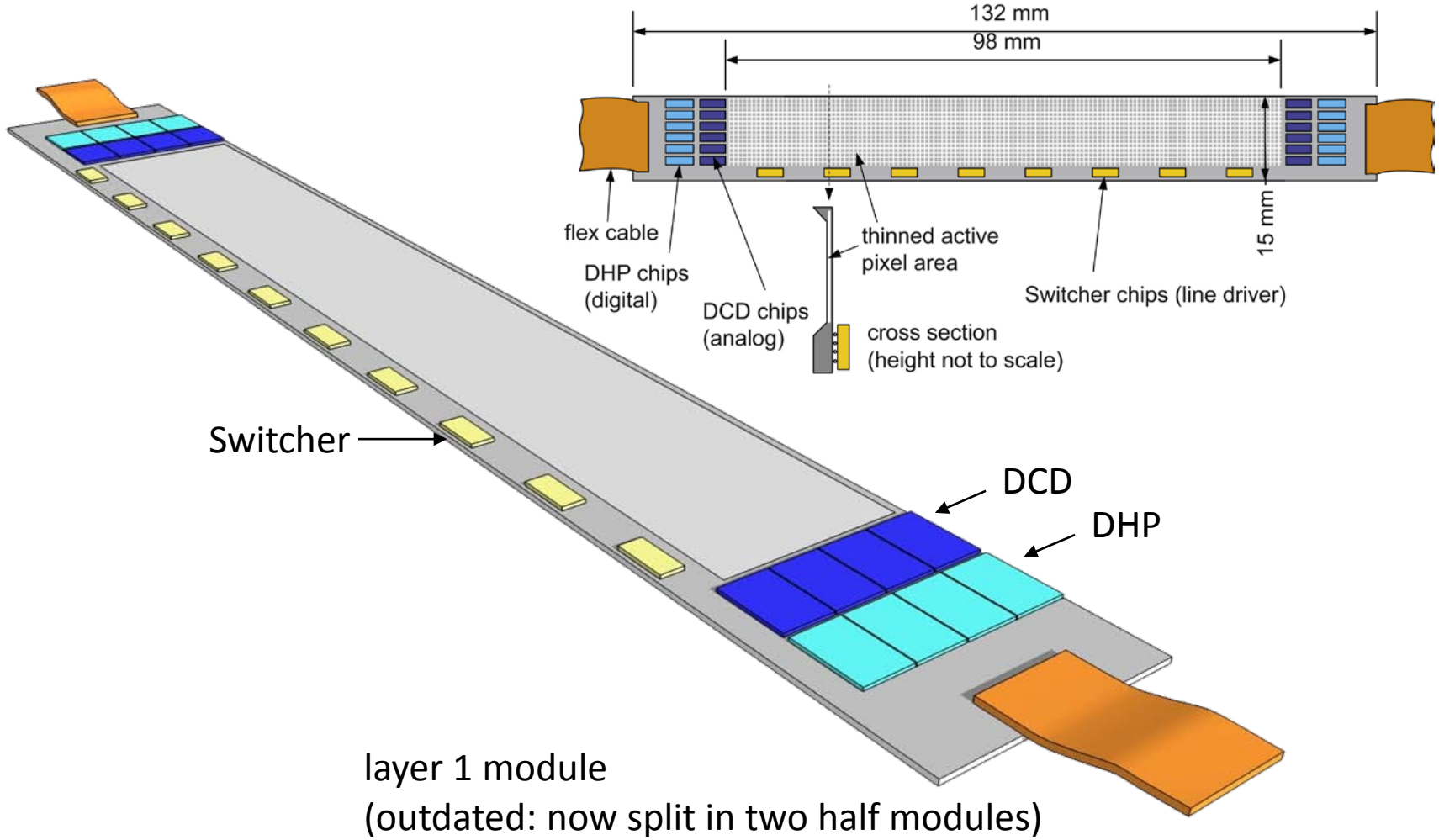
DHP 0.2 Status

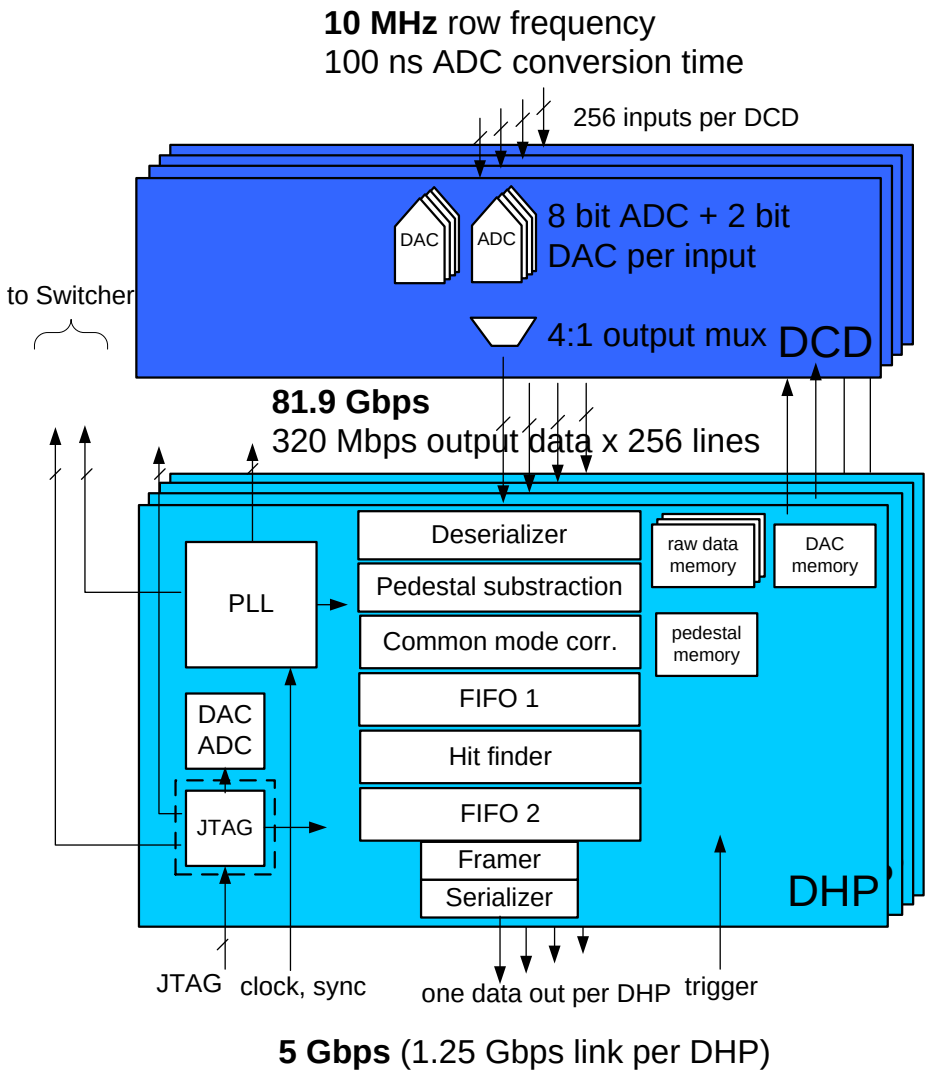
Raimon Casanova (U Barcelona), Mirek Havranek, Tomasz Hemperek, Hans Krüger, Andre Kruth, Ichi Kishishita, Mikhail Lemarenko

