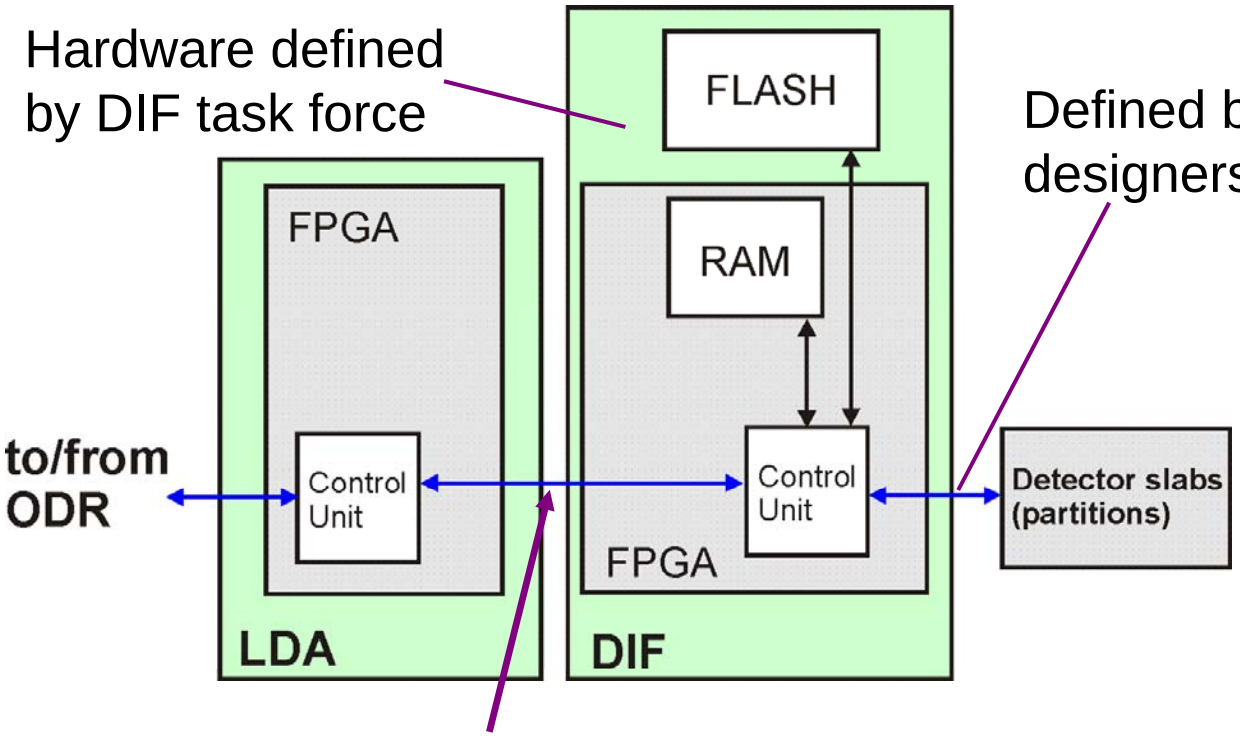
DIF – LDA Command Interface

Julie Prast, Guillaume Vouters, Rémi Cornat,
Bart Hommels, Mathias Reinecke, Frantisek Krivan,

