN2, K0, K1, K2 of the DTU Test registers.

bit	Field Name	Description	Access
1:0	Mode	operation mode (calibration, automatic, manual, full-manual)	R/W
5:2	ADC_INPUT_SELECTION	ADC input selection, see 3.25	R/W
7:6	SetIcomp	Comparator supply current	R/W
8	DiscriSign	Comparator Sign, see 3.10.4	R/W
10:9	RampSpd	Speed of the generated Ramp	R/W
11	HalfLSBTrim	Half Bit Fine Tuning, see 3.10.4	R/W
14:12	Not used	Not Used	R/W
15	COMP_OUT	Output of the comparator	R

Table 3.39: ADC Control Register.

5. Constant bit pattern with 8b/10b encoding at 400 Mb/s. Cyclic transmission of a single, programmable data or control word using 400 Mb/s line rate. This mode uses the same configuration of the Outer Barrel Master Chip including the encoding logic. This configuration is obtained setting *high* bits 0, 2 of **DTU Test Register 1 (0x0018)** and setting *low* bits 1, 5. The data or control words are programmed with fields DIN0, K0, of the DTU Test registers.

3.10 Operation of the ADC

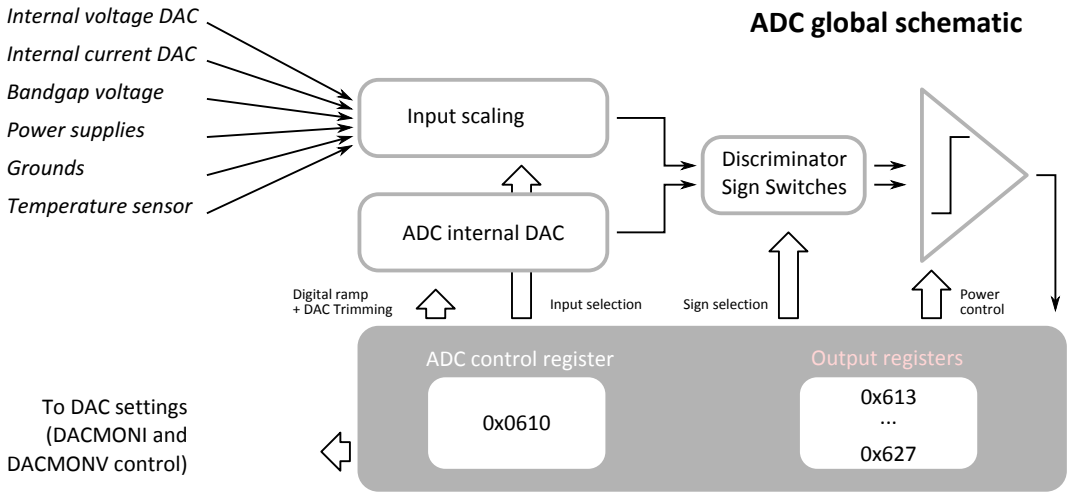

Figure 3.22: ADC schematic

The ALPIDE includes a 10 bits resolution ADC to monitor quasi-static internal signals:

- Analog Supply Voltage and Ground
- Digital Supply Voltage and Ground
- DACMONV - Output of the Voltage DAC, see 4.3
- DACMONI - Output of the current DAC, see 4.3
- Internal Temperature Sensor
- BANDGAP voltage

The digital conversion is done by comparing input signals with a temperature independent / power supply independent ramp voltage generated by an ADC-internal DAC. The different possible inputs are automatically scaled to fit the dynamic range of the ramp. The ramp generation, the ADC settings and the ADC measured values are controlled by a dedicated ADC Control Register. (address 0x0610, see Fig. 3.22)

Four measurement modes are available through the ADC Control Register (Table 3.39) :

	Manual Mode=0	Calibration Mode=1	Automatic Mode=2	Super-Manual Mode=3
ADC_INPUT_SELECTION	Used	Used		Used
SetIcomp	Used	Used	Used	Used
DiscriSign	Used	Used	Used	Used
RampSpd	Used	Used	Used	Used
HalfLSBTrim	Used	Used	Used	Used
COMP_OUT				Used

Table 3.40: ADC Operating Modes.

SetIcomp	Comparator Current
0	163 μA
1	190 μA
2	296 μA (nominal)
3	410 μA

Table 3.41: Comparator Current (SetIComp).

- Automatic - Measurement of all inputs connected to the ADC . See 3.10.5.
- Manual - Measurement of a single input connected to the ADC, according to input selection registers. See 3.10.3.
- Calibration mode : Measurement of a "know" input for the offset calibration of the ADC.
- Super-Manual - The whole measurement procedure can be done step by step, through register settings. No ramp is generated in this mode (this mode is not detailed in the following section).

Before operating the ADC in manual or automatic mode, a calibration procedure shall be performed. This calibration procedure will give the optimal parameters for the ADC to minimize systematic errors (3.10.4).

3.10.1 General settings

Depending of the chosen mode, Table 3.40 details the use of the fields of the ADC Control Register.

General settings for all modes Both SetIComp, and RampSpd parameters allow to fine tune the ADC parameters to optimize the results against the comparator current (power consumption) and the ramp speed, see Tables 3.41 and 3.42.

The default values of both *comparator current* and *ramp speed* have been chosen to fit ALPIDE ITS/MFT nominal operation.

DiscriSign and HalfLSBTrim bits will be detailed in 3.10.4. Their values are extracted from the calibration procedure.

RampSpd	Time per Step	Time for a Full Scale Input
0	500 ns/step	$\leq$ 600 μs/channel
1	1 μs/step (nominal)	$\leq$ 1.1 ms/channel
2	2 μs/step	$\leq$ 2.2 ms/channel
3	4 μs/step	$\leq$ 4.3 ms/channel

Table 3.42: Ramp Speed (RampSpd).

ADC_INPUT_SELECTION According to 3.25, the ADC_INPUT_SELECTION field is used to select the input of the ADC for both calibration, manual, and super-manual modes. It will be detailed in 3.10.3.

COMP_OUT The read-only *COMP_OUT* field of the *ADC Control Register* together with the *ADC DAC input VALUE* register (address 0x0611, see Table 3.23) are used only in super-manual mode to externally control the DAC value (and manually generate the ramp), knowing the comparator output.

3.10.2 ADC Outputs

The measurements performed by the ADC are saved internally in a local memory. The registers located at addresses from 0x613 to 0x627 (Table 3.23) store the hexadecimal value of the last measurement executed. Each register address corresponds to a dedicated input.

To alleviate internal offset errors, all these values shall be corrected by the value of the calibration register (address 0x0612), by subtracting the calibration value.

Voltage measure The voltage measures *VCASN*, *VCASP*, *VPULSEH*, *VPULSEL*, *VRESETP*, *VRESETD*, *VCASN2*, *VCLIP* and *VTEMP* have a scaling factor described by eq 3.1.

$$Value_{read} = \frac{V_{mV}}{2 \cdot 1.068} + offset \tag{3.1}$$

Current measure The current measures *IRESET*, *IAUX2*, *IBIAS*, *IDB* and *ITHR* have a scaling factor described by eq 3.2.

$$Value_{read} = \frac{5 \cdot I_{\mu A}}{1.068} + offset \tag{3.2}$$

Temperature measure The temperature measure has a scaling factor described by eq 3.3.

$$Value_{read} = \frac{Temp_{degCelsius} + 51.5}{0.147} + offset \tag{3.3}$$

3.10.3 Manual Measurement

The manual measurement procedure is used to sample one ADC input at a time. The procedure detailed below consists in setting the input via the ADC Control Register and send an ADCMEASURE pulse. After a time $\leq$ 5ms (see Table 3.42), the measured value can be read in the *AVSS* register (address 0x0613), whatever the settings of *ADC INPUT SELECTION*, *VOLTAGE DAC SELECTION* and *CURRENT DAC SELECTION*.

Procedure

1. Setup the ADC Control Register
   - Set the manual mode, i.e set *ADC_MODE=2*
   - Set ADC input according to Table 3.31. For example for *AVSS*, set *ADC_INPUT_SELECTION=0*
     - if *ADC_INPUT_SELECTION* is on *DACMONV* (i.e value is 5), set Voltage DAC input according to Table 3.25. For example for *VCASP*, set *VOLTAGE_DAC_SELECTION=1*

– if *ADC_INPUT_SELECTION* is on *DACMONI* (i.e value is 6), set Current DAC input according to Table 3.25. For example for *IBIAS*, set *CURRENT_DAC_SELECTION*= **Beware, the *IREF* input shall never be used**

- Keep the *comparator current* field value used at calibration step
- Keep the *comparator output sign* field value used at calibration step
- Keep the *ramp speed* field value used at calibration step
- Keep the *half bit trimmer* field value used at calibraton step

2. Send a *START_MEASURE* pulse, command *ADCMEASURE* (0xFF20 , see Table 3.3)
3. After a time $\leq$ 5ms (see Table 3.42) read the *AVSS* register (address 0x0613), whatever the settings of *ADC INPUT SELECTION*, *VOLTAGE DAC SELECTION* and *CURRENT DAC SELECTION*.

3.10.4 Calibration procedure

The calibration procedure includes 3 steps:

1. Find the comparator sign to minimize the systematic measurement error
2. Find half bit trimmer to optimize ADC accuracy
3. Store the measure of a known signal as a reference into a dedicated calibration register (address 0x0612, see Table 3.23).

The first part of the calibration is obtained by selecting the right set of [DiscriSign, HalfLSBTrimmer] parameters for which the comparator threshold is the closest to, and just above the analog ground AVSS.

The second part, i.e. the measure of a known voltage, is intended to alleviate the unknown offset of the measurement.

Procedure The value of both *comparator current* and *ramp speed* shall be chosen to fit the application. The default values shall be fine for most cases.