Reminder – DHP Function



layer 1 module
(outdated: now split in two half modules)

DHP Function Blocks

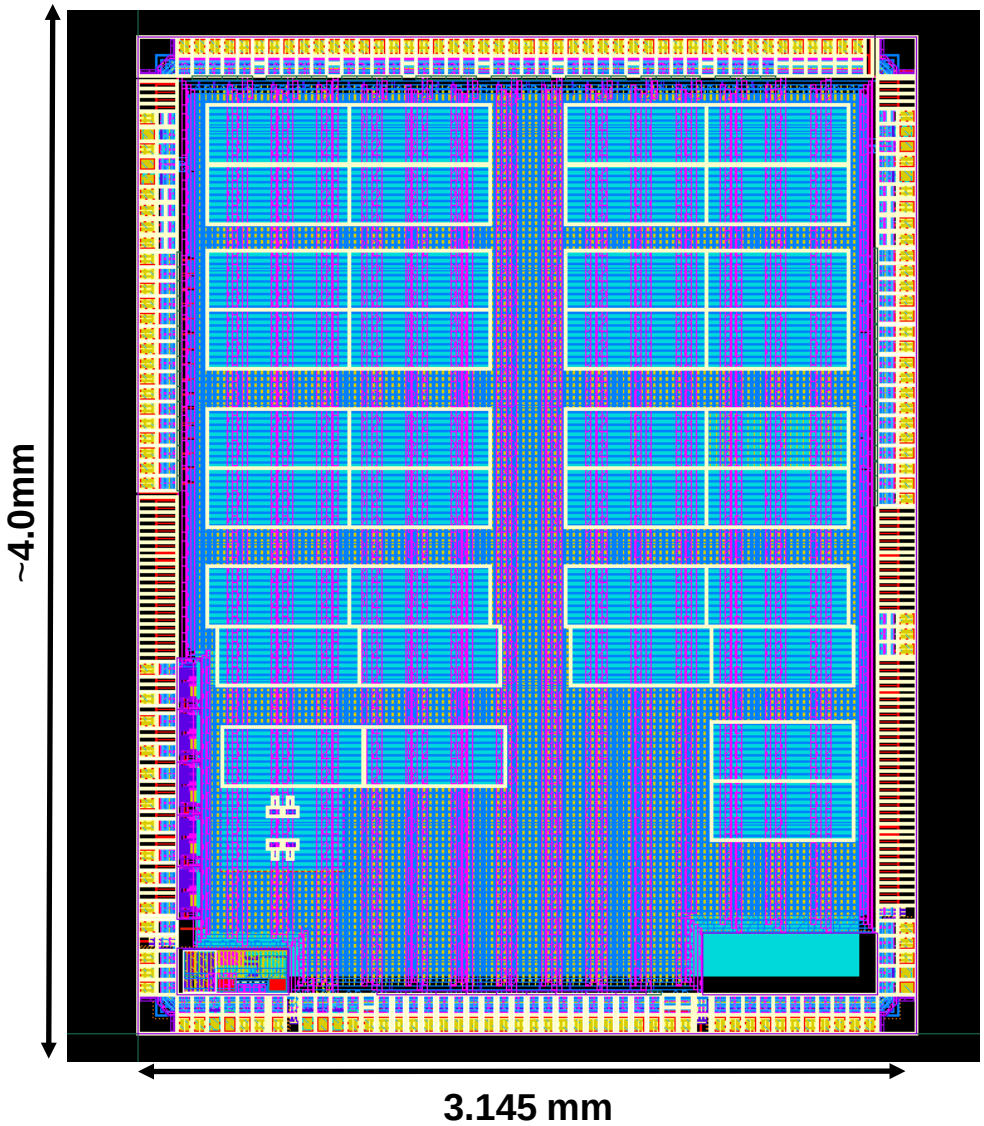


1. Receive and write raw data to memories (continuously)
2. Trigger → read mem and process data
3. Pedestal subtraction
   - Static pedestals (update via JTAG)
4. Two pass common mode correction
   - First pass: all pixels → find hits (biased)
   - Second pass: average w/o hits → 0-supp.
5. Hit finder & de-randomizing FIFOs
   - 64 inputs (FIFO 1)
   - 1 output (FIFO 2)
6. Framer
   - Data formating
   - AURORA protocol
7. Serializer
   - 20:1 mux
   - CML driver with pre-emphasis

