



UCL



DAQ Hardware Status

9 September 2008

Matt Warren

*Valeria Bartsch, Veronique Boisvert Maurice
Goodrick, Barry Green, Bart Hommels, Marc Kelly,
Andrzej Misiejuk, Vishal Panchal, Martin
Postranecky, Tao Wu*



Detector Unit: ASICs

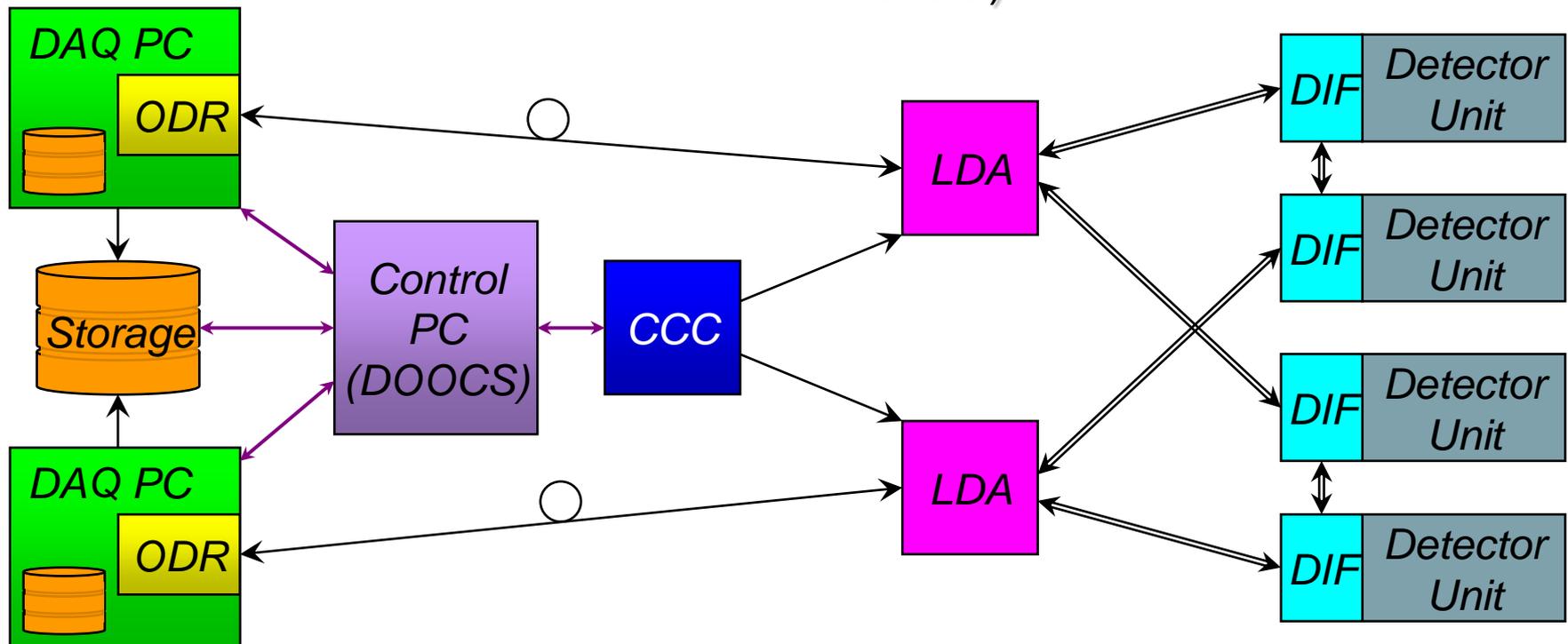
DIF: Detector InterFace connects
Generic DAQ and services

LDA: Link/Data Aggregator – fanout/in
DIFs and drives link to ODR

ODR: Off Detector Receiver – PC
interface for system.

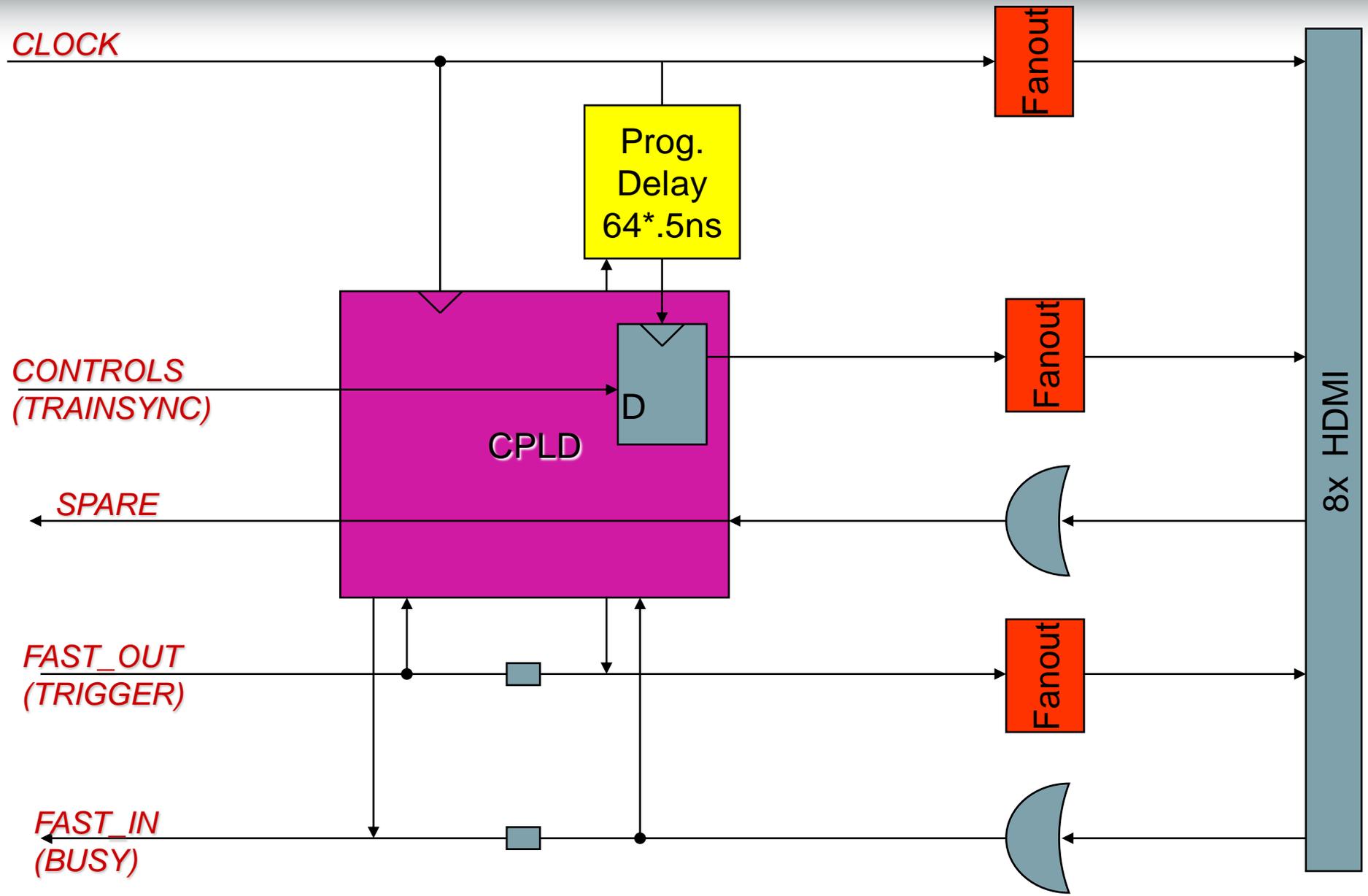
CCC: Clock & Control Card: Fanout to
ODRs (or LDAs)