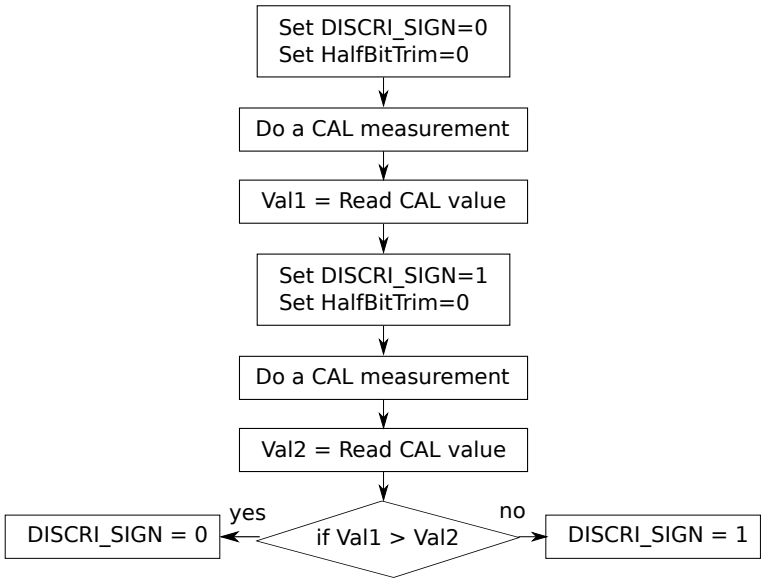

Figure 3.23: Calibration 1 : Discriminator sign

Figure 3.23, shows first part of the calibration procedure to find out the right value for the *comparator output sign*, named DISCRI_SIGN here. For this step, select *ADC_INPUT_SELECTION=0* to select *AVSS*.

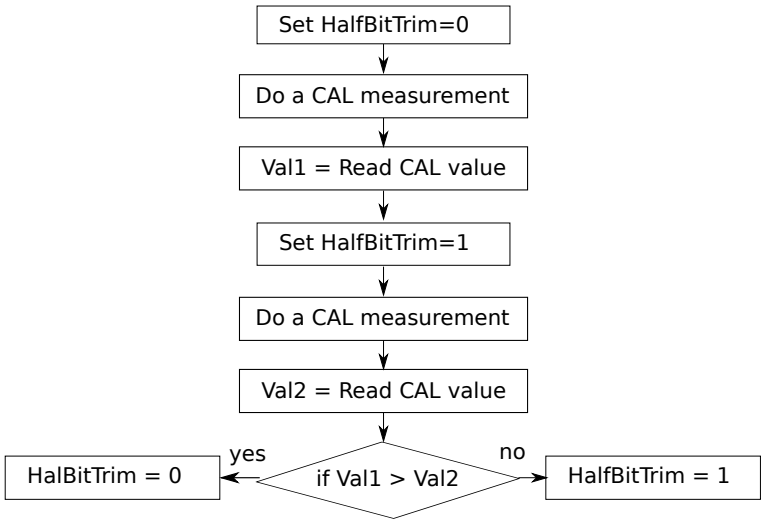

Figure 3.24: Calibration 2 : Half lsb trimming bit

Figure 3.24 shows the second part of the calibration procedure to find out the *half bit trimmer* value that minimizes the quantization error of the ADC. For this step, select *ADC_INPUT_SELECTION=7* to select *BANDGAP*.

Finally, the last step of the calibration procedure is the measurement of a known voltage (*AVSS* is the preferred value, as a reference).

The measurement procedure in calibration mode is the following:

1. Setup the ADC Control Register
   - Set the calibration mode, i.e set *ADC_MODE=1*
   - Set ADC input according to Table 3.31. For example for *AVSS*, set *ADC_INPUT_SELECTION=0*
     - if *ADC_INPUT_SELECTION* is on *DACMONV* (i.e value is 5), set Voltage DAC input according to Table 3.25. For example for *VCASP*, set *VOLTAGE_DAC_SELECTION=1*
     - if *ADC_INPUT_SELECTION* is on *DACMONI* (i.e value is 6), set Current DAC input according to Table 3.25. For example for *IBIAS*, set *CURRENT_DAC_SELECTION*= **Beware, the *IREF* input shall never be used**
   - Set the *comparator current*
   - Set the *comparator output sign*
   - Set the *ramp speed*
   - Set the *half bit trimmer*
2. Send a *START_MEASURE* pulse: command *ADCMEASURE* (0xFF20 , see Table 3.3)
3. After a time $\leq$ 5ms (see Table 3.42) read the *CAL* register (address 0x0612), whatever the settings of *ADC INPUT SELECTION*, *VOLTAGE DAC SELECTION* and *CURRENT DAC SELECTION*.

3.10.5 Automatic Measurement

This procedure shall not be used in background of data recording as it genererates an increase of the current consumption with a transient when the *IREF* input is selected.

The automatic measurement procedure is used to automatically sample all ADC inputs with a single *START_MEASURE* pulse.

After setting the Automatic Measurement Mode in the ADC Control Register, when receiving a *START_MEASURE* pulse, the ALPIDE chip will automatically switch both ADC input and current and voltage DAC monitor selection to sample all ADC inputs.

The ALPIDE chip includes hardware protections to avoid bad switches settings. Therefore, to ensure the validity of the read values, this automatic operation mode shall never be used when DACMONV or DACMONI are overridden (see 4.3).

The measurement sequence is the following:

1. *AVSS*
2. *DVSS*
3. *AVDD*
4. *DVDD*
5. *VCASN*
6. *VCASP*
7. *VPULSEH*
8. *VPULSEL*
9. *VRESETP*
10. *VRESETD*
11. *VCASN2*
12. *VCLIP*
13. *VTEMP*
14. *IRESET*
15. *IAUX2*
16. *IBIAS*
17. *IDB*
18. *IREF* **the value is not correct for this input**
19. *ITHR*
20. *BANDGAP*
21. *TEMPERATURE*

After a time $\leq$ 100ms (see Table 3.42), each measured value can be read in its dedicated register, which addresses are from 0x0613 to 0x0627 (Table 3.23).

Procedure

1. Setup the ADC Control Register
   - Set the automatic mode, i.e set *ADC_MODE=2*
   - *ADC input* is not used. Set any value, for example set *ADC_INPUT_SELECTION=0*
   - Keep the *comparator current* field value used at calibration step
   - Keep the *comparator output sign* field value used at calibration step.
   - Keep the *ramp speed* field value used at calibration step
   - Keep the *half bit trimmer* field value used at calibration step
2. Send a *START_MEASURE* pulse, command *ADCMEASURE* (0xFF20 , see Table 3.3)
3. After a time $\leq$ 100ms (see Table 3.42) read the measured values (register address 0x0613 to 0x0627, see Table 3.23)

4 Principles of Operation

4.1 Pixel circuits. Analog Front-End and Digital Pixel

4.1.1 Analog Front-End

The ALPIDE pixel matrix has 512 rows and 1024 columns. The pixel width is 29.24 μm and the pixel height is 26.80 μm. A comprehensive scheme for the pixel front-end circuit is shown in Figure 4.1.

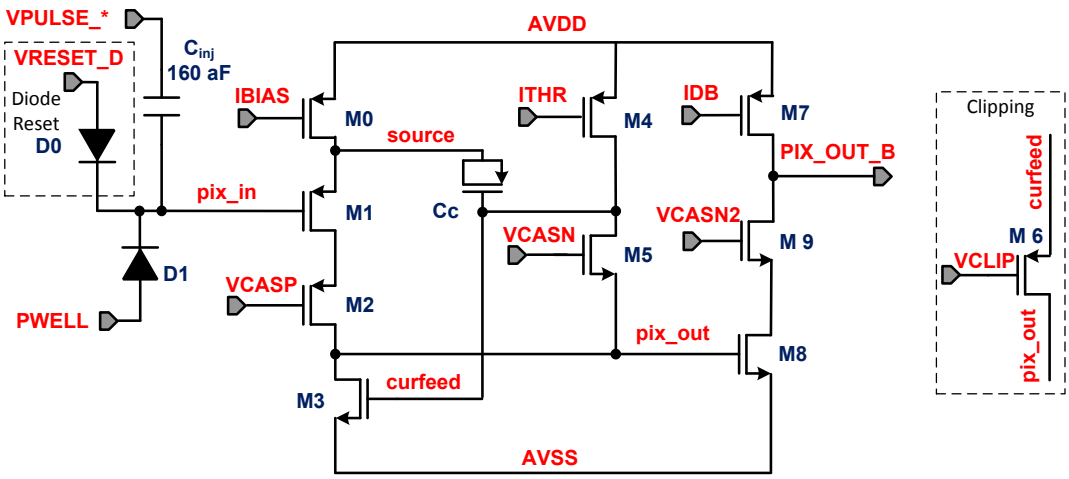

Figure 4.1: ALPIDE Front-end scheme

The collection n-well has octagonal shape with 2 μm diameter and n-well to p-well spacing of 3 μm. Diode D1 is the sensor p-n junction. The input node is continuosly reset by diode D0. VRESETD establishes the reset voltage of the charge collecting node (pix_in). A particle hit will lower the potential at the pixel input pix_in by a few tens of mV. This will cause the source follower formed by M1 and the current source M0 to force the source node to follow this voltage excursion and dump charge associated with the voltage change and the capacitance of the source node onto the analog output node pix_out. In addition the coupling of the source node to the curfeed node by Cc will cause the current in M3 to be reduced. Both effects add and cause the potential on the output node pix_out to swing upwards by several hundreds of mV. This forces M8 into conduction and if the charge deposit from the particle hit is sufficiently large to overcome the current setting IDB on M7, M8 will drive the PIX_OUT_B node to zero. The charge threshold of the pixel is defined by ITHR, VCASN and IDB. The effective charge threshold is increased by increasing ITHR or IDB. It is decreased by augmenting VCASN. The cascode transistor M9 reduces the equivalent Miller capacitance on pix_out. Voltage bias VCLIP controls the gate of the clipping transistor M6. The lower VCLIP, the sooner the clipping will set in. The active low PIX_OUT_B signal is applied to the digital section of the pixel where it is used to set the hit status register.

It is possible to electrically inject a test charge into the input node. This is achieved by applying a voltage pulse of controllable amplitude to the VPULSE pin of the C_{inj} capacitor. The design value of C_{inj} is 230 aF. The pulse generation is controlled by the digital section of the pixel and the periphery.