DHP 0.2 Main Features

- 64 channels (**full size chip**)
- C4 bump bonds
- Chip size ~ 3.2 mm x 4.0 mm
- Full implementation of data processing & r/o modes
  - Static pedestal compensation (JTAG update)
  - 2 pass common mode correction
  - Fast last frame raw data transfer
- Improved data format (up to 26% bandwidth reduction)
- Switcher sequencer
- On-chip bias DACs and temperature sensor (U Barcelona)
- Gbit link with programmable pre-emphasis

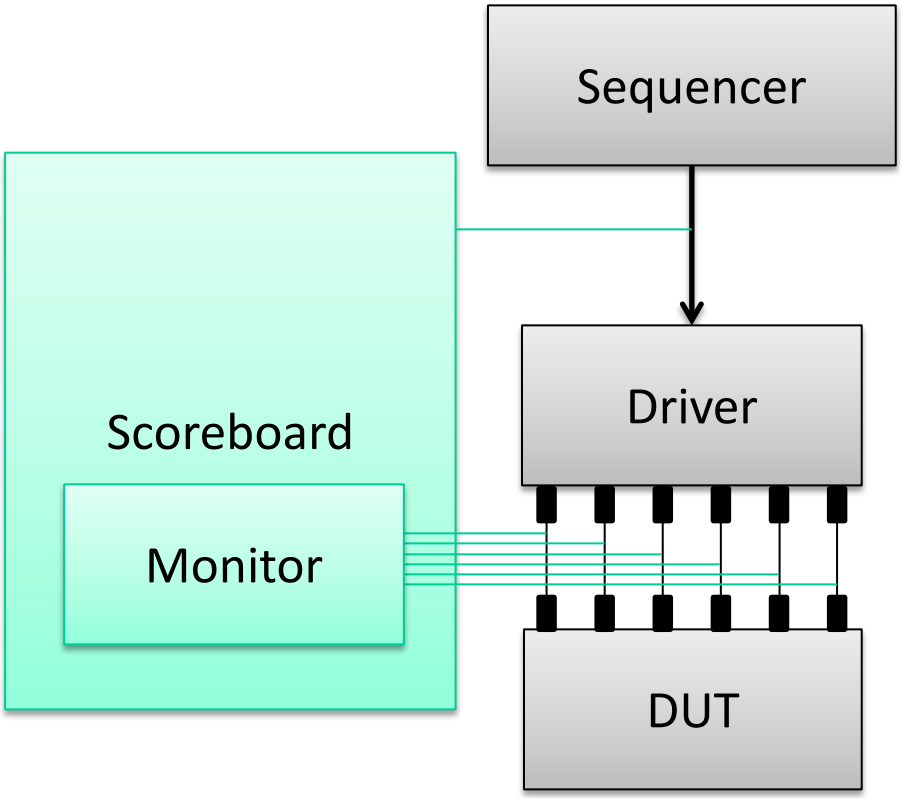
DHP 0.2 Floorplan



- Memory blocks (2048 rows equivalent)
  - Raw data
  - Offset data for 2 bit DAC
- Data processing core
- Output FIFO
- Serializer
  - PLL
  - 20:1 mux
  - CML driver (pre-emphasis)
- Analog blocks
  - 11 bias DACs (8 bit)
  - ADC (10 bit)
  - temp. measurement

DHP 0.2 Verification

- System Verilog OVM (open verification methods library)