NEW! CONTROL PC: DOOCS GUI (run-
control)



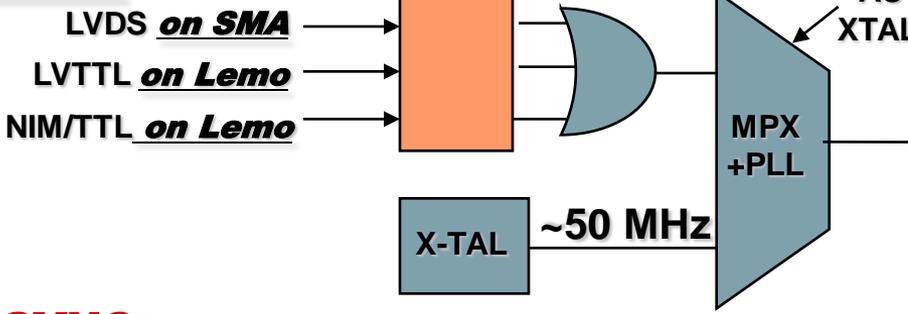
CCC

Overview Schematic



Detailed Overview Schematic

CLOCK



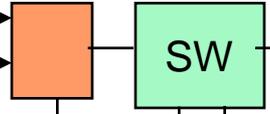
NOTES:

- CPLD + Gates
- Busy == Gen
- Added SPARE line
- Delayed clock option
- 4x 8 way DIL to CPLD

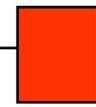
- 8x LVDS *on HDMI*
- 8x LVDS *on SMA*
- 2x LVTTTL *on Lemo*
- 2x NIM *on Lemo*

ASYNC

LVDS *on SMA*
ECL *on 2pin Lemo*



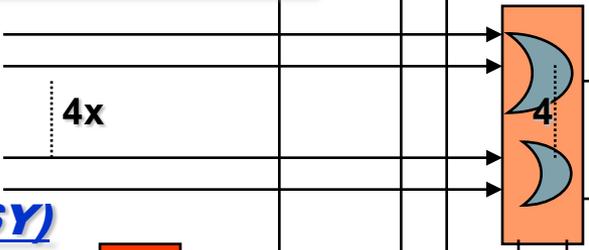
clock



→ 8x LVDS *on HDMI*

Controls (SYNCCMD, BUSY-IN etc)

4x LVDS *on SMA*
4x NIM *on Lemo*



DELAY

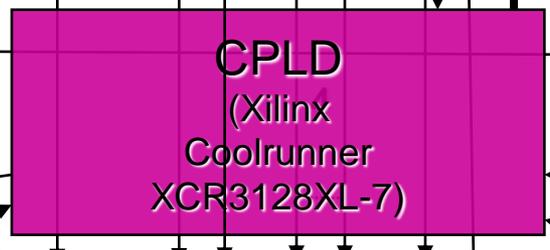
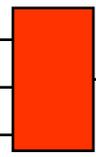
SW-2
5->1



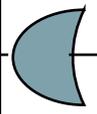
→ 8x LVDS *on HDMI*

GEN (was BUSY)

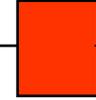
LVDS *on SMA*
NIM *on Lemo*
o/c TTL *on Lemo*



SW-3
2->1



← 8x LVDS *on HDMI*



← 8x LVDS *on HDMI*

SPARE (DATA D2L)

LVTTTL *on Lemo*

RS232

PCB Header

DEBUG 4x 8way DIL

CPLD (XCR3128XL-7) replacing many jumpers and switch logic

RS232 interface as a means of control

Many buffers, 0Ω resistors and solder links for better signal integrity, isolation and configuration

Signal Inputs:

- CLOCK
 - 1x LVDS (SMA DC)
 - 1x LVTTTL DC (Lemo)
 - 1x NIM/TTL (Lemo)
AC/DC
- ASYNC
 - LVDS (SMA) DC
 - ECL (2 pin LEMO) AC
- Controls (SYNCCMD, BUSY etc. + more)
 - 4x LVDS (SMA)
 - 4x NIM/TTL (Lemo)
AC/DC

Signal Outputs:

- CLOCK
 - 2x LVTTTL on Lemo
 - 2x NIM on Lemo
 - 2x LVDS on SMA
 - 8x LVDS on DIL
Header
- TRAINSYNC
 - LVTTTL on Lemo
- GEN (was Busy)
 - LVDS on SMA
 - NIM on Lemo
 - OC-TTL on Lemo
- Spare (DATA_D2L)
 - LVTTTL on Lemo

HDMI I/O: x8

- LVDS AC/DC

OUT:

- CLOCK
- ASYNC
- TRAINSYNC

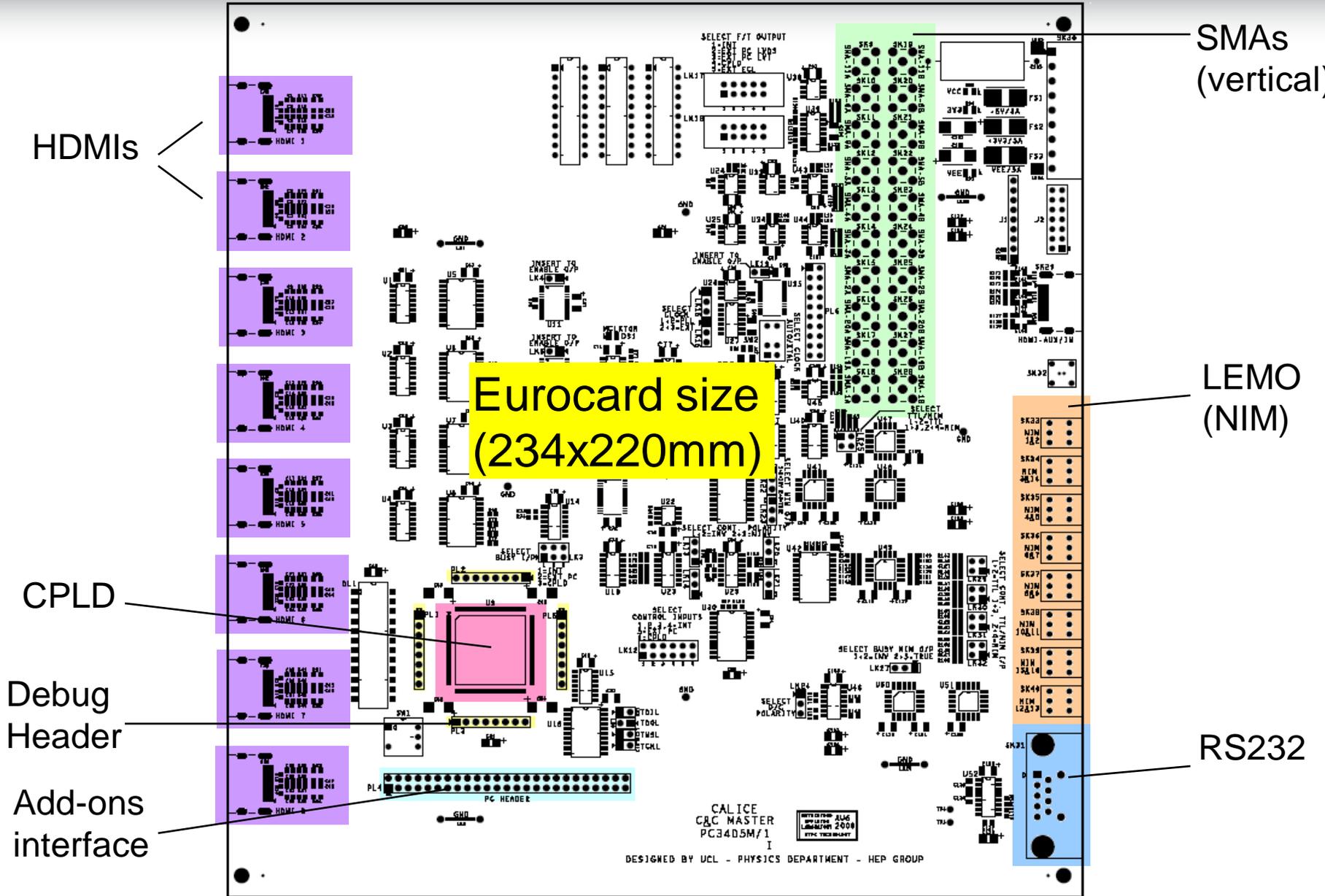
IN:

- GEN (was BUSY)
- SPARE(DATA_D2L)

***NO RJ45**

- Clock:
 - PLL/MUX - *ICS581-02*
 - +/-150 ps jitter
 - 45min/55max Duty Cycle
 - Failover if external clock missing for 3 cycles.
 - Local Osc. $100 \text{ MHz}/2 = 50\%$ duty-cycle 50MHz
- CPLD: *Xilinx CoolRunner XPLA3 XCR3128XL-7*
 - 3.3V, low power
 - 128 macrocells with 3,000 usable gates
 - 5.5ns pin-to-pin logic delays
- Extra IO via IDC header.
- Single PCB with connectors at the edge (big!)
- Separate PSU
- Clock Delay Option to CPLD – $64 \times 0.5\text{ns}$
 - For *signal* deskew (CLOCK unaffected)

Board Layout



- Schematic **DONE**.
- Layout **DONE**.
- Manufacture **IN PROGRESS** .
 - Started last week (1 Sept).
 - Run of 2 with 8 more when satisfied working
 - Manufacturing PCBs for all 10 (2x price = 10x price!)
 - Will do second run of PCBs if any problems found
 - First 2 boards due week of 15 Sept, **BUT** component lead-times may delay a few weeks.
 - Next 8 loaded as soon as testing is complete.
 - Procured components for all 10 now, so won't delay further.
- Next: Firmware development!

Uses same HDMI cable and signal types/direction

CCC HDMI Signals		
<i>CLink Signal</i>	<i>CCC Signal</i>	<i>Function</i>
CLOCK_L2D	CLOCK_OUT	Clock
DATA_L2D	TRAINSINC_OUT	Trainsync signal output
DATA_D2L	Unused	Unused
ASYNC_L2D	FAST_OUT	Asynchronous signal
GEN_D2L	FAST_IN	Busy

- **CLOCK**

- Machine clock (50-100MHz)

- **TRAINSINC_OUT**

- Synchronisation of all the front-end slow clocks.
 - An external signal will be synchronized with/to CLOCK, phase adjusted and transmitted as a single clock-period wide pulse to the LDA.
 - To allow communicating with a stand-alone DIF, the CCC board will can be configured to send the LDA 8b/10b serialised command for train-sync.

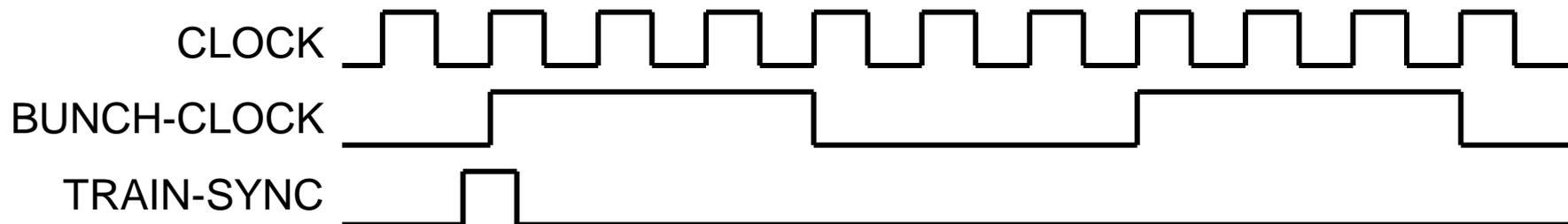
- **FAST_OUT**

- Transfer asynchronous triggers as fast as possible.
 - In AUTO mode, used to Transfer BUSY to detector (toggle = level)

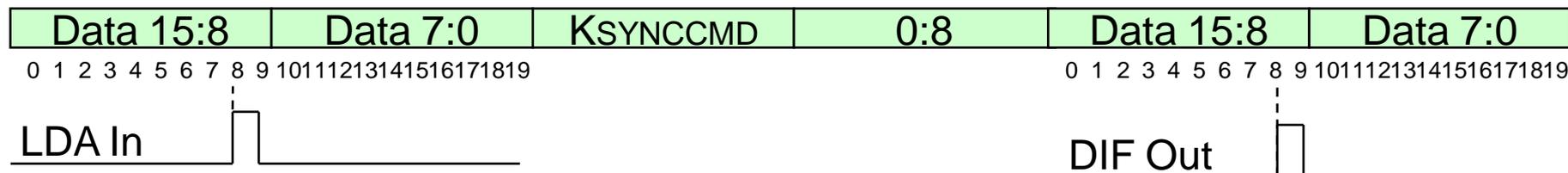
- **FAST_IN**

- Used by DIFs (via LDA) signal to “stop acquisition” when needed.
 - Due to AC coupling the busy must asserted by constantly toggling this line.

- Presume machine CLOCK period $<$ bunch-period
 - Expected to be 50-100MHz, local or machine.
 - Common fanned-out to the detector
- BUNCH-CLOCK (slow clock) derived as $CLOCK/n$
 - Re-produced locally on DIF (with TRAINSYNC & counter)
- Start of train signal (TRAINSYNC) synchronises bunch-clocks on all DIFs.
 - Requires fixed-latency signal – a SYNC CMD.
 - TRAINSYNC “qualifies” CLOCK edge