Marc Kelly, Matt Warren

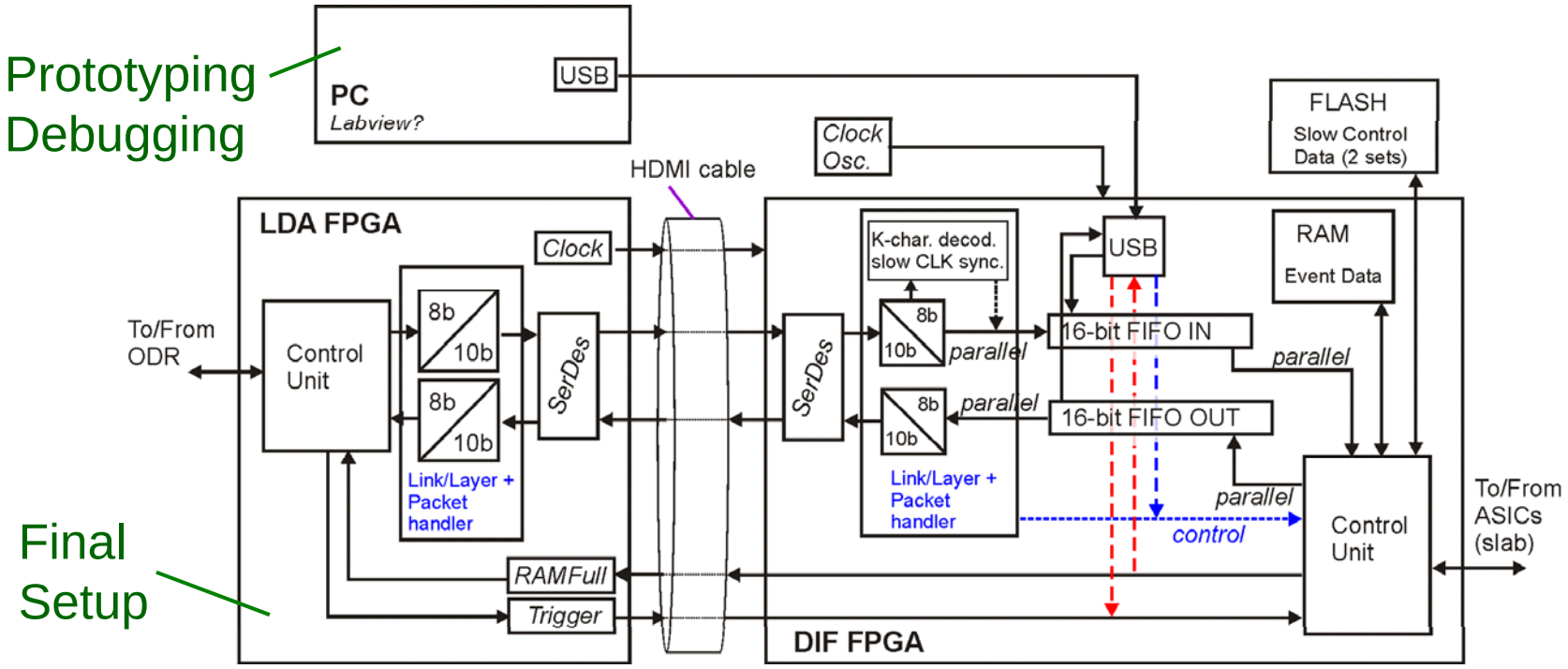
Introduction (DIF interfaces)



How data is sent is defined by DAQ designers,
but **what** is sent (for **DIF/detector operation**) was undefined.

Command definition was necessary, common for DHCAL, ECAL and AHCAL. Also needed for DOOCS development very soon.

Command Interface - Structure



- DIF clock (from LDA): 100MHz (40-120MHz).
- Standard data transfer: 8b/10b channel-coding.
- Trigger/RAMFull: uncoded.

USB interface emulates LDA interface (clock-source: free of choice).

Data / Command Transfer I

Two types of data transfer “frames” are defined between LDA and DIF:

1. Fast Command Frame (16-bit length):

15 … 8	7 … 0
komma character (K)	command word (D)

Used for link synchronization,
timing-critical commands and broadcasts to the DIFs only.

Several Ks and „special sequences“ are predefined. See
http://www.hep.manchester.ac.uk/u/mpkelly/calice/lda/Calice_LDA_Overview.html

FAST Commands for the DIF

not all shown …

FAST Command see section 1.1	komma character	command word D	Operation	Change DIF State?
reset_BCID	K28.3	D1.1	reset BCID	no
start_acquire	K28.3	D2.1 D2.2 D2.3	start data-taking (int. trig) start data-taking (ext. trigger) stop data-taking	yes
#### ECAL specific ####				
	K28.3	D5.0		
#### DHCAL specific ####				
	K28.3	D8.0		
#### AHCAL specific ####				
calibrate	K28.3	D11.1	do a calibration run: with light sys., int. trig	yes
#### DCC identifier ####				
	K28.3	D15.0		

Command list has 5 sections:

General commands, ECAL specific commands, DHCAL specific commands, AHCAL specific commands, DCC identifiers.