4.1.2 Digital Pixel

The digital section of the pixel is illustrated in Figure 4.2. The corresponding signals are listed in Table 4.1. The pixel features three State Registers. Each State Register is a Set-Reset Latch that can keep the hit information. The State Register is normally set by the front-end discriminated output PIX_OUT_B if the corresponding STROBE_B signal (STROBE_B<0> for State Register 0, STROBE_B<1> for State Register 1, STROBE_B<2> for State Register 2) is asserted simultaneously. It can also be set programmatically by the DPULSE signal (digital pulse functionality) if the corresponding STROBE_B is asserted simultaneously. The State Register can be selected for read/reset by asserting the corresponding MEMSEL_B signal (again <0>, <1> or <2>, depending on the State Register). The selected State Register is reset either by a PIX_RESET pulse generated by the Priority Encoder during the readout, either by a global FLUSH_B signal. The State Register is sensitive to the falling edge of PIX_RESET and it is level sensitive with respect to the FLUSH_B input. The selected State Register output bit can be masked and the result is the output to the Priority Encoder (STATE signal). If no State Register is selected STATE is 0.

The logic provides two programmable functions: masking and pulsing. When control bit MASK_EN is set high, the STATE output is forced to 0, effectively masking the pixel otuput to the priority encoder. The low value provides normal functionality. The testing functionalities are enabled by setting PULSE_EN=1, disabled otherwise. DPULSE assertion allows for the pixel digital pulsing. This consists in forcing to logic high the hit latch (STATE_INT), bypassing the pixel front-end signal. This can be done asserting DPULSE. The analog testing consists in the injection of test charge in the input node through Cinj (230 aF nominal). The amplitude of the applied voltage pulse is defined by the difference between VPLSE_HIGH and VPLSE_LOW, both set in the DAC unit. Please note that the two edges of the pulse provoke the injection of two charge pulses of opposite polarities. The rising edge of APULSE corresponds to the discharge of the collection diode, in a manner equivalent to the passage of a charged particle. There are two D-latches to store the PULSE_EN and MASK_EN configuration bits. Please note that their values after power-on are undefined. Setting of these latches is done by the PIXCNFG_COLSEL, PIXCNFG_ROWREGPSEL, PIXCNFG_ROWREGMSEL, PIXCNFG_DATA lines, all driven by the periphery control circuitry. The addressing of the pixels for configuration is based on the simultaneous selection of a specific row and a specific column. The simultaneous assertion of PIXCNFG_COLSEL and PIXCNFG_ROWREGMSEL pixel inputs selects the mask latch. The simultaneous assertion of PIXCNFG_COLSEL and PIXCNFG_ROWREGPSEL pixel inputs selects the pulse latch. PIXCNFG_DATA provides the value to be stored in the selected latch. There is no direct way to read back the values in the latches from the control interface.

4.2 Priority Encoders and pixel indexing

Looking at the chip with the digital periphery on the bottom, the leftmost region is region 0 and the rightmost region is region 31 (see Figure 4.3).

Each region contains 16 double columns. Double column 0 is the leftmost and double column 15 is the rightmost (see Figure 4.4).

The matrix of pixels is read out by an array of 512 Priority Encoder blocks. The pixels are arranged in double columns and the regions at the middle of each double column are occupied by the Priority Encoders. The indexing of the pixels in the readout data words is defined by the Priority Encoders. The indexing of the pixels in each double column is illustrated in Figure 4.5.

4.3 Analog bias and internal DACs

The ALPIDE chip has eleven internal 8-bit DACs: 6 voltage and 5 current DACs, used to set biases required by the pixel front end circuits. Figure 4.6 shows the DACs scheme. Table 4.2

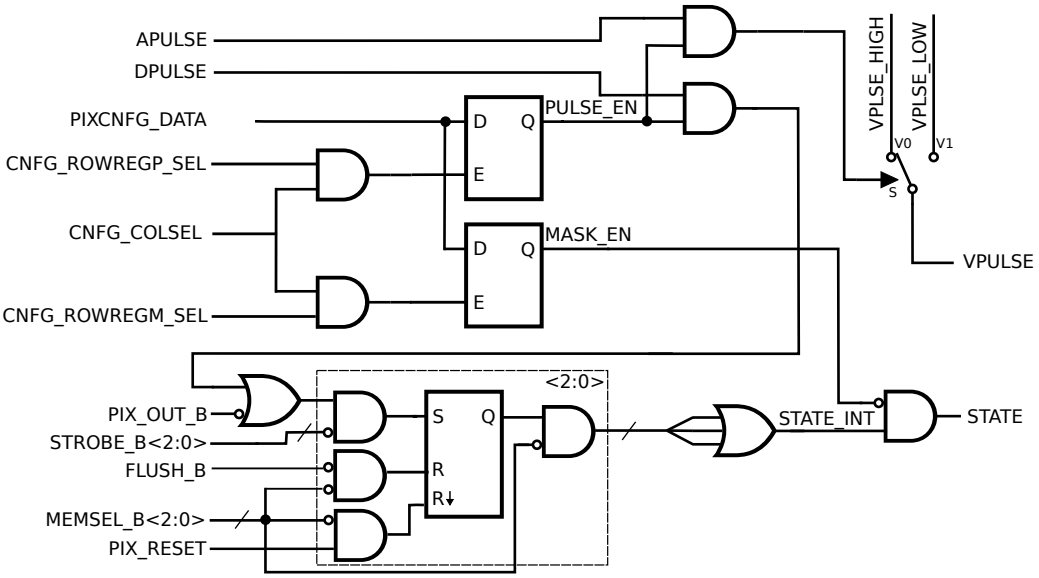

Figure 4.2: Functional diagram of the pixel logic

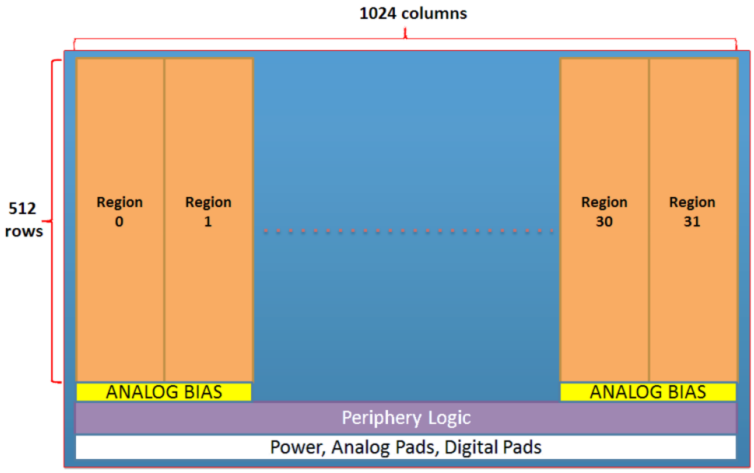

Figure 4.3: Region numbering

provides an overview of the specifications of the DACs.

The DAC block has three operation modes:

1. Normal - the outputs of all DACs are connected directly to the pixel matrix.
2. Monitor - it is possible to select a voltage DAC and monitor its output on the DACMONV pad. It is also possible to select a current DAC and monitor its output on the DACMONI pad.
3. Override - it is possible to override the output of one selected voltage DAC by the DACMONV pad. It is possible to override the output of one selected current DAC by the DACMONI pad. It is also possible to override the internally generated IREF current that defines the LSB value of the current DACs.

The voltage DACs are based on a 256 stages resistive divider connected between AVDD and AVSS (one divider in common for all 6 voltage DACs). Each resistor has a nominal value

	Signal	Description	Logic level
	APULSE	VPULSE voltage level selection if PULSE_EN = 1	Positive edge charge injection
	DPULSE	Digital Pulse if PULSE_EN = 1	Active high
	PIXCNFG_DATA	Configuration data	D-LATCH data line
	PIXCNFG_COLSEL	Column selection	Active high
	PIXCNFG_ROWREGPSEL	Row and Pulse reg. selection	Active high
	PIXCNFG_ROWREGMSEL	Row and Mask reg. selection	Active high
	PIX_OUT_B	Pixel front-end output	Active low
	STROBE_B<2:0>	Enable State register for hit acquisition	Active low
	MEMSEL_B<2:0>	Select State register for read and reset	Active low
	FLUSH_B	General reset of the selected the state register(s)	Active high
	PIX_RESET	Priority encoder reset of the selected register	Effective on falling edge
	VPLSE_HIGH	Analog pulse high level	Analog
	VPLSE_LOW	Analog pulse low level	Analog
	MASK_EN	State register mask enable	Active high
	STATE_INT	State register data	Active high
	VPULSE	Voltage step for test charge injection into pix_in net	$Q_{inj} = \Delta(\text{VPULSE}) \cdot 160$ aF
	STATE	State register value to priority encoder (if MASK_EN = 0)	Active high

Table 4.1: Signals of the pixel cell

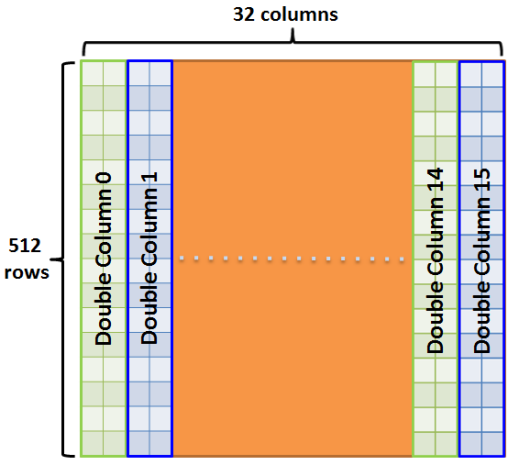

Figure 4.4: Double column numbering inside of a region

of 40 Ω, for a total resistance of 10.2 kΩ. This allows to generate voltage levels between AVSS and AVDD (256-1)/256 with 8 bit resolution. The values of the voltage DAC setting registers are decoded and used to control arrays of analog switches connected between the 256 nodes of the resistor divider and the output pins of the DACs. The VCASN and VCASP outputs are directly applied to the matrix without any amplification or scaling. The VRESET, VPLSE_LOW and VPLSE_HIGH and VAUX outputs are buffered with unit gain followers. This causes an offset of about 370 mV and saturation for codes above about 200 for these DACs. At the nominal bias value (AVDD = 1.8 V) the current sunk by the AVDD pad is ≈ 180 μA. The current DACs are implemented by repeating 255 times the same building unit that is a current source generating the current corresponding to the LSB. This is 1/256 of IREF, an internally generated reference current. IREF is nominally 10.24 μA, the LSB

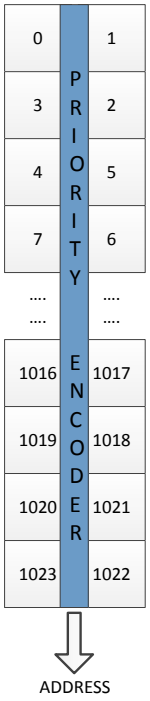

Figure 4.5: Indexing of pixels inside a double column provided by the Priority Encoders

value is nominally 40 nA. The values of the current DACs setting registers are decoded and used to control the analog switches connecting the LSB sources in parallel into the output node of each DAC. The outputs of the current DACs are then scaled to appropriate levels before being applied to the matrix. The scaling factors are given in Table 4.3. The switching of the DACs is thermometer coded for improved linearity: for the current DACs current sources corresponding to 1 LSB are gradually switched into the DAC output as the DAC code increases. Once a current source has been used in the DAC output, it will not be removed and replaced by another one to reach a higher DAC code. Since the voltage DACs are resistive divider based, they are naturally thermometer coded.