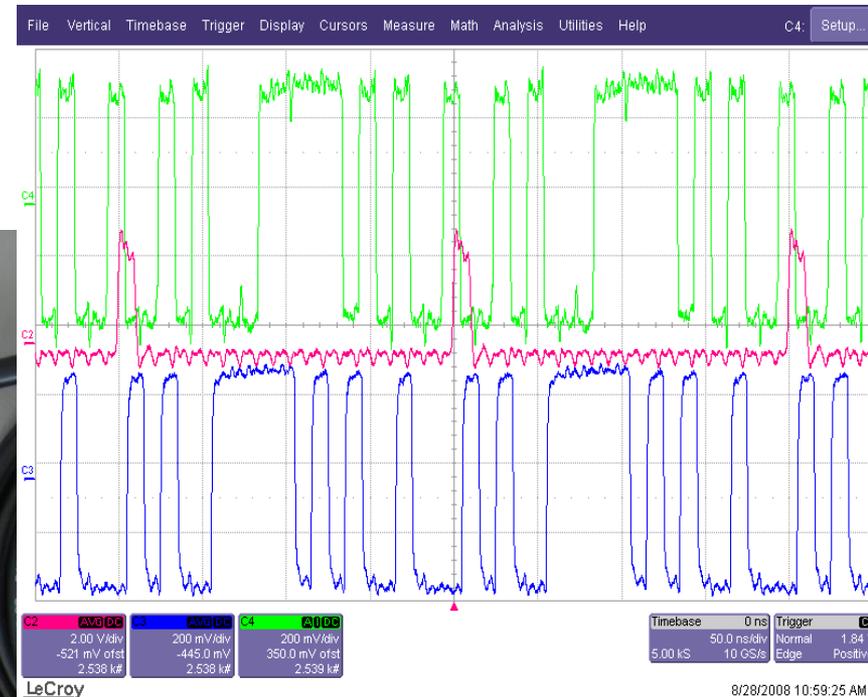
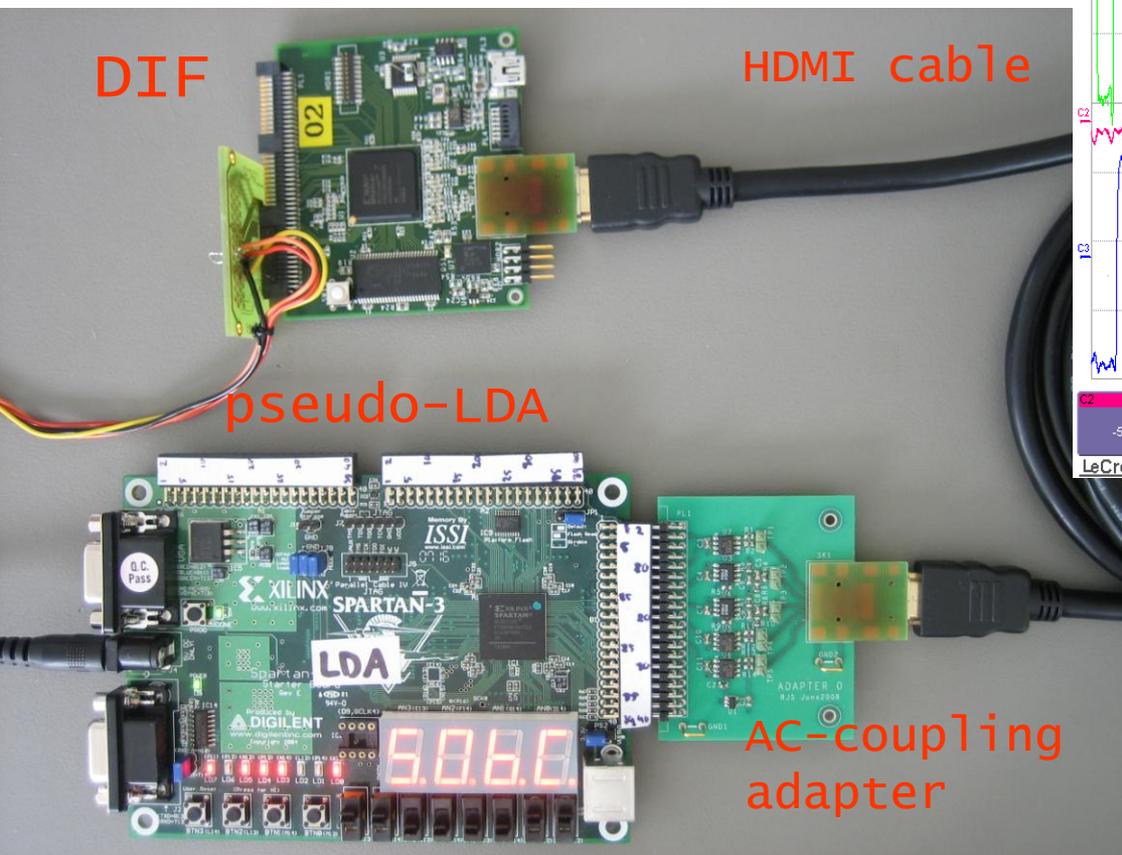
- SYNCCMD is the ONLY mechanism for synchronising DIFs
 - 4 (or 16) types of command are possible.
- Expects a *PRE* bunch-train/spill signal
 - Signal in known phase with BUNCH CLOCK
 - Fixed period prior to first bunch of train
 - Synchronous to CLOCK
- CCC card forwards signal to LDAs
 - Synchronises signal to local clock when needed
- LDA stores arrival time wrt serialiser bit counter.
- Next Word to DIFs replaced with special SYNCCMD word
 - First byte dedicated K character
 - Second byte (7:6): Type; (5:0): Delay (could be 3:5 ratio too)
- SYNCCMD system on DIF delays signal specified number of CLOCKS and issues the required signal.



(generic)
DIF

DIF-LDA link testing

- Link shows signs of life.
 - pseudo-LDA sends CLK & 8B/10B data @ 100MHz over AC-coupled LVDS on HDMI cables