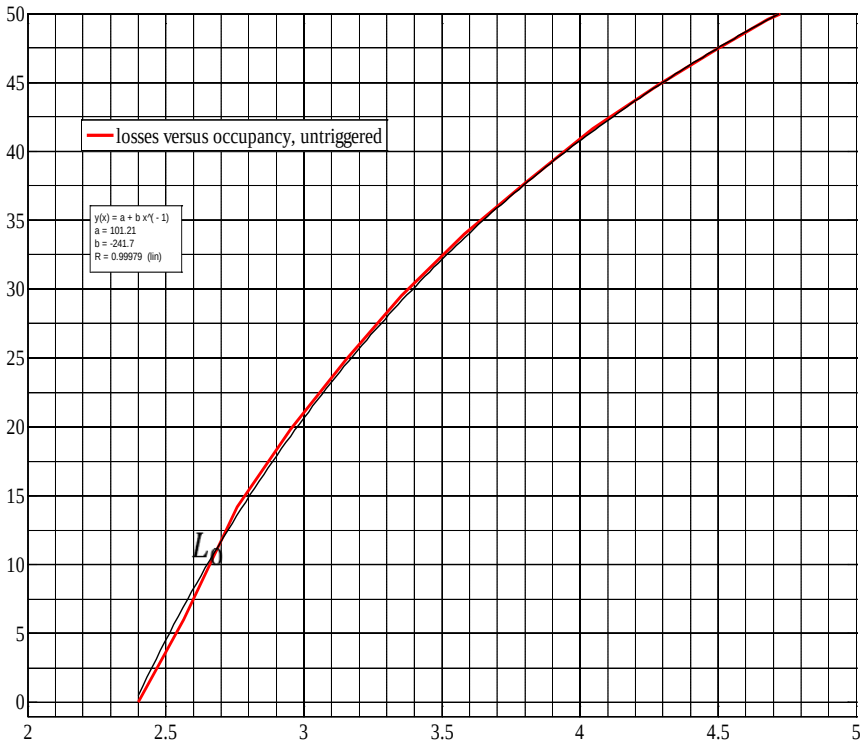
Verification Status

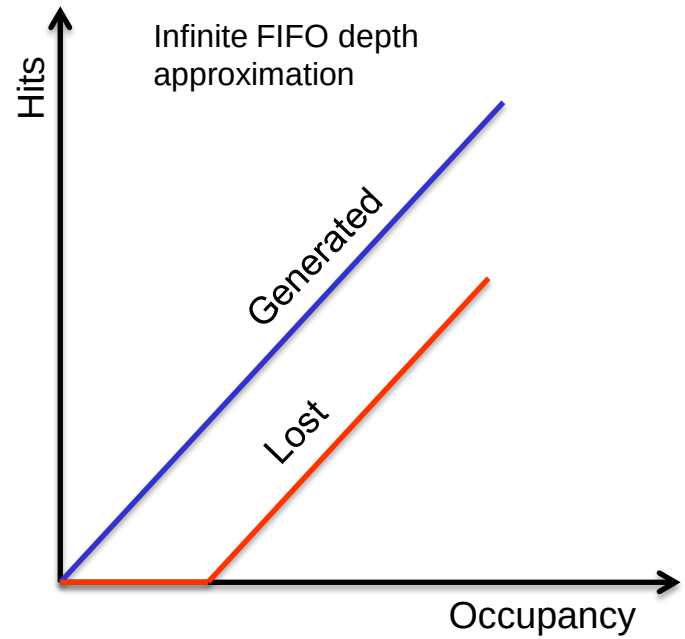
Task	Date	Owner	test name to launch	Notes	Status
number representation (check all possible values					
row data record and send		Tomasz	vsim_raw_data_test	?	OK?
pedestal update			vsim_pixel_masking_test	it was proven to be able to change the pedestal value through the JTAG	OK
common mode algorythm (+ overflow)	4/12/2011	Mikhail		some issue with desyncronization	issue corrected
	4/14/2011	Mikhail		CM work properly	OK
pedestal subtraction (+overflows)	4/11/2011	Mikhail	vsim_pedestal_test	looks ok. After loading the pedestals through the ACQUISITION_TO_MEM pedestals are correctly subtracted	OK
hit recognition	4/15/2011	Mikhail	vsim_pedestal_test	no problem	OK
hit recognition	4/1/2011	Mikhail	vsim_pedestal_test		OK
losses test	4/1/2011	Mikhail	vsim_occupancy_test	up to 2.4%	OK
offset compensation (shifting)					ongoing...
sequencer					
PRBS pattern					
configuration (JTAG)	4/12/2011	Mikhail	vsim_occupancy_test	ACQUISITION, ACQUISITION TO MEM, NOP	OK
read-out JTAG losses counter	4/26/2011	Mikhail	vsim_error_counter_test		OK
disable channel (255 in pedestals) -pixel masking test	4/28/2011	Mikhail	vsim_pixel_masking_test	in principle works... but the masking is not trivial	OK
disable memory					? what is that?
trigger latency test	4/26/2011	Mikhail	vsim_trigger_test		OK
Output Framing:					
- different frame size (1,2,3,4, max etc.)		Tomasz		max size set to 792, works correctly	OK
Data/Fifo overflow(for which opccupancy, should correspond to the simulated one)	4/12/2011	Mikhail	vsim_pedestals_test vsim_occupancy_test	works properly	OK
Test on random data	4/1/2011	Mikhail	vsim_pedestal_test	works up to 1.2% untriggered, then mixes up frames	issue corrected
	4/12/2011	Mikhail		works up to 2.4% untriggered. Frames fixing up fixed	OK
include random pedestals	4/11/2011	Mikhail	vsim_pedestals_test	pedestals working	OK
include random pedestals	40644	Mikhail	vsim_pedestals_test	works but some bugs observed	issue corrected
	4/12/2011	Mikhail		works properly	OK
trigger test	4/19/2011	Mikhail	vsim_trigger_test	works properly. trigger[9:0] - refers to previous frame due to latency, of the latter is set to 4(standard setting)	OK
random data triggered FULL TEST!!!!	4/28/2011	Mikhail	vsim_full_triggered_test	work up to 4 % even better !!!!	OK
output synchronization/post route					
CDC Check					
half output speed					

Verification Summary Snapshot

• JTAG configuration	works
• Aurora	works
• Common mode bock	works
• Hit finder	works
• Pedestal subtraction	works
• Pedestal update through JTAG	works
• Trigger latency	works
• Raw data send-readout	works?
• Pixel masking	works
• Random data un-triggered losses test	works
• Random data triggered losses test	works

DHP 0.2 Data Inefficiency (un-triggered r/o)





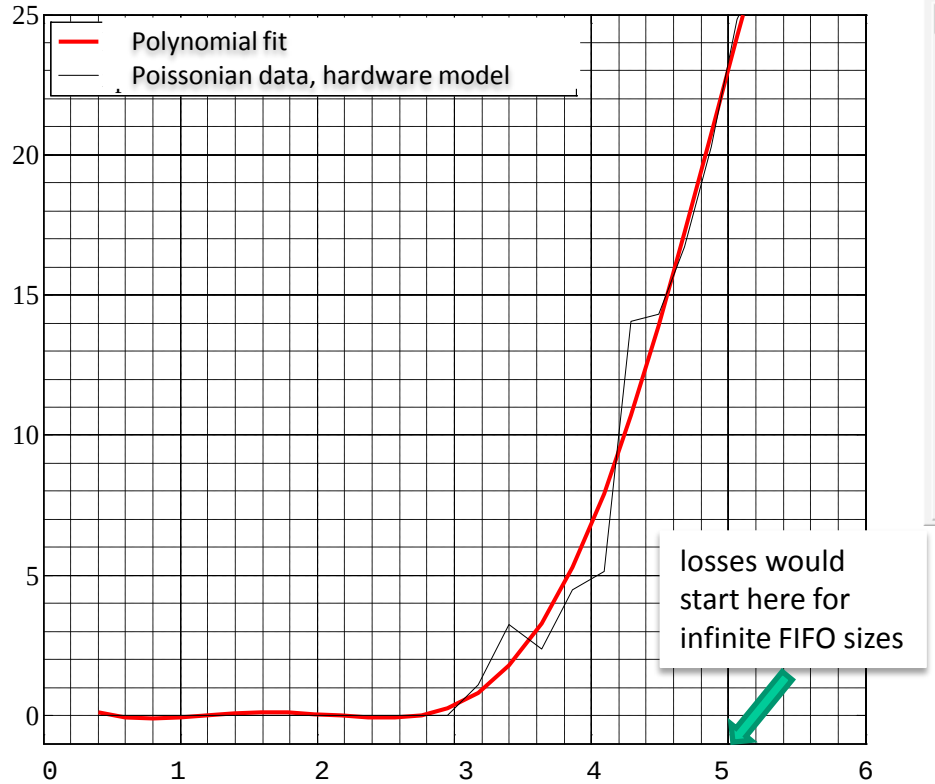
$$inefficiency = \frac{data\_lost}{data\_generated} = \frac{x - L_0}{x} = 1 - \frac{L_0}{x}$$