4.3.1 Monitoring and Overriding of the DACs

It is possible to monitor the output of a selected voltage DAC using the DACMONV pad. The DACMONV pin should be monitored with a high input impedance circuit (Rin $>$ 1 MΩ). Only one voltage can be monitored at a given time. It is possible to monitor the output of a selected current DAC using the DACMONI pad loaded with a shunt resistor to AVSS. The recommended shunt resistance is 5 kΩ. Only one current can be monitored at a given time. The current on the shunt resistor is equal to ten times the output current of the selected DAC, upstream the scaling towards the pixels. It is possible to override a selected voltage DAC using the DACMONV pad. Once the functionality is activated a voltage between 0 and AVDD needs to be applied to the DACMONV pad. This will feature high input impedance. The voltage applied to DACMONV goes directly to the pixel matrix. It is possible to override a selected current DAC using the DACMONI pad. Once the functionality is activated a current needs to be sourced from DACMONI as illustrated in Figure 4.7.

This current is divided by 10 internally and this replaces the output of the DAC before the internal scaling towards the pixel matrix. The range of interest for the external overriding current is 0 to 200 μA, covering almost twice the internal nominal range. Finally the internal

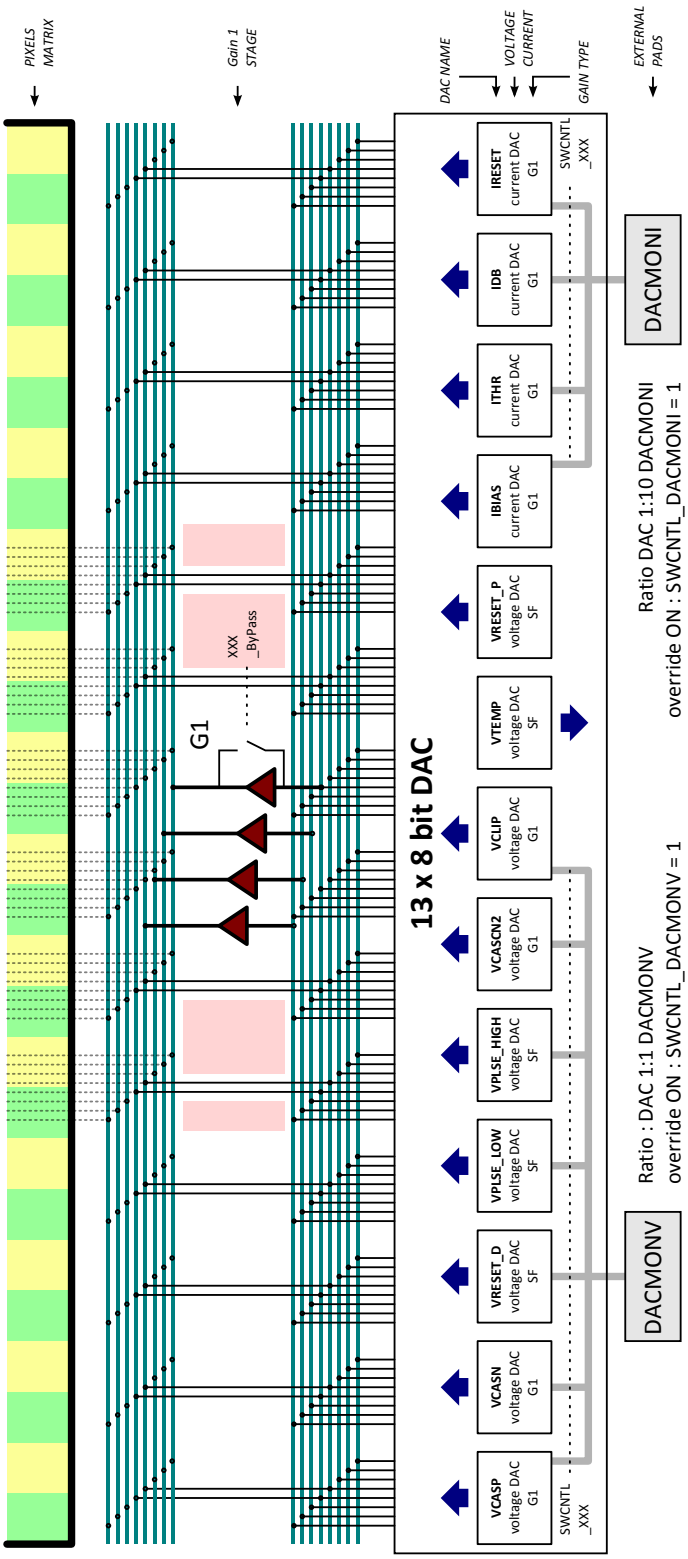

Figure 4.6: DACs scheme

IREF current constituting the reference for all the current DACs can be overridden. In this case the current sourced by the DACMONI pad is divided by 11 before being used by the internal DACs. The configuration of the DAC block for monitoring or overriding and the

	Minimum	Maximum	Nominal setting	Nominal value
IBIAS	0 nA	80 nA	64	20 nA
ITHR	0 nA	80 nA	51	0.5 nA
IDB	0 nA	80 nA	64	10 nA
IRESET	0.7 nA	26 pA	50	5 pA
IAUX2	-	-	-	-
VCASP	0 V	1.8 V	86	0.6 V
VCASN	0 V	1.8 V	57	0.4 V
VCASN2	0 V	1.8 V	62	0.44 V
VCLIP	0 V	1.8 V	0	0 V
VRESET_P	0.37 V	1.8 V	117	1.2 V
VRESET_D	0.37 V	1.8 V	147	1.4 V
VPLSE_LOW	0.37 V	1.8 V	0	0.37 V
VPLSE_HIGH	0.37 V	1.8 V	255	1.8 V

Table 4.2: DACs specifications overview.

DAC	Scaling from DAC to Matrix	Scaling from DACMONI to Matrix
IBIAS	1:128	1:1280
ITHR	1:4096	1:40960
IDB	1:256	1:2560
IRESET	$\approx$ 1:4 $\times$ 10^5	$\approx$ 1:4 $\times$ 10^6

Table 4.3: Scaling factors for the current DACs.

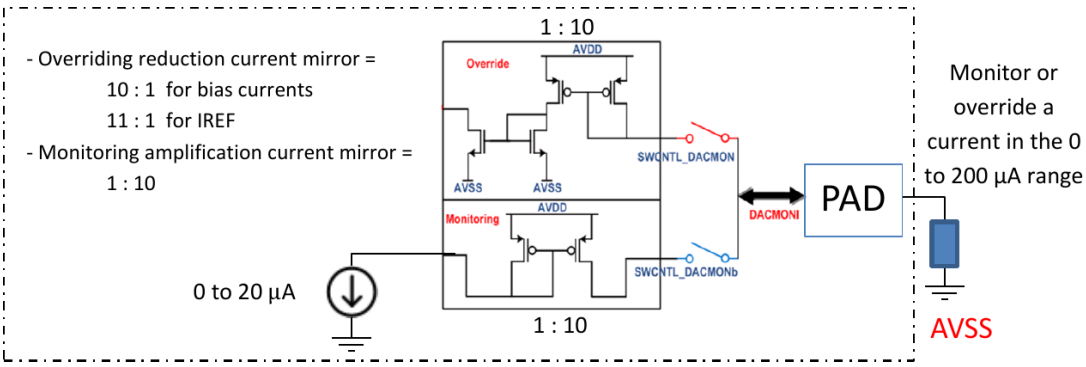

Figure 4.7: Current DACs monitoring and overriding scheme

selection of the DACs are done by the dedicated Current/Voltage Monitoring and Overriding control register

4.4 Shadow registers and Debug streams

Debug Streams

When a DEBUG command is received, the ALPIDE peripheral logic store a snapshot of a large number of internal registers into shadow registers. This allows to sample at a controlled moment the values of state machines registers, counters, fifo pointers, ports and many other signals of potential importance for observability and debugging. Refer to fig. 4.8 for an illustration of this feature.

The shadow registers are chained in serial shifting chains and can be read by subsequent read operations of dedicated registers. There are distinct registers chain, corresponding to the major functional modules of the chip. Each of the chains has a corresponding Debug Stream read register. Consecutive read accesses to these registers produce a stream of words. The format of these streams is detailed in the following paragraphs, with reference to each of the Debug Stream read registers (see section 3.2.6).