Data / Command Transfer II

2. Block transfers (length not fixed):

packettype	pktID	type_modifier (command def.)	data_length	data	CRC
16 bit	16 bit	16 bit	16 bit	data_length *16 bit	16 bit

Used for standard commands and
data transfer (e.g. slow-control, result data) between LDA and DIF

packettype: defines block to be: block data / generic command / command ECAL only / command DHCAL only / command AHCAL only / Command (Block) is for DIF-DIF Link / data is DIF firmware / …
pktID: numeration of sent blocks, used to identify block losses.
type_modifier: command definition. This is the address of the respective command register.
data_length: Number of 16-bit vectors sent in the “data”-section of the block.
data: 16-bit data vectors, e.g. slow-control data for the ASICs
CRC: cyclic redundancy check (look for transmission errors).

BLOCK Transfer Commands

not all shown …

Bock Transfer Name (command) see section 1.2	type_modifier (command def.) 16-bit hex	data 16-bit hex	Operation	Change DIF State?
power_on	0x0002	0x0000 0x0001 0x1000	turn power regulators off turn power regulators on read power register	yes
reset	0x0004	0x0001 0x0002 0x0004 0x1000	reset of DIF reset of slab reset all read reset register	yes
set_DIF_mode	0x0006	0x0001 0x0002 0x1000	set detector into “SLEEP” set detector into “IDLE” read DIF_mode register	yes
power_pulsing	0x0008	0x0001 0x0002 0x0004 0x0008 0x0010 0x0020 0x1000	turn pwr_analog ON turn pwr_digital ON turn pwr_ss/pwr_sca ON turn pwr_adc ON turn pwr_dac ON turn all ON read power_pulsing register	no

Sections: General, ECAL specific, DHCAL specific, AHCAL specific

Command Registers in DIF

For each command (LDA to DIF) the DIF has a dedicated **command register**.
The address of this **command register** is defined

- by the X in the incoming DX.Y command word for FAST Commands
- by the type_modifier for BLOCK Transfer Commands.

Command registers are 16-bit, and can be subdivided for several functional purposes.

15 10	9 5	4	3	2	1	0
Reserved	Status Bits(4:0)	Bit	Bit	Bit	Bit	Bit
R, +0	RC, +10100	RW, +0	RS, +0	RW, +0	RW, +0	RW, +0

Note: R = Readable by the LDA,
W = Writeable by the LDA,
C = Clearable by the LDA,
S = Settable by the LDA,

+x = Value undefined after reset,
+0 = Value is 0 after reset,
+1 = Value is 1 after reset

Command Register Example

power_on, BLOCK Transfer command, address 0x0002

15	1	0
reserved		slab_power
R, +0		RW, +0

Bit no.	Bit Field	Description
15 – 1	reserved	reserved
0	slab power	Switch on or off slab power: slab_power = '1': slab power is on (set by data=0x0001) slab_power = '0': slab power is off (set by data=0x0000)