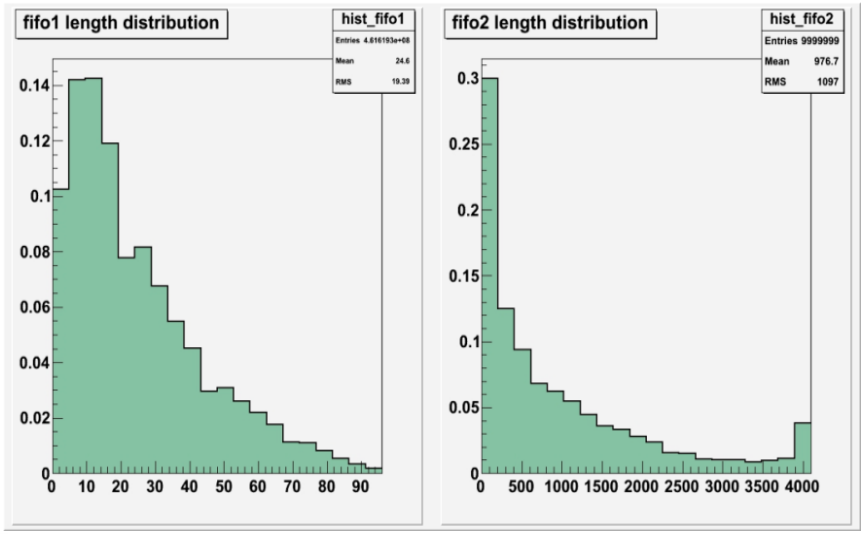
$$L_0 = 0.024$$

FIFO depths (as in DHP 0.2)

- Fifo1: 16
- Fifo2: 512

DHP 0.2 Data Inefficiency (30 kHz trigger)





FIFO fill level distributions for 5% occupancy

FIFO depths (as in DHP 0.2)

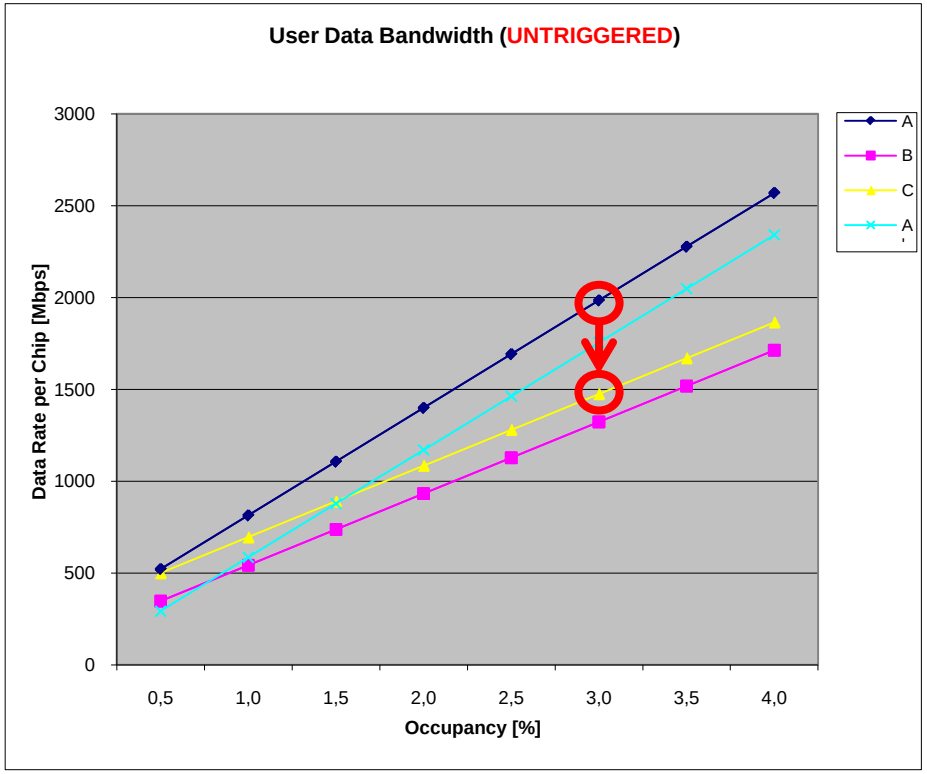
- Fifo1: 16
- Fifo2: 512

DHP 0.2 Data Format

- generic 24 bit hit data format: <10b row | 6b col | 8b ADC>
  - 10 bit row address: 768 pixels
  - 6 bit column address: 64 pixels
  - 8 bit ADC range: 0..255
- want to add common mode value once per 256 pixels (every four rows)
  - send c.m. as data word (keep 24 bit alignment) ➔ waste of bandwidth
  - alternative: store c.m. values for one frame and send the data in a special block ➔ makes the data path more complicated
- proposal: “row header” format
  - send row address only once and send corresponding data words with column address and ADC value only
  - 16 bit length for cm and data words
    - row header (including c.m.): < 10b row | 6b cm>
    - data word: <6b col | 8b ADC | 2b not used>
  - disadvantage: one cannot distinguish row headers form data words by their structure (maybe heuristics can do)

DHP 0.2 Data Format

- reorder addresses bits to define flags for bits row headers and data words
  - row header: <row flag = 0 | 9b row | 6b cm>
  - data word: <data flag = 1 |7b col | 8b ADC>