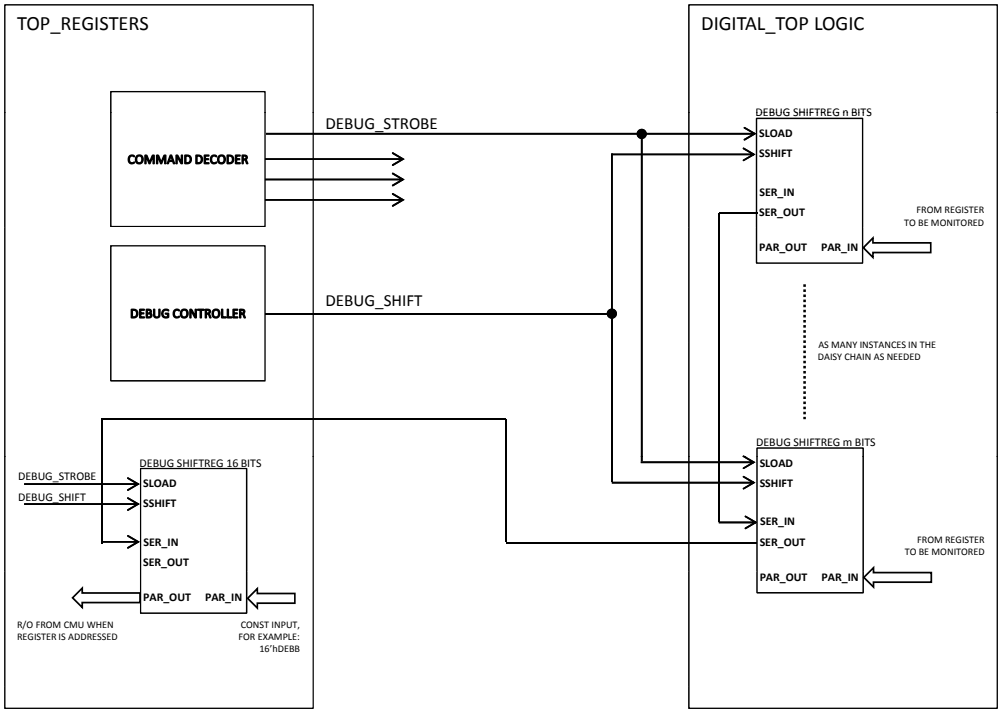

Figure 4.8: Debug Chain Prinicple of Operation.

0x0702 - BMU Debug Stream

Busy Management Unit debug stream is composed of 2 words.
Word 0 is the stream header with value 0xDEBB

Bits	Description
15:14	Busy Request State Machine
13	Busy Generator State Machine
12	Busy In State
11	SEU Error OR
10:0	Not Used

Table 4.4: BMU Debug Stream word 1

0x0703 - DMU Debug Stream

Data Management Unit debug stream is composed of 4 words.
Word 0 is the stream header with value 0xDEBD

Bits	Description
15:14	Data FIFO Read pointer 2
13:12	Data FIFO Write pointer 2
11:10	Data FIFO Read pointer 1
9:8	Data FIFO Write pointer 1
7:6	Data FIFO Read pointer 0
5:4	Data FIFO Write pointer 0
3:2	Busy FIFO Read pointer
1:0	Busy FIFO Write pointer

Table 4.5: DMU Debug Stream word 1

Bits	Description
15:8	Local Bus Value
7	Local Bus OEN Generator State Machine
6:4	Mux Fifo Ctrl State Machine
3:2	Ctrl Word Decoder Local Bus State Machine
1:0	Token Grantor State Machine

Table 4.6: DMU Debug Stream word 2

Bits	Description
15:14	Send Comma State Machine
13:11	Data Packing State Machine
10	SEU Error OR
9	Busy Mismatch Error
8	Busy FIFO Error
7	Data FIFO Error
6:0	Not Used

Table 4.7: DMU Debug Stream word 3

0x0704 - TRU Debug Stream

Top Readout Unit debug stream is composed of 5 words.
Word 0 is the stream header with value 0xDEB7

Bits	Description
15:10	Frame Start FIFO Read pointer 2
9:4	Frame Start FIFO Write pointer 2
3:0	Frame Start FIFO Read pointer 1, bits 5:2

Table 4.8: TRU Debug Stream word 1

Bits	Description
15:14	Frame Start FIFO Read Pointer 1, bits 1:0
13:8	Frame Start FIFO Write pointer 1
7:2	Frame Start FIFO Read Pointer 0
1:0	Frame Start FIFO Write pointer 0, bits 5:4

Table 4.9: TRU Debug Stream word 2

Bits	Description
15:12	Frame Start FIFO Write Pointer 0, bits 3:0
11:6	Frame End FIFO Read pointer
5:0	Frame End FIFO Write Pointer

Table 4.10: TRU Debug Stream word 3

Bits	Description
15:13	TRU State Machine
12:11	Clock Enable State Machine
10	SEU Error OR
9	Frame Start FIFO Error
8	Frame End FIFO Error
7:0	Not Used

Table 4.11: TRU Debug Stream word 4

0x0705 - RRU Debug Stream

Region Readout Unit debug stream is composed of 65 words.
Word 0 is the stream header with value 0xDEB8. For each region, from 31 to 0, there are 2 words.

Bits	Description
15:9	MEB FIFO Read pointer
8:2	MEB FIFO Write pointer
1:0	MEB FIFO State Machine

Table 4.12: RRU Debug Stream word 1

Bits	Description
15	SEU Error OR
14:13	Region Readout State Machine
12:11	Region Valid State Machine
10	Gen RGN Header State Machine
9:8	Fifo Self Test State Machine
7:6	Ro Clock Enable State Machine
5:4	Tru Clock Enable State Machine
3:2	Cfg Clock Enable State Machine
1:0	Ft Clock Enable State Machine

Table 4.13: RRU Debug Stream word 2

0x0706 - FROMU Debug Stream

FROMU debug stream is composed of 9 words.
Word 0 is the stream header with value 0xDEBF

Bits	Description
15:13	Not Used
12	SEU Error OR
11:0	Bunch Counter

Table 4.14: Fromu Debug Stream word 1

Word 2 - Trigger Counter

Word 3 - Strobe Counter

Word 4 - Frame Counter

Word 5 - Readout Counter

Bits	Description
15:4	Bunch Counter
3:1	Strobe Manager State Machine
0	Flush Value

Table 4.15: Fromu Debug Stream word 6

Bits	Description
15:11	Writer State machine
10:6	Reader State machine
5:0	PRST State Machine

Table 4.16: Fromu Debug Stream word 7

Bits	Description
15:13	Strobe Value
12:10	Memsel Value
9:8	Busy Manager State Machine
7:5	Event In MEB
4:0	Not Used

Table 4.17: Fromu Debug Stream word 8

0x0707 - ADC Debug Stream

ADC debug stream is composed of 4 words. Word 0 is the stream header with value 0xDEBA

Bits	Description
15	Comparator Out
14:4	Sampled Value
3:1	Select DAC Row
0	Select DAC Column, bit 7

Table 4.18: ADC Debug Stream word 1

Bits	Description
15:9	Select DAC Column, bits 6:0
8:6	ADC State Machine
5:0	ADC Sequencer State, bits 7:1

Table 4.19: ADC Debug Stream word 2

Bits	Description
15	ADC Sequencer State, bit 0
14	SEU Error OR
13:0	Not Used

Table 4.20: ADC Debug Stream word 3

4.5 Data Transmission Unit principles of operation

The Data Transmission Unit (DTU) provides a fast serial link for the transmission of the data from the Alpide ASIC. Alpide integrates in the same substrate the silicon pixel detector and the readout electronics for the Inner Tracking System (ITS) of the ALICE experiment at the LHC accelerator. The DTU has to work in two different environments:

1. Inner Barrel (IB) : each chip transmits its data over a differential serial line with a line rate of 1.2 Gb/s to the off-detector electronics (*Readout Unit*). The data stream is transmitted over a aluminum over kapton FPC (*Flexible Printed Circuit*) with a maximum length of 300 mm and then to a micro-coaxial cable over a length of $\tilde{5}$ m.
2. Outer Barrel (OB) : each half module has a master chip which receives the data from 6 other chips and send them over a serial line with a line rate of 400 Mb/s to the RU. The data stream is transmitted over a copper over kapton FPC with a maximum length of 1.5 m, again followed by $\tilde{5}$ m of micro-coaxial cable.

The DTU interface is designed to have an output signaling compatible with the LVDS standard (at least for the voltage swing, the common mode voltage is reduced to 0.9 V). Resistive termination is required.

Input clock	40 MHz
Transmission clock	600 MHz
Trasmission type	DDR
Line rate (inner layers)	1.2 Gb/s (600 Mb/s, 400 Mb/s)
Line rate (outer layers)	400 Mb/s
Data encoding	8b10b
Data rate (inner layers)	960 Mb/s
Data rate (outer layers)	320 Mb/s
Electrical protocol	*(pseudo)*LVDS

Table 4.21: Main specifications of the Data Transmission Unit

Line rate (inner layers)	1200/600/400	Mb/s
Line rate (outer layers)	400	Mb/s
Electrical protocol	LVDS	
Load termination	100	Ω
Current range	0÷5	mA
Pre-emphasis current range	0÷2.5	mA
Common mode voltage	900	mV
Input termination	no	
Total jitter (max)	0.2	UI

Table 4.22: Specifications of the DTU Line Driver

The main specifications for the DTU are summarized in table 4.21.

The driver has to provide a line rate up to 1.2 Gb/s over a cable length of up to 5 m. Therefore a pre-emphasis function will be implemented in order to compensate for limitation due to the RC delay of the line. Table 4.22 summarize the driver specifications

The 600 MHz transmission clock is provided by an on-chip PLL.

The DTU architecture, shown in figure 4.9, is based on a Double Data Rate (DDR) transmission scheme. The 3-bytes input word is converted in a 30 bits word by the DMU with the 8b10b encoding and loaded into two 15 bits shift registers every 40 MHz clock cycle (providing that a load enable signal is asserted). The two shift registers are synchronized on the two 600 MHz clock edges and drive the line driver after a single ended to differential conversion. A secondary path, equal to the main one but with two extra delay latches drives a second driver in order to provide pre-emphasis.

The 1.2 Gb/s line rate, combined with the 8b10b gives a data rate of 960 Mb/s for the IB, while the 400 Mb/s line rate gives 320 Mb/s for the OB. Assuming that the input bus is organized in bytes and the clock frequency is 40 MHz, this implies that the number of bytes processed per clock cycle is 3 for the IB and 1 for the OB.

The DTU interface with the core logic and with the external world are listed in table 4.23.