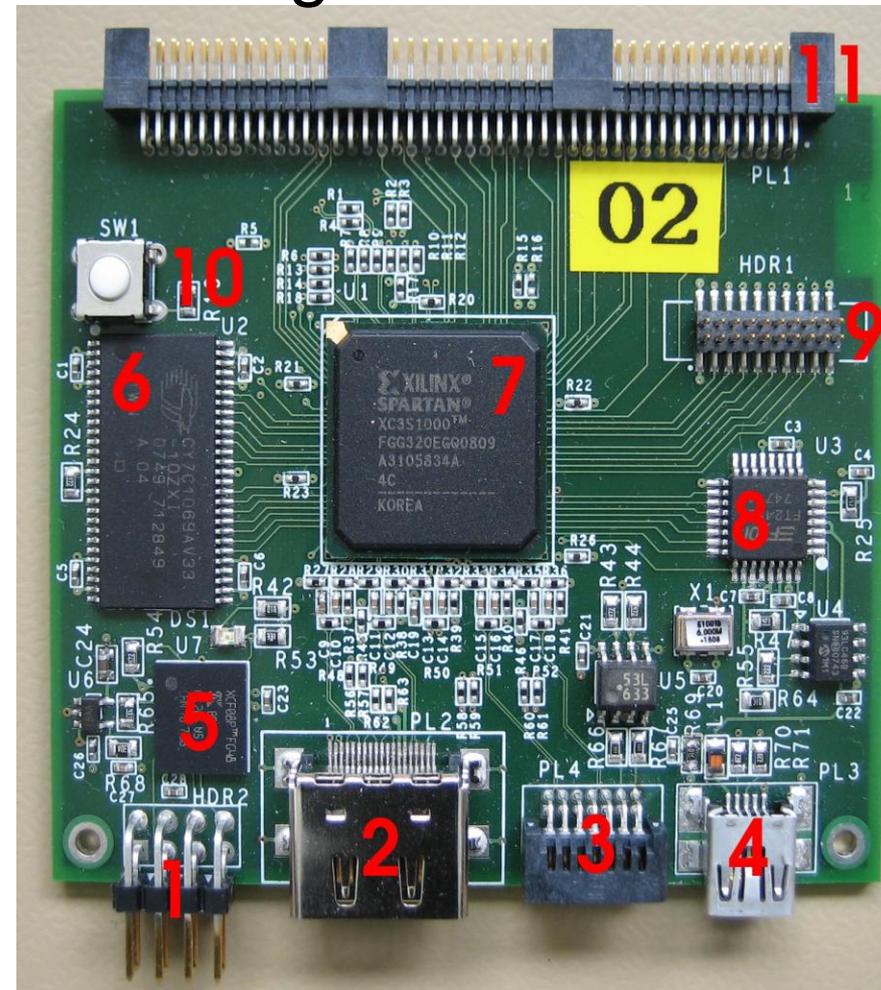
data loop-back in firmware
stable

- Test hardware in place
- Firmware development started
 - LDA Link integration

e.g: ECAL DIF

ECAL DIF prototype: 65x72mm, 8 layers

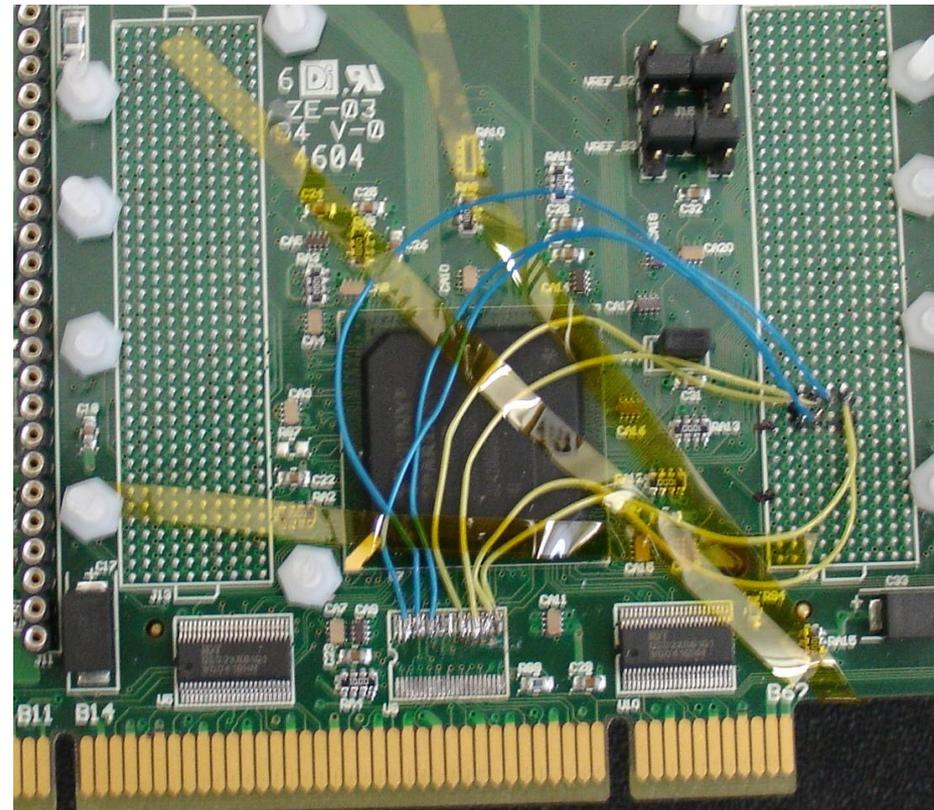
1. JTAG programming header
2. LDA link HDMI connector
3. DIF link connector
4. mini-USB connector
5. Xilinx PROM
6. Cypress 2MB 10ns SRAM
7. Xilinx Spartan3-1000 FPGA
8. FDTI FT245R USB controller
9. 20p user header connector
10. reset pushbutton
11. 90pin SAMTEC IB connector



- 2 DIFs produced, parts available for 10 more.
- DIF hardware is (at least partly) functional

LDA

- Prototype LDA has hardware problems.
 - Mainly incorrect signals routed on PCB to Eth add-on
 - LDA has been modified to attempt fix (see pic!)
 - Ethernet RX OK, auto-negotiation starts OK. BUT ..
 - Ethernet TX appears corrupt
 - random glitches and/or clock recovery problems.
 - PHY in loop-back OK, so looks like the SERDES
 - Investigating ...
- **SOME GOOD NEWS:**
ODR-LDA protocol almost finalised



- Current boards have 8 working HDMI links,
 - Option for 10 on future boards with simple change of FPGA.
- FPGA is basically used as an LVDS transceiver and clock fanout, although it will probably also handle the separate prompt/fast signals to/from the DIF.
- Link documentation is proceeding, large amounts have been already done.



- Enterpoint is designing a replacement board for the BroadDown2 known as the Mulldonoch2.
 - Extra I/O capabilities.
 - EBX format board.
 - PCI connector is replaced by a PC104 connector.
 - SDRAM onboard.
 - SPI flash ram.
 - Better power system
- **Prototype production is expected sometime this month.**
- Is not designed for us especially, but rather is a generic board Enterpoint had planned already. Design time table got shifted when we found the error in the existing BD2 design.
- A corrected BroadDown2 design is also going to be available in roughly the same time frame.

- Need to get Ethernet working, without that we are dead in the water currently.
- Then we can proceed with more of the firmware development and begin to get to a point where it might actually be useful and talk to the outside world.
- Need to consider what the BD2->MD2 migration path means for us, with regard to possible changes to LDA design, to include more features etc.
 - Possible re-spin of the Ethernet Board, to remove the bits we don't want/need and lower cost of it. (MD2 might already have USB built in, and we might decide to skip the TLK2501 all together).
 - Possible extra HDMI interface expansion given the extra IO.
- Bench tests need to be done on the timing/synchronisation system to make sure we can achieve in practice what we thought was possible.

ODR

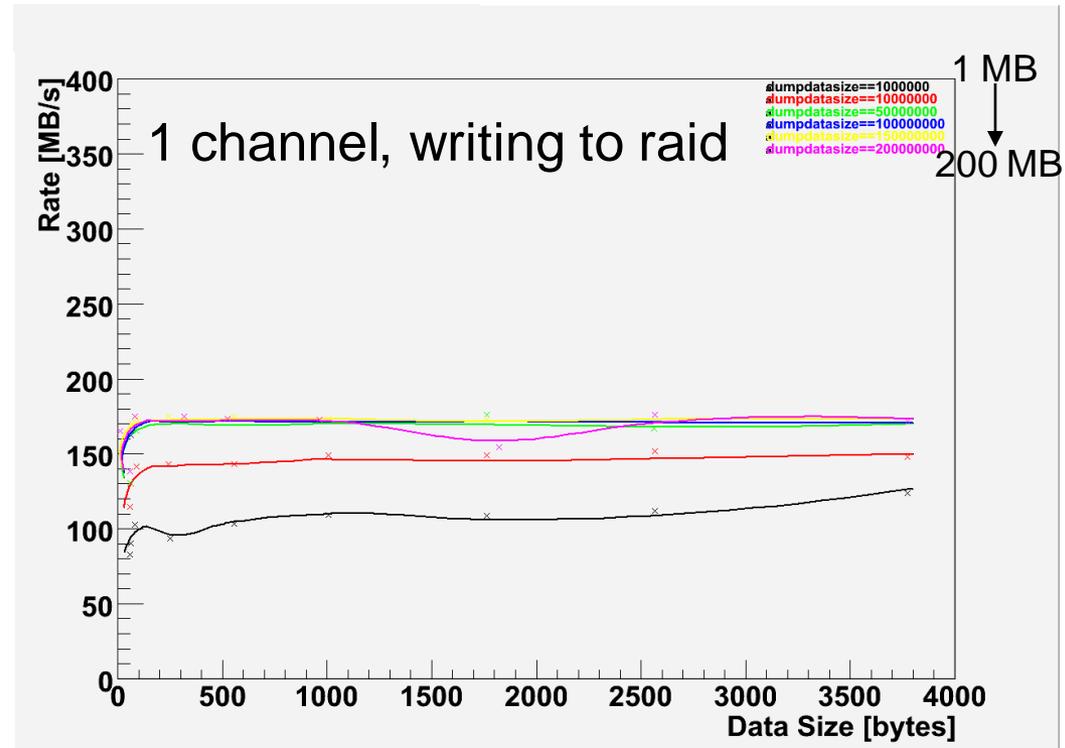
- ODR is working(!)
 - Receive data on 4x fibre (RX),
 - Write to disk FAST (250MB)
 - Send data up fibre (TX)
 - Controlled from Linux driver
- Future upgrade: Decode event header from LDA
 - Provides on-line info
 - Can deal with control messages from LDA
 - Allows host to write to disk without processing