A: generic 24b data (+ 24b c.m. word per row)
B: 10b/6b row/col addresses
C: reordered 9b/7b row/col addresses
A': A without c.m. word

➔ data reduction of 26% @ 3% occ. for a change from A to C

DHP 0.2 Data Format

- Frame Header *SOF* (32 bit): <data type | flags| frame ID>
  - data type (3 bit): [raw data or 0-supp. data]
  - flags, not used yet (13 bit)
  - frame ID (16 bit)
  - ➔ will always be send at start of a new frame, independent of trigger
- Row Header *SOR* (16 bit): <flag = 0 | row address | common mode>
  - flag (1 bit): 0 → row header
  - row address (9 bit)
  - common mode (6 bit)
  - ➔ will only be send if hit data for the active row is available
- Data Word *DW* (16 bit): <flag = 1 | column address | ADC>
  - flag (1 bit): 1 → pixel data
  - column address (7 bit)
  - ADC (8 bit)

DHP 0.2 Data Format

- Typical data stream:

<SOF n><SOR m><DW i><DW i+x> … <SOR m+y><DW j><DW j+y> …
<SOF n+1> …

- SOF will be send for every frame independent of trigger

- SOR will be send only if corresponding row contains hits

- Data will be sorted
  - Frame wise
  - Row wise
  - Column wise

- Data form different DHPs will not be aligned
  - Data per DHP (3% occupancy) ≈4 kByte

DHP 0.2 Data Format (orig. Verilog code)

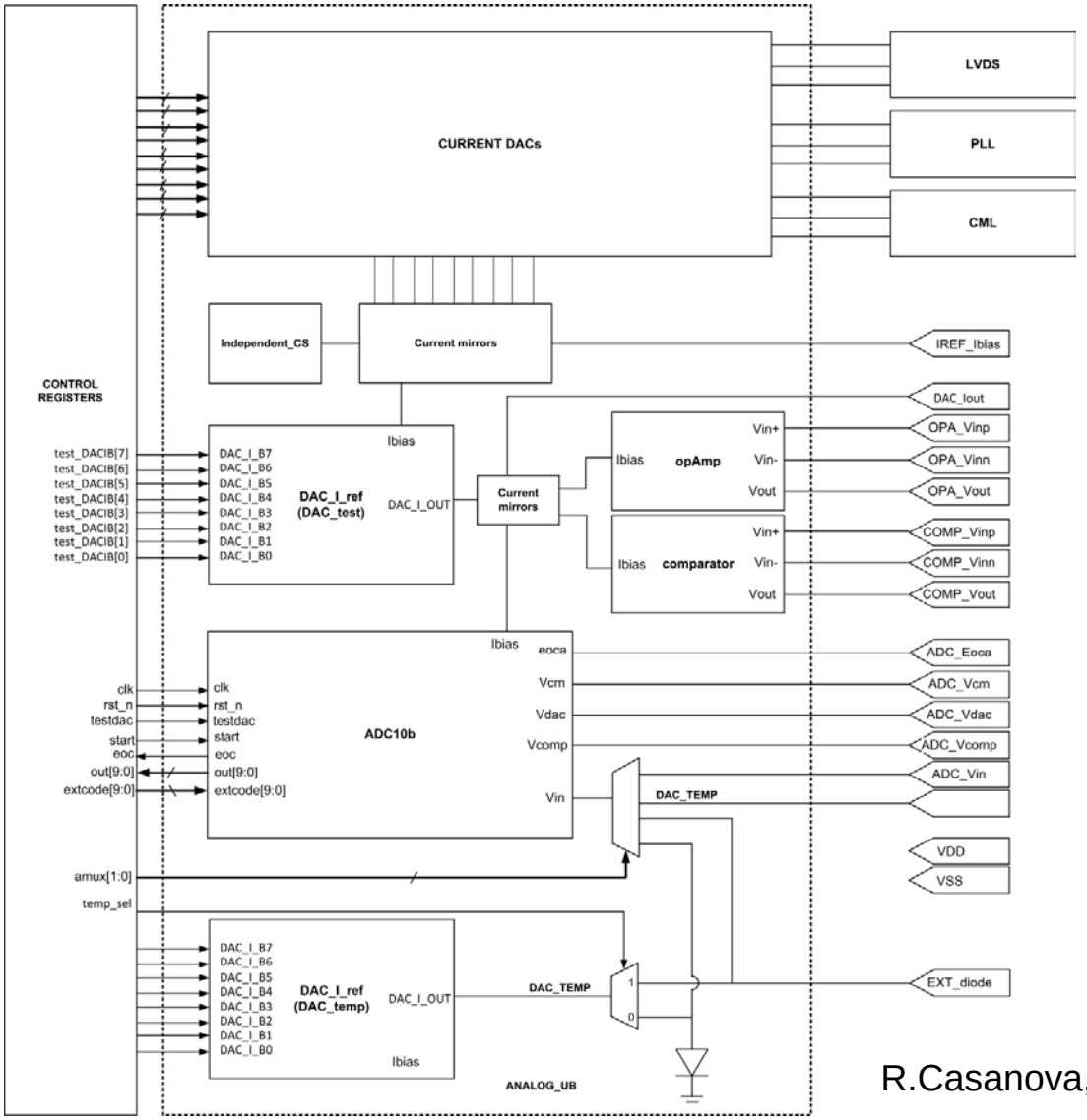
```
// data type flags
typedef enum bit [2:0] {
    RX_PAIR_DATA_S = 3'b100, // not used
    RX_ROW_SINGLE = 3'b101,  // 0-supp. data
    RX_RAW_DATA = 3'b000     // raw data
} t_rx_data_type;

// row header and data word flag
typedef enum bit {
   RX_ROW_SINGLE_HEADER=1'b0,  // row header
   RX_ROW_SINGLE_DATA=1'b1}    // data word
   t_rx_single_type_word;


// start of frame header SOF
typedef struct packed {
    t_rx_data_type frame_type;
    bit [12:0] flags;       // not used
    bit [15:0] frame_id;    // frame counter
} t_rx_frame_header;
```

```
// data following a SOF
typedef union packed {
    struct packed {
        t_rx_single_type_word flag;
        union packed{
            struct packed {
                bit [8:0] row;
                bit [5:0] cm;
            } header;
            struct packed {
                bit [6:0] column;
                bit [7:0] val;
            } data;
        } value;
    } row_single;
    struct packed {  // for raw data
          bit [1:0][7:0] data;
    } raw;
} t_rx_data;
```