Table 8: power_on register description

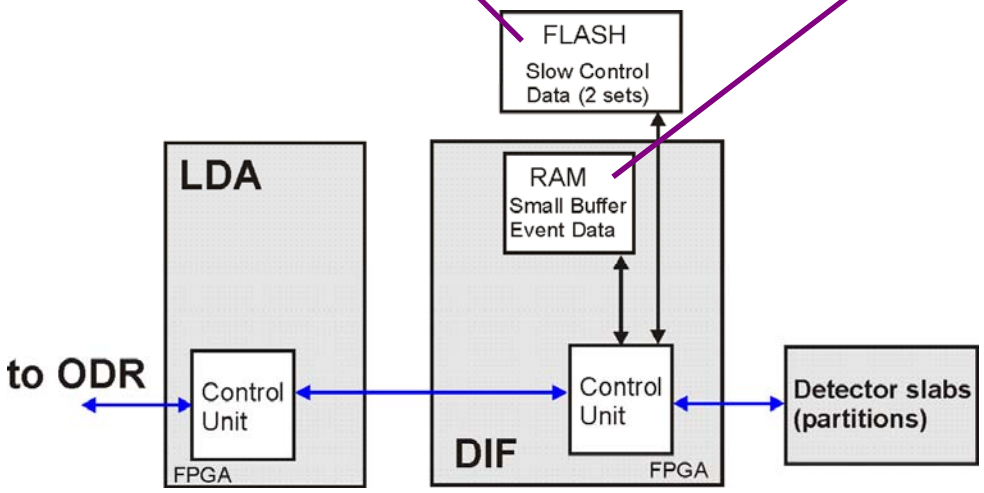
Slow Control / Readout

Two sets of slow-control data
can be stored at the DIF for each
partition (slab) in external Flash memory.

⇒Time-optimized (re-)programming
of slabs (very simple)

RAM in DIF FPGA is too small to store a full set of result data (slab readout)

⇒Readout of slabs and data transfer to LDA **in parallel.**

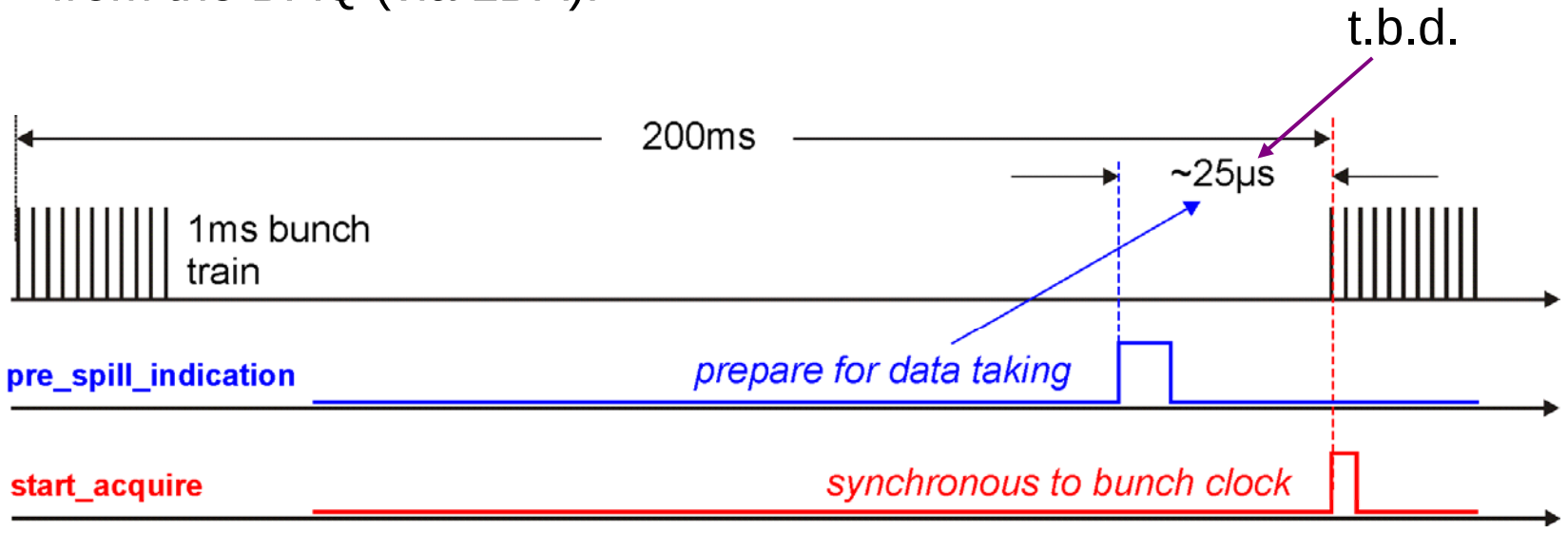


From the DIF task force

Pre-Spill Indication I

DIF needs time to prepare for data-taking (power-on, check configuration, maybe „reset Bunch Counter“, change to ACTIVE mode).