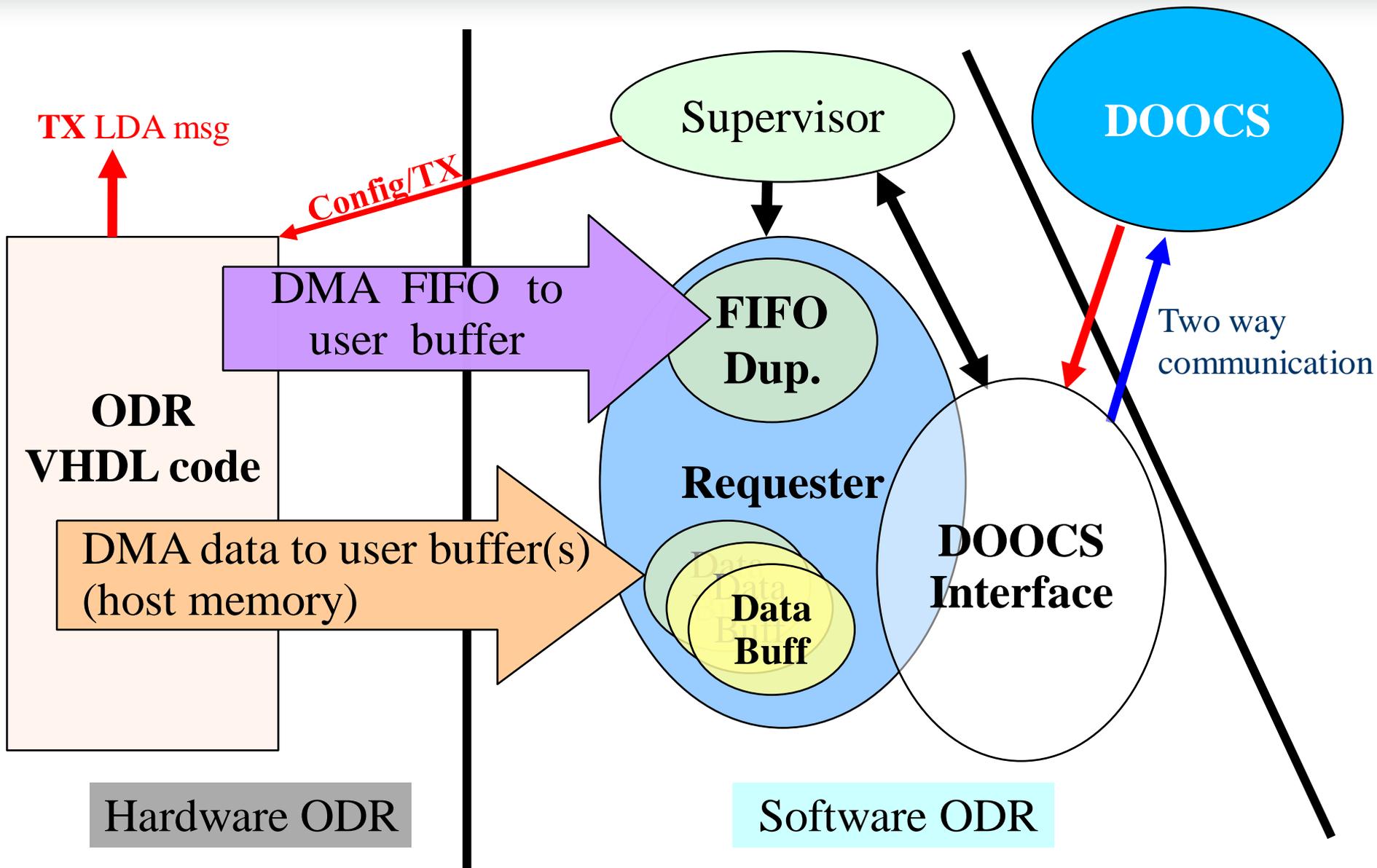
(from ODR to disk)

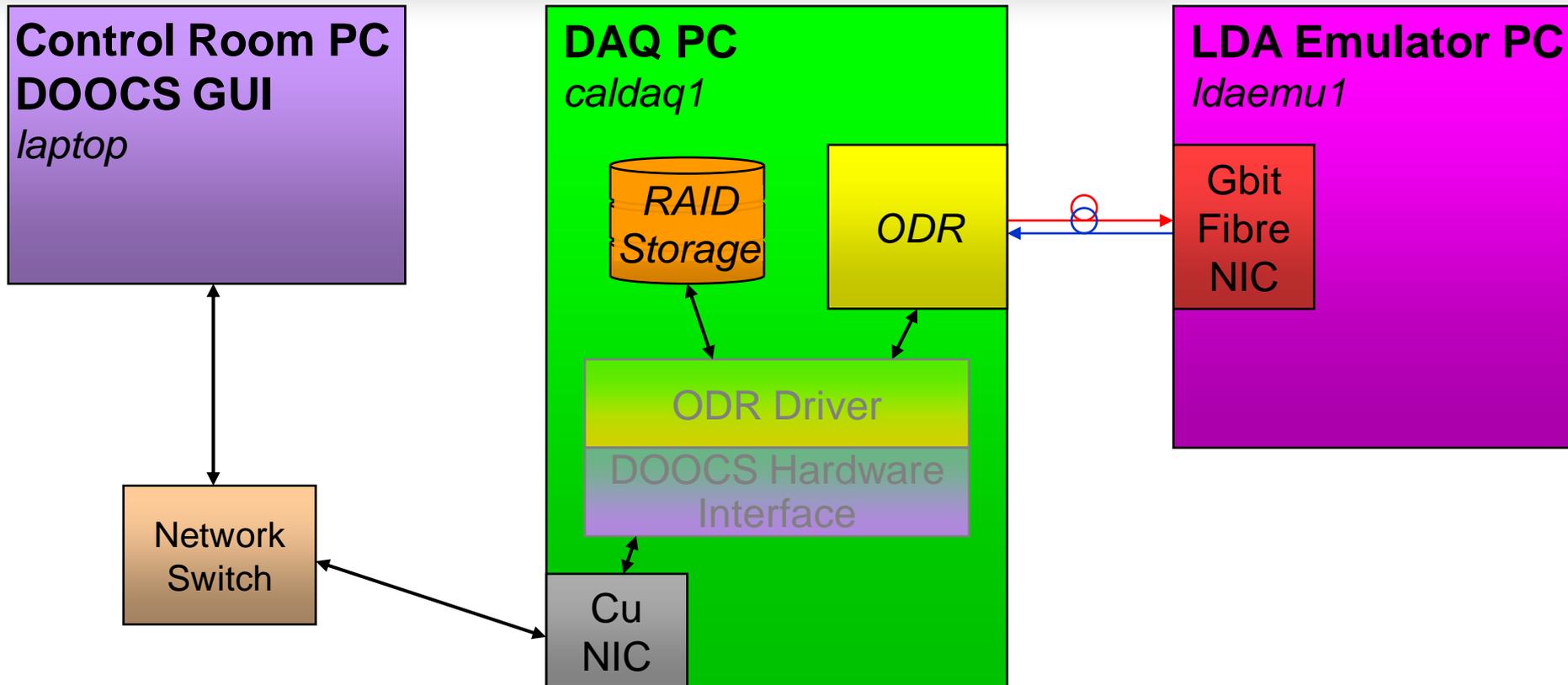
Several factors to optimize:

- Architecture of the host (hyper-threading, raid array disks, kernel version, etc)
- Number of DMA buffers
 - currently using 950
- Number of buffers to fill before dumping the data to disk
 - best to have about N DMA buffers - 200 (so 750 for 950 DMA buffers)
- **Size of files to write** (grouping of data files)
 - Called dumpdatasize on plot
 - Dominant factor
 - chose 100 MB



V. Panchal, V. Boisvert, A. Misiejuk



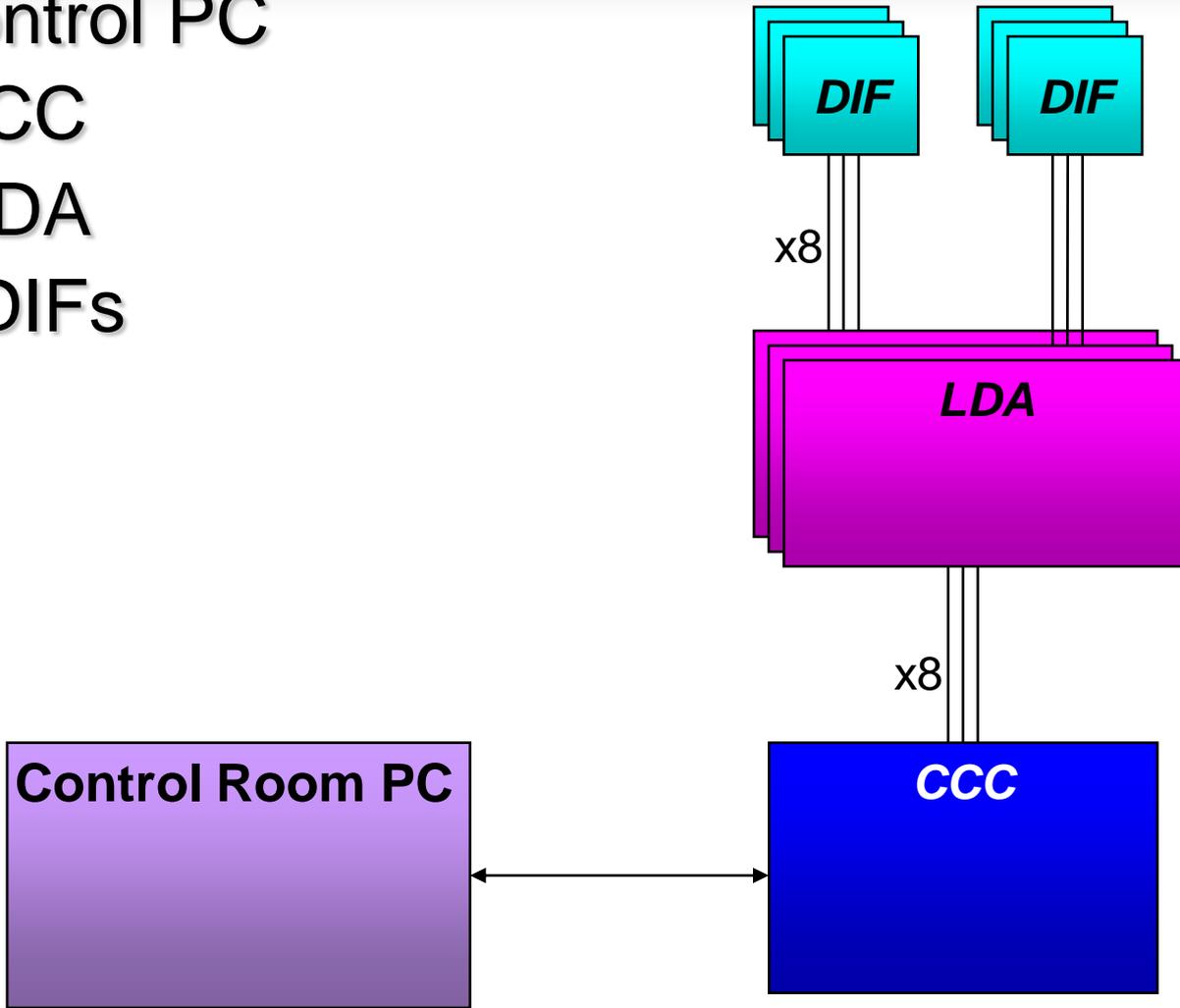


- Control PC signals LDA Emulator (via ODR) to start or stop sending data with command messages
- ODR receives data and writes it to disk.

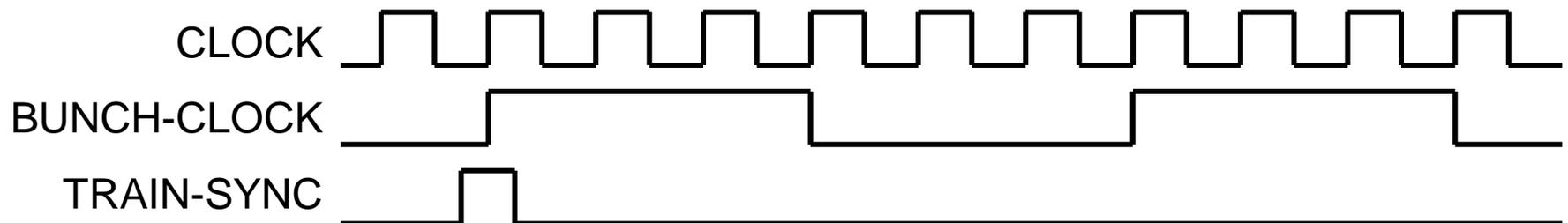
The End

- Goal: A timing system compatible with everyone
(Remi/Vincent/Mathias etc. been asking good questions)
- Looked at details of:
 - Operating modes
 - Trigger
 - Auto
 - Single
 - Machine interface/sync
 - Synchronisation of detector
 - Signalling over our cables
 - Common usage AND SIGNAL NAMING
 - Handling FE errors etc (BUSY)