Bias Generation DACs & ADC

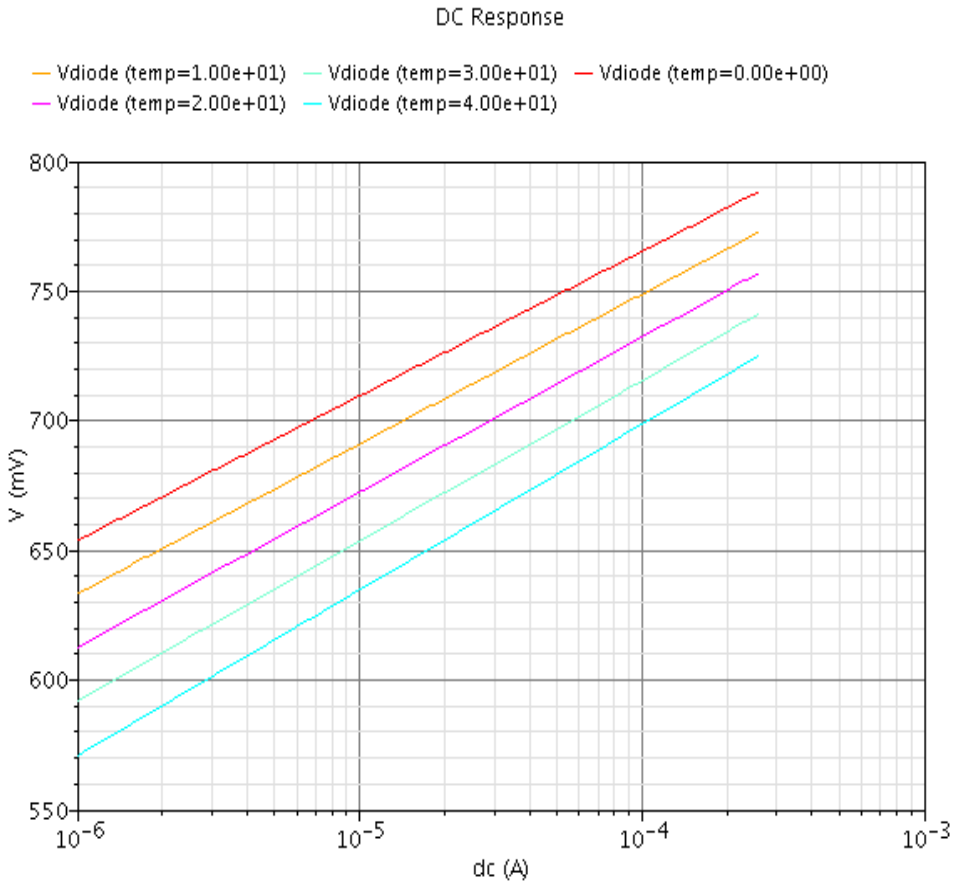


- Bias DACs , 8 bit
  - LVDS RX/TX (3x)
  - PLL (3x)
  - CML driver (3x)
- Temp. indep. current reference
- Test structures
  - OPA
  - Comparator
- ADC, 10bit
  - SAR
  - Rail to rail input range
  - CML driver (pre-emphasis)
- Temperature measurement
  - Internal or external diode

R.Casanova, L.Freixas

Temperature Measurement

- Principle: Measure forward voltage U_D for two diode currents I_1 and I_2
- For $I_1 = N \cdot I_2$ the voltage difference is $dU_D = \eta \cdot kT/q \cdot \ln(N)$ and thus