The LVDS driver provides a current between 0 and 5 mA with a 0.312 mA resolution over a 100 Ω differential cable. The driver is compatible with both commercial LVDS receivers and the GBTX SLVS receivers. A pre-emphasis current of up to 50% has been implemented in order to be able to compensate for excessive RC on the cable. The pre-emphasis time width is one bit period, i.e. the current bit is emphasized if different from the previous one. The output common mode has been set to 0.9 V (i.e. lower than the 1.2 V from the LVDS standard) in order to reduce the power consumption and have a better match with the 1.8 V supply voltage.

The driver receives the single-ended outputs of the odd and even serializer, as well as the delayed (by half a clock cycle) versions of the two stream. The two pairs are fed into two

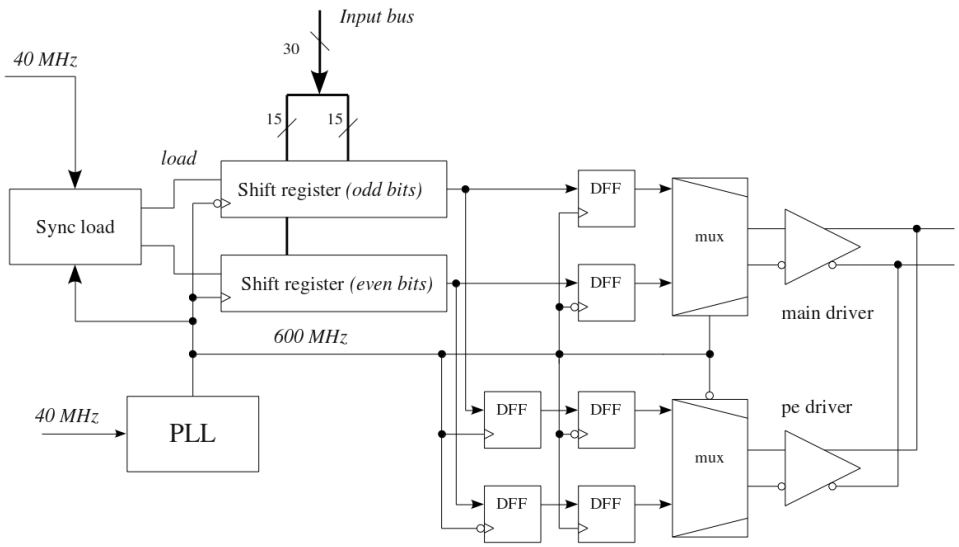

Figure 4.9: Functional diagram of the Data Transmission Unit

Signal name	# of bits	Direction	Remarks
RESET_B	1	Input	Active Low
CLOCK	1	Input	40 MHz, must be clean
DISABLE	1	Input	DTU disable for power saving
LOAD_EN	1	Input	Load Enable
DATA_IN	30	Input	Already 8b10b encoded
PLL charge-pump current	4	Input	From 7.5 to 15 µA in 500 nA steps. Nominal value 10 µA.
PLL CFG<3>	1	Input	PLL off signal *(active high)*
PLL CFG<2>	1	Input	*Not used*
PLL CFG <1:0>	2	Input Input Input Input	PLL VCO delay stages control: 00: VCO with 3 stages *(slow case)* 01: VCO with 4 stages *(typical case)* 11: VCO with 5 stages *(fast case)*
PLL LOCK	1	output	Lock detector output
DRVOUT	4	Input	From 0 to 5 mA
PEOUT	4	Input	From 0 to 2.5 mA
DATA_OUT	2	Output	Differential, LVDS levels

Table 4.23: DTU interface signals

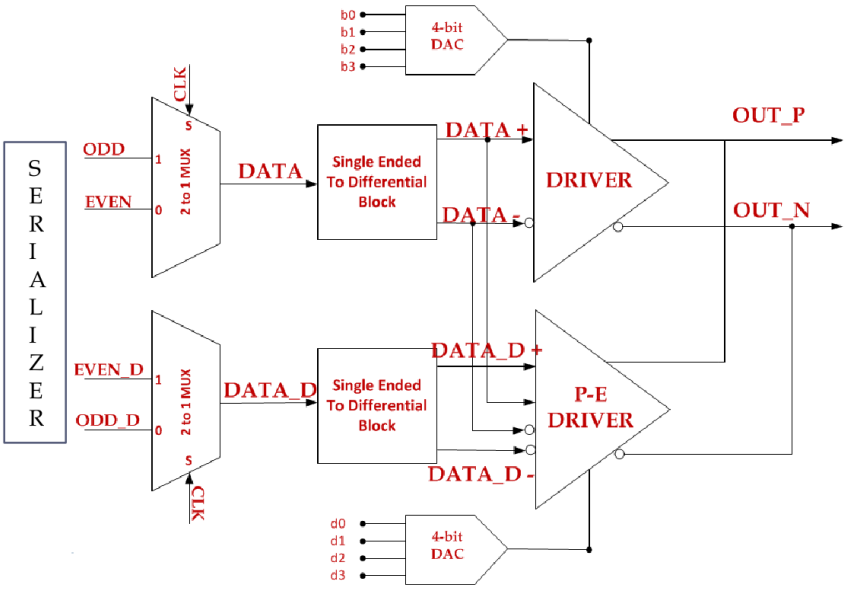

Figure 4.10: Functional diagram of the Data Transmission Unit

clock-driven multiplexers to obtain a single data stream (and its delayed copy). These two signals are converted to differential and drive the main driver and the pre-emphasis driver, respectively. The current given by the two drivers is controlled via two 4-bits DACs. The driver schematic is shown in figure 4.10.

The Double Data Rate (DDR) serializer is based on two 15 bits shift registers (10 bits in the current prototype). The two inputs are connected to the even and odd bits of the 8b10b encoder outputs. The input data are loaded every 40 MHz clock cycle, providing that the LOAD_EN signal is asserted. A dedicated circuit generates the actual shift register load signal and guarantees the synchronization between the 40 MHz and the 600 MHz clock.

The serializer also provides the delayed copies of the two data streams. The 400 Mb/s operation mode is performed by simply copying 3 times each bit on the 30 bits bus. *It should be noted which such an arrangment the pre-emphasis in the 400 Mb/s case is active only for 1/3 of the bit period.*

The circuit is protected against SEU via TMR.

The serializer is loaded with a bit pattern that is already including 8b10b encoding. The 8b10b encoder is integrated in the chip core logic and it is not part of the DTU block.

The specifications of the PLL are summarized in table 4.24. A simplified schematic of the PLL is shown in figure 4.11. It is based on the charge pump architecture commonly used in integrated PLLs. The input 40 MHz clock is fed into a Phase-Frequency Detector (PFD). A Voltage Controlled Oscillator (VCO) with a center frequency of ˜600 MHz provides the high frequency clock, which is first divided by 3 and then by 5 to obtain a ÷15 frequency which is compared with the one at the input of the PFD. Therfore both the 600 MHz and the 200 MHz frequencies are available.

The PFD output fed a charge pump – loop filter circuit which controls the VCO oscillation frequency, thus closing the loop. The loop filter comprises a 80 pF main capacitor in series with a 12 kΩ damping resistor. A second 15 pF capacitor is added in parallel to the main RC series to help decreasing the effect of the noise on the supply voltage.

Note: the C_2 capacitor is connected to the VCO input via the uppermost metal layer. It is therefore possible to reduce the total capacitance to 1,3,5,7,9,11 and 13 pF via FIB in case of stability problems due to unexpected parasitic capacitance on the VCO_ctrl node.

Parameter	Conditions	Min	Typ	Max	Unit
Technology			TowerJazz 0.18 CIS		
Die area			0.12 × tbd		
Supply voltage		1.6	1.8	1.92	V
Supply current			tbd		mA
Temperature range		0	25	85	°C
Multiplication factor			15		
Input frequency			40		MHz
VCO frequency		500	600	700	MHz
Output frequency	200-600 MHz	200		600	
Output duty cycle		45	50	55	
Output jitter	pk-pk ($\pm 3\sigma$)			80	ps
Output capacitive load				50	fF
Lock time		6	9	12	μs
Logic levels			CMOS		

Table 4.24: Specifications of the PLL

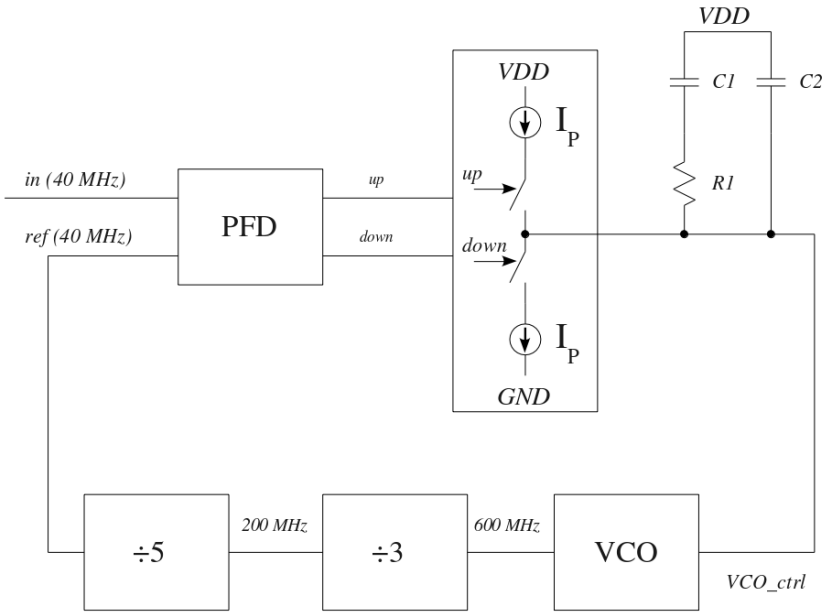

Figure 4.11: Simplified schematic of the PLL circuit.

Appendices

Appendix A Application note. Chip and modules clocking schemes

Figure A.1 illustrates the clocking schemes supported by the chip.

The chip has two differential ports dedicated to the clock signals, DCLK and MCLK (section 2.2).