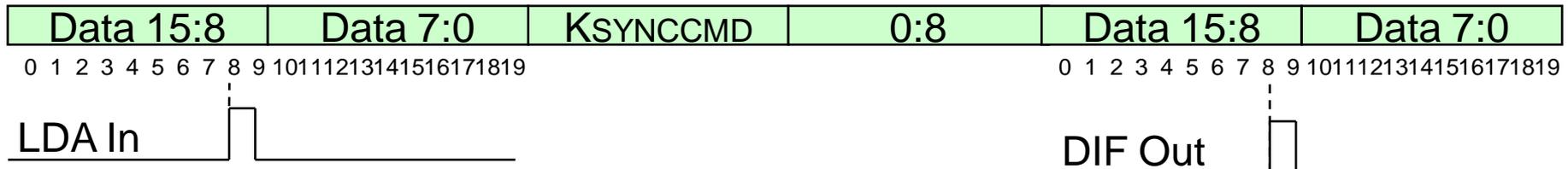
- 1 Control PC
- 1 CCC
- 8 LDA
- 8 DIFs



- Presume machine CLOCK period $<$ bunch-period
 - Expected to be 50-100MHz, local or machine.
 - Common fanned-out to the detector
- BUNCH-CLOCK (slow clock) derived as $CLOCK/n$
 - Re-produced locally on DIF (with TRAINSYNC & counter)
- Start of train signal (TRAINSYNC) synchronises bunch-clocks on all DIFs.
 - Requires fixed-latency signal – a SYNC CMD.
 - TRAINSYNC “qualifies” CLOCK edge



- SYNCCMD is the ONLY mechanism for synchronising DIFs
 - 4 (or 16) types of command are possible.
- Expects a *PRE* bunch-train/spill signal
 - Signal in known phase with BUNCH CLOCK
 - Fixed period prior to first bunch of train
 - Synchronous to CLOCK
- CCC card forwards signal to LDAs
 - Synchronises signal to local clock when needed
- LDA stores arrival time wrt serialiser bit counter.
- Next Word to DIFs replaced with special SYNCCMD word
 - First byte dedicated K character
 - Second byte (7:6): Type; (5:0): Delay (could be 3:5 ratio too)
- SYNCCMD system on DIF delays signal specified number of CLOCKS and issues the required signal.



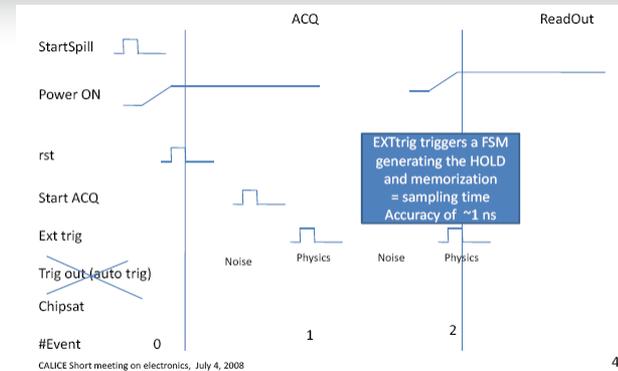
- CCC does NOT support varied delays on individual outputs.
- LDAs can NOT adjust individual link timings
 - Presumed cables to all LDAs equal length
- DIFs need to adjust own timing if needed
 - FPGA resources (or board)
 - Custom cables (available?)
- CCC card can adjust timing of TRAINSYNC wrt CLOCK (1/2 ns steps)

- CCC connects to LDA, DIF and ODR using the ‘standard’ HDMI cabling and connectors and pinout (*CLink*)
 - CCC can be used as a pseudo-LDA for stand-alone DIF testing
- A distinction is made between fast and fixed latency signals:
 - Fast signaling is asynchronous and uses a dedicated line to transfer a pulse. No attempt is made to encode data.
 - Fixed-latency signaling will not arrive fast, but will arrive a known latency after reception by CCC (Jitter 1 CLOCK).

HDMI Signals			
<i>CLink Signal</i>	<i>Direction</i>	<i>Function</i>	<i>Type</i>
CLOCK_L2D	LDA→DIF	Distributed DIF Clock	STP
DATA_L2D	LDA→DIF	Data to DIF (mainly configuration)	STP
DATA_D2L	DIF→LDA	Data from DIF (mainly events)	STP
ASYNC_L2D	LDA→DIF	Asynchronous Trigger	UTP*
GEN_D2L	DIF→LDA	General use	STP

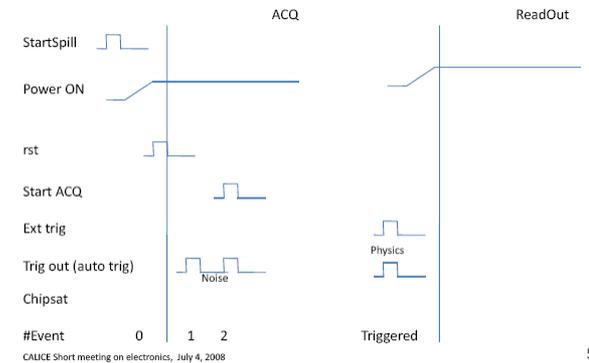
* Twisted pair not guaranteed by HDMI specification but seen in commercial cables

- Triggered
 - External signal causes ASICs to take data

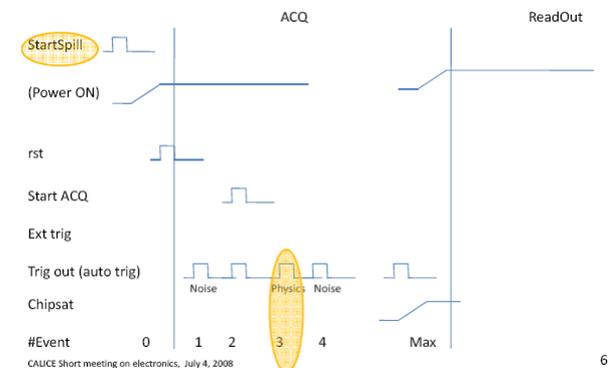


Remi Cornat

- Single (auto-trig)
 - ASICs auto select data, but readout controlled by trigger



- Burst (auto-trig)
 - ASICs signal when full
 - Fed to CCC as BUSY
 - CCC sends stop-acquisition signal to entire detector



CCC HDMI Signals		
<i>CLink Signal</i>	<i>CCC Signal</i>	<i>Function</i>
CLOCK_L2D	CLOCK_OUT	Clock
DATA_L2D	TRAINSINC_OUT	Trainsync signal output
DATA_D2L	Unused	Unused
ASYNC_L2D	FAST_OUT	Asynchronous signal
GEN_D2L	FAST_IN	Busy

- **CLOCK**

- Machine clock (50-100MHz)

- **TRAINSINC_OUT**

- Synchronisation of all the front-end slow clocks.
 - An external signal will be synchronized with/to CLOCK, phase adjusted and transmitted as a single clock-period wide pulse to the LDA.
 - To allow communicating with a stand-alone DIF, the CCC board will can be configured to send the LDA 8b/10b serialised command for train-sync.

- **FAST_OUT**

- Transfer asynchronous triggers as fast as possible.
 - In AUTO mode, used to Transfer BUSY to detector (toggle = level)

- **FAST_IN**

- Used by DIFs (via LDA) signal to “stop acquisition” when needed.
 - Due to AC coupling the busy must asserted by constantly toggling this line.