⇒„pre_spill indication“ is sent before the actual „start_acquire“ from the DAQ (via LDA):

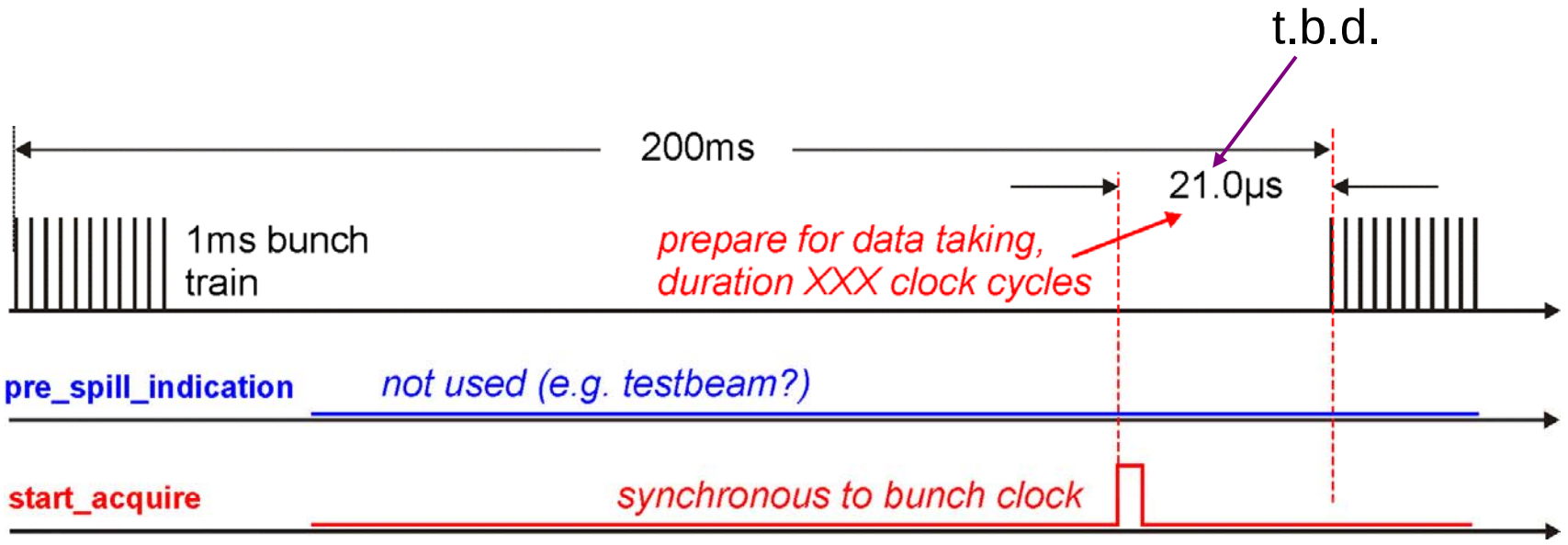


From the DIF task force

Pre-Spill Indication II

If „pre_spill indication“ is not used, „start_acquire“ is sent exactly XXX clock cycles in advance to the bunch trains.

=> Choose operation mode via DIF Control Register



From the DIF task force

Conclusions

- Document available (current version 1.7) defining the command interface LDA-DIF, as well as basic DIF operations. DAQ people greatly supported the development! („Many thanks!!“). See: http://adweb.desy.de/~reinecke/DIF_Firmware_vers1_7.pdf

- Slab- (partition-) interface is already defined by ASIC designers

- DIF hardware specifications have been agreed on in the DIF task force.

- Naming conventions within DIF firmware code under development.

⇒DIF firmware development can take place now in excellent conformity between the detectors DHCAL, ECAL and AHCAL!