- The *receiver* of the DCLK port is always enabled.
- The internal clock used by the core circuits is always the signal received on the DCLK port.
- In the IB module scenario (left side of the figure), the chips receive the clock from an external circuit driving the multi-drop differential line connected to the DCLK ports.
- When the chip is configured to operate as an Outer Barrel Master (CHIPID[2:0]=3'b000), the *driver* of the DCLK port and the *receiver* of the MCLK port are enabled. The signal received on the MCLK port is internally buffered and forwarded to the driver of the DCLK port. This allows to implement a local regeneration of a clock signal and a local clock bus (LCLK) on a module.
- The line connected to the DCLK port of a chip configured as OB Master shall not be driven by any external circuit. The external clock shall be applied to the MCLK port of an OB Master chip.
- When the chip is configured to operate as an Outer Barrel Master or it is configured as an Outer Barrel Slave with the specific value of the *chipid* field CHIPID[3:0]=4'b0110=d6, an on-chip termination resistor (100 Ω) gets activated on the DCLK port. This is to remove the need of connecting termination resistors on the LCKL local clock bus.
- The *driver* of the MCLK port is never enabled.

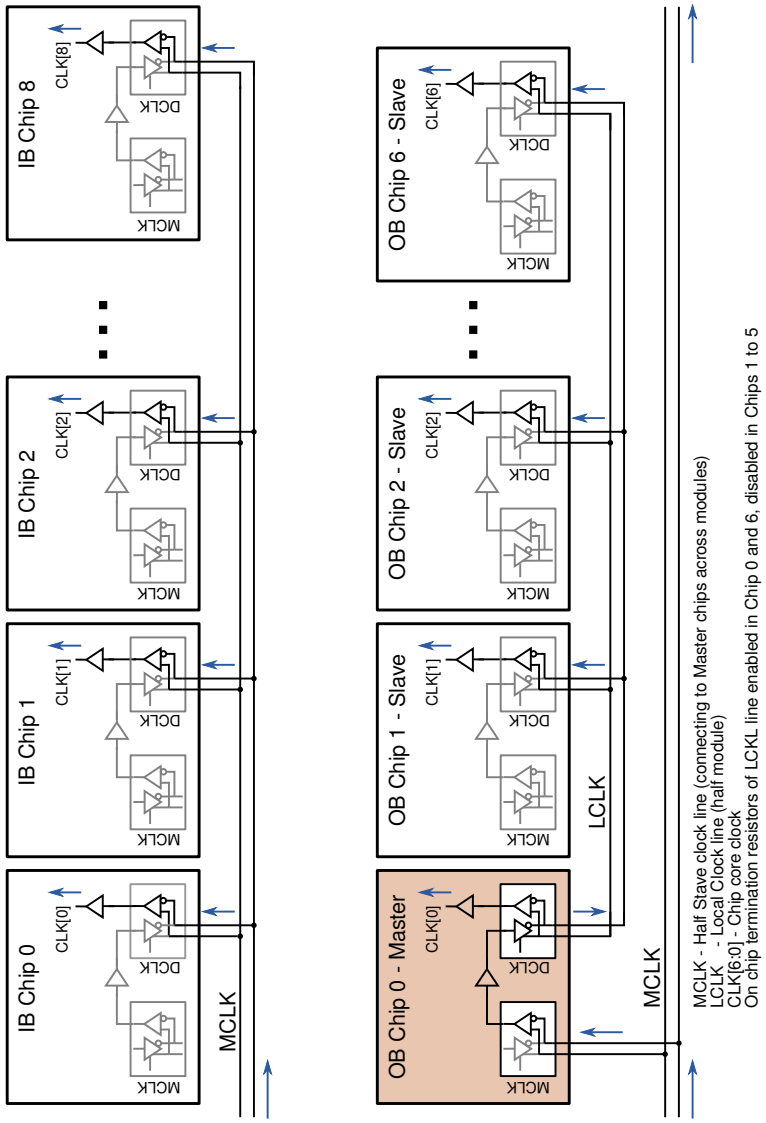

Figure A.1: Illustration of the clock distribution scheme for the ITS Inner Barrel and Outer Barrel Modules.

Appendix B Application note. ALICE ITS Inner Barrel Modules

- Inner Barrel Module include 9 chips
- The 9 chips receive from the off detector electronics a global clock signal on the shared differential line MCLK.
- The MCLK lines connects in a multi-drop configuration the DCLK_P, DCLK_N terminal pairs. Termination of the MCLK line on transmitter side and module far end side is required.
- An unavoidable skew of the internal clocks related to the propagation delays on the line is expected.
- The nominal clock frequency for the Inner Barrel module prototypes is **40.08 MHz** (LHC clock frequency).
- A DCTRL differential line connects in a multi-point configuration the DCTRL_P, DCTRL_N terminals with a differential transceiver (MLVDS) on the off-detector side. Termination resistors shall be provided at both ends of the DCTRL line.
- Signaling on the DCTRL bus is half-duplex and synchronous to the clock. Topological symmetry between the DCTRL line and the MCLK line ensures that the sampling of the DCTRL bus by the chips can be achieved and maintained with correct timing.
- The off-detector electronics shall be capable to transmit serially on the DCTRL bus with a bit period twice (or at least equal) to the clock period.
- The off-detector electronics shall be capable to disable its line driver during the responding periods in which one chip on the line activates its own driver.
- Chips respond on DCTRL according to a pre-defined protocol (section 3.1.3).
- Chips responses on DCTRL are clock synchronous serial transmissions.
- The DCTRL signal sampled at the off-detector electronic side will present a changing phase with respect to the transmitted clock, depending on the distance of the transmitting chip down along the line. Over the 30 cm length of IB Module prototypes this is expected to be negligible. Techniques for appropriate re-timing could be foreseen on the off-detector electronics.
- Chips transmit their data off-detector over point-to-point uni-directional differential links (HSDATA). Signaling on these lines is at 1.2 Gbps by default. Lower rates (600 Mb/s and 400 Mb/s) can be selected optionally.

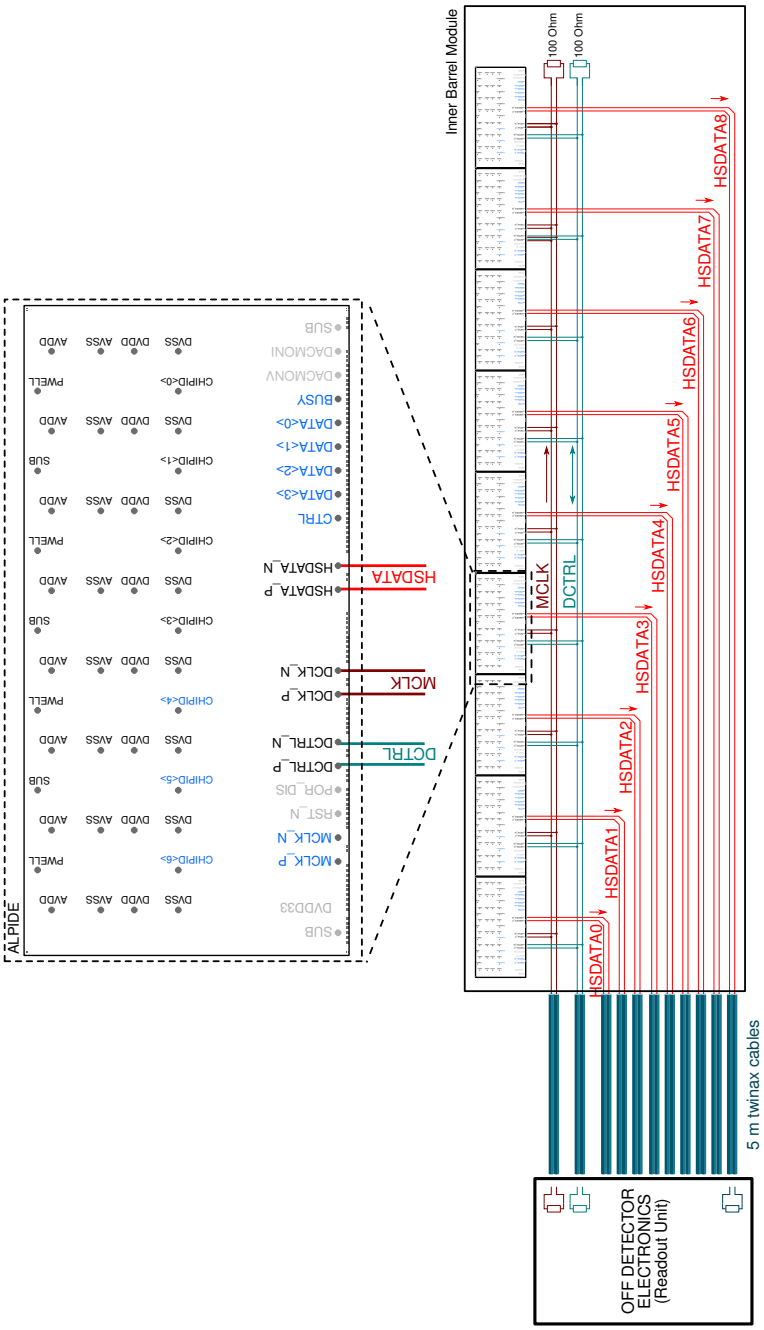

Figure B.1: Schematic diagram of the electrical interconnections between the ALICE ITS Upgrade Inner Barrel module and the off-detector electronics.