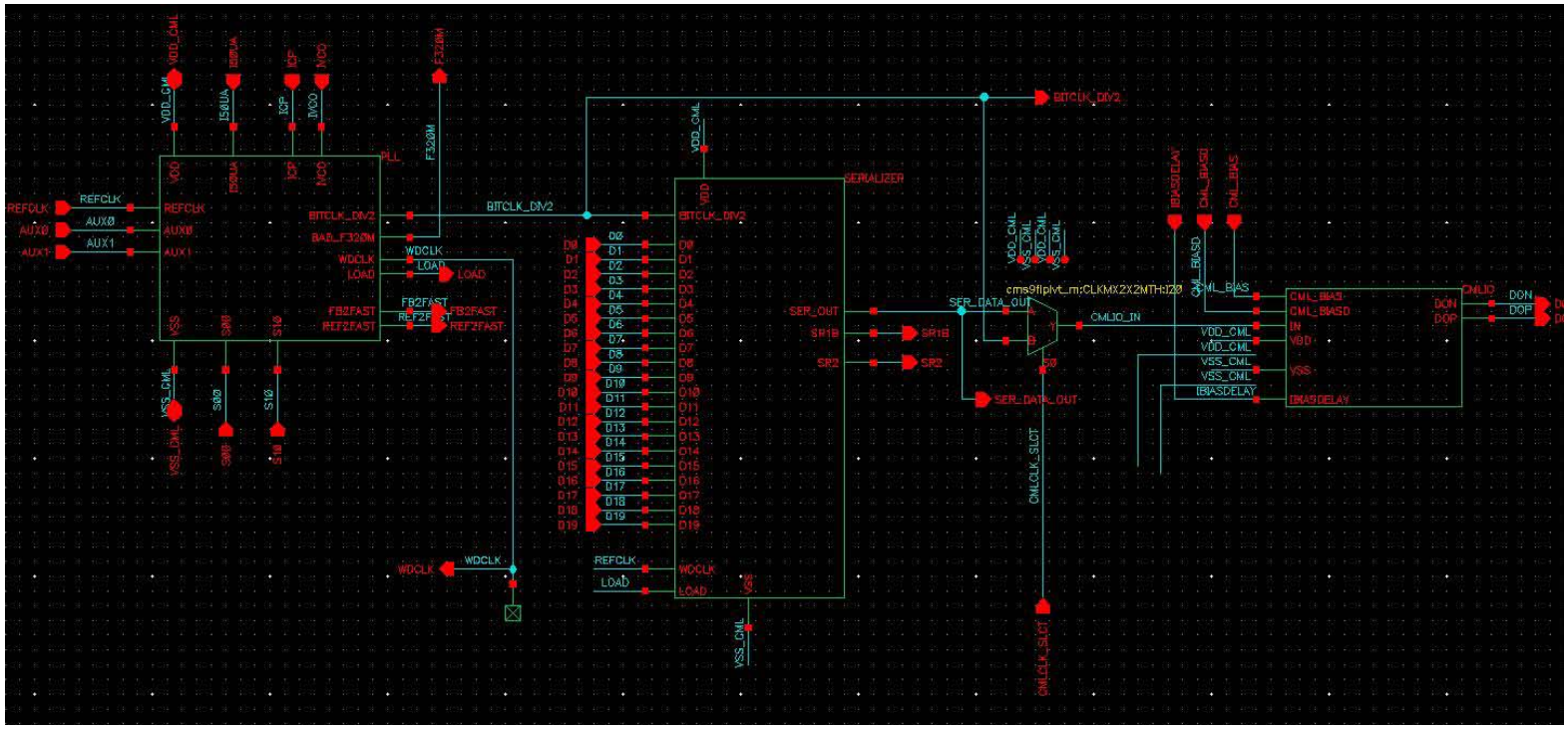


$$T = \frac{q \cdot (U_1 - U_2)}{\eta \cdot k \cdot \ln\left(\frac{I_1}{I_2}\right)}$$

Single measurement accuracy: ~2°C (limited by LSB size of the ADC)

→ Improve by averaging

Data Serializer & Output Driver



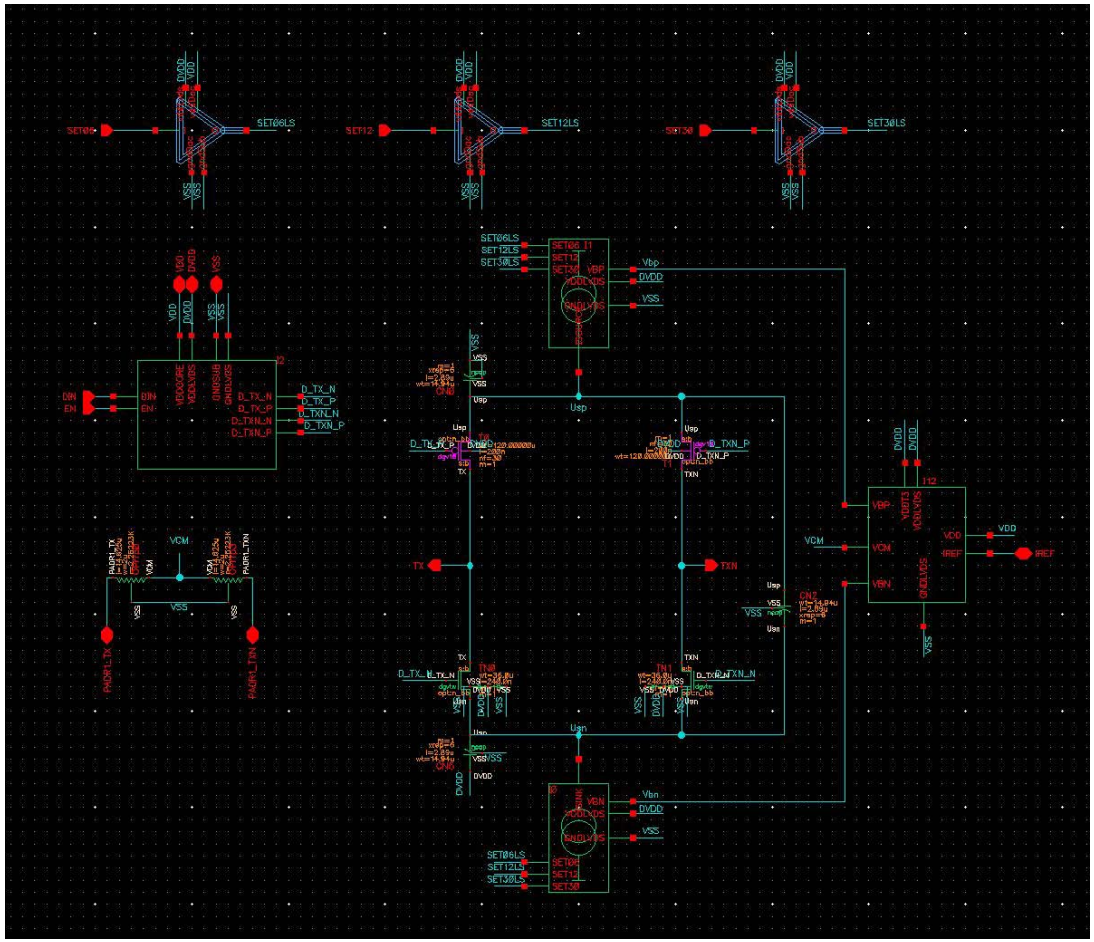
- PLL
  - Input: 80 MHz
  - Out1: 320 MHz (DCD clock)
  - Out2: 800 MHz (Serializer)
- Serializer (20:1)

- CML driver

  Programmable pre-emphasis
  - boost amplitude,
  - boost pulse width

A. Kruth, M.Havranek

LVDS Driver