Appendix C Application note. ALICE ITS Outer Barrel Modules

- Outer Barrel Module prototypes include 14 chips divided in two sets of 7 (seven). Each subset includes one Outer Barrel Module Master and six associated Outer Barrel Module Slaves.
- The OB Module Master chips receive from the off detector electronics a global clock signal on a shared differential line MCLK.
- 7 (or 4) Master chips share one MCLK line routed along 7 (or 4) modules constituting one Outer (Middle) Layer half stave.
- The MCLK lines connects in a multi-drop configuration the MCLK_P, MCLK_N terminal pairs of the OB Module Master chips. Termination of the MCLK line on transmitter and far end sides is required.
- An unavoidable skew of the chip clocks related to the propagation delays on the line is to be expected.
- The nominal clock frequency for the Outer Barrel module prototypes is **40.08 MHz** (LHC frequency).
- The reference clock is forwarded by the Master chips to the Slave chips using a module local bus (LCLK).
- The reference clock is forwarded by the Master chips to the Slave chips using a module local bus (LCLK).
- The signal received on the MCLK port by the Master chips is replicated (driven) through the driver of the DCLK port.
- All chips receive the clock for the internal circuits through the receivers of the DCLK port.
- The LCLK bus gets terminated by on-chip termination resistors enabled on the DCLK ports of the OB Master chip and of the OB Slave chips with CHIPID[3:0]=6 or CHIPID[3:0]=14.
- A DCTRL differential line connects in a multi-point configuration the DCTRL_P, DCTRL_N terminals of the OB Master chips with a differential transceiver on the off-detector side.
- Proper termination of the line shall be provided at both extremities.
- Signaling on the DCTRL bus is half-duplex and synchronous to the clock. Topological symmetry between the DCTRL line and the MCLK line ensures that the sampling of the DCTRL bus by the OB Master chips can be achieved and maintained with correct timing.
- The off-detector electronics shall be capable to transmit serially on the DCTRL bus with a bit period twice (or at least equal) to the clock period.
- The off-detector electronics shall be capable to disable its line driver during the responding periods in which one of the OB Master chips on the line activates its own driver.
- Chips respond on DCTRL according to a pre-defined protocol (section 3.1.3).

- Chips responses on DCTRL are clock synchronous serial transmissions.
- The DCTRL signal sampled at the off-detector electronic side will present a changing phase with respect to the transmitted clock, depending on the distance of the responding chip down along the line. Means of appropriate re-timing should be foreseen on the off-detector electronics.
- A local control bus (LCLK) is implemented between the OB Master and the OB Slaves using the single ended CTRL ports.
- The OB Master acts as a hub on the control bus, replicating the characters received on the DCTRL port and forwarding from the CTRL to the DCTRL port when one of the associated Slaves is the target of a control read transaction.
- OB Module Master chips transmit their own data and data of the associated six OB Module Slaves on point-to-point uni-directional differential links, using the HSDATA output port.
- Serial transmission on the HSDATA port is at 400 Mbps.
- The signal sampled on the HSDATA links at the off-detector electronic side will present a changing phase with respect to the reference clock, depending on the length of the line between the transmitting chip and the receiver. Means of appropriate re-timing should be foreseen on the off-detector electronics.
- The data exchange between the OB Module Slave chips and the OB Module Master is realized on a OB Module Local DATA Bus.
- The OB Module Local Data Bus is a shared parallel bus realized interconnecting the parallel DATA ports of the chips.
- The 4 lowermost lines of the DATA port operate in Double Data Rate mode, with bits launched or sampled at both clock edges and one complete byte transfer at every clock cycle. The uppermost 4 bits can be left unconnected and the OB Local DATA Bus on the OB moules is implemented using 4 wires shared by the chips.
- Optionally, the chips can be configured with Single Data Rate signaling on the DATA[3:0] IOs and in this case the 8 wires DATA[7:0] would be interconnected across chips. This requires interconnection of the small pads of the chips.
- The OB Module chips drive the Local DATA Bus in turn. Write access to the bus is granted by default to a pre-configured chip. This chip (typically the OB Master) actively drives the DATA bus while idling. The other chips on the bus are configured with the identifier of a preceeding chip in a token exchange chain. All chips sample the data on the Local Bus monitoring the Chip Data Frame transmitted on it by the other chips. On detection of the completion of a frame by the preceeding chip in the chain, a given slave is granted the right to access the bus and transmit onto it a complete Chip Data Frame. At the end of one Chip Data Frame the chip disables the Local Bus Drivers and enters a waiting state for a new time access slot.
- The ordered sequence of chip identifiers governing the write access to the Local DATA Bus and the chip having access to the bus by default are programmable by means of dedicated configuration registers.

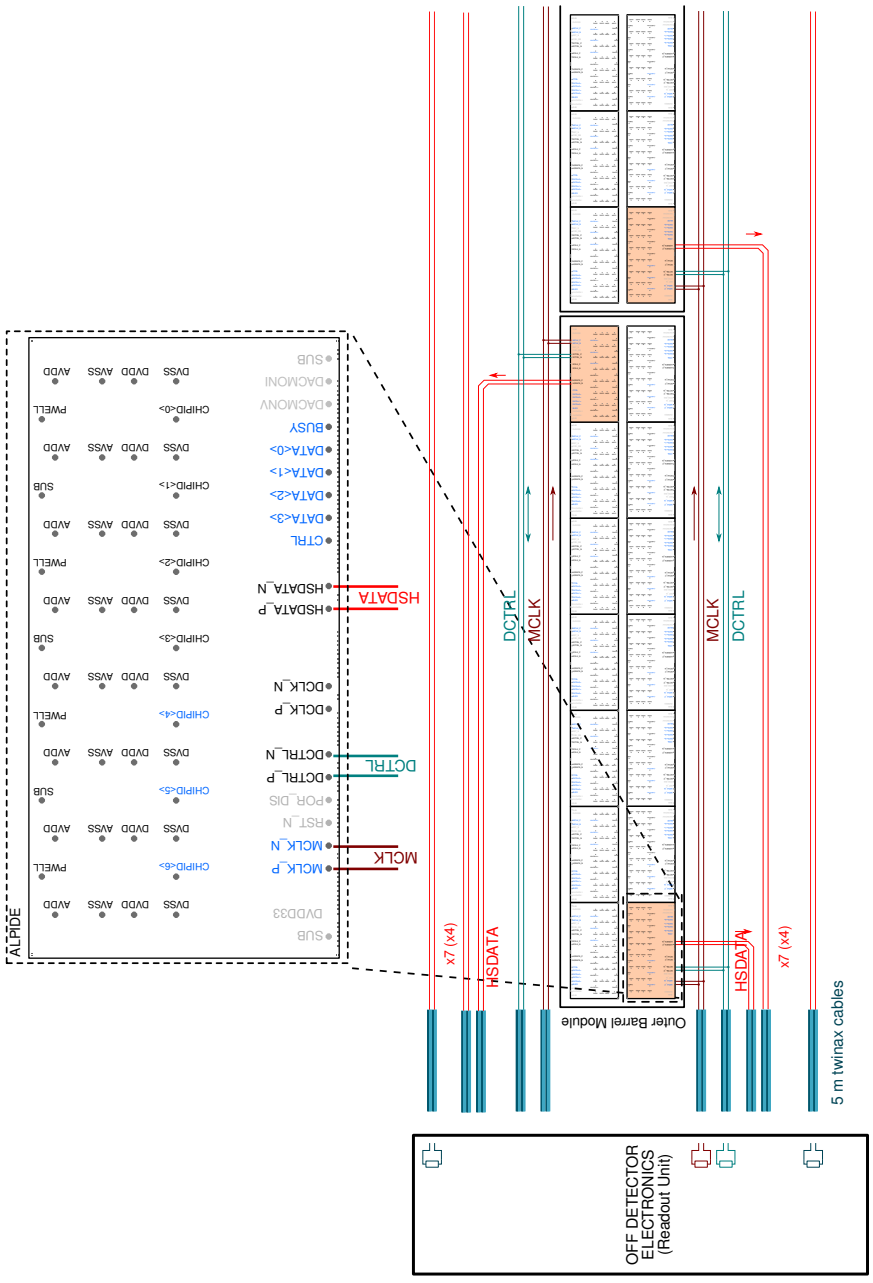

Figure C.1: Schematic diagram of the electrical interconnections between ALICE ITS Upgrade Outer Barrel modules and off-detector electronics.

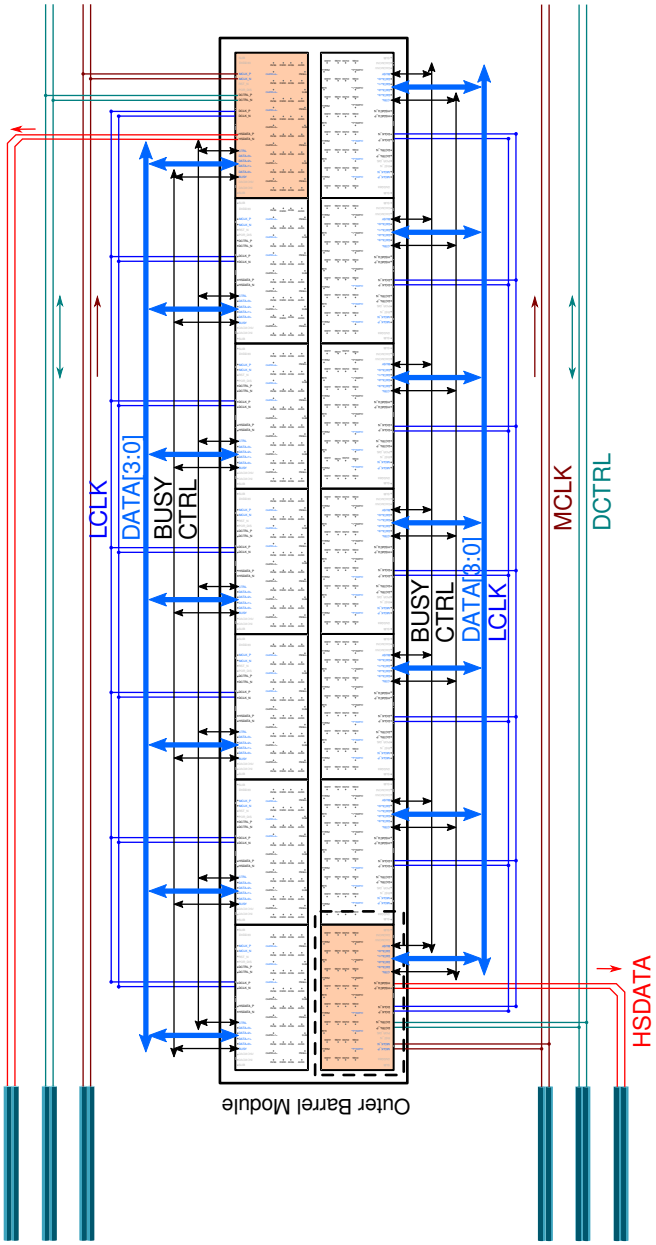

Figure C.2: Schematic diagram of the electrical interconnections between ALICE ITS Upgrade Outer Barrel modules and off-detector electronics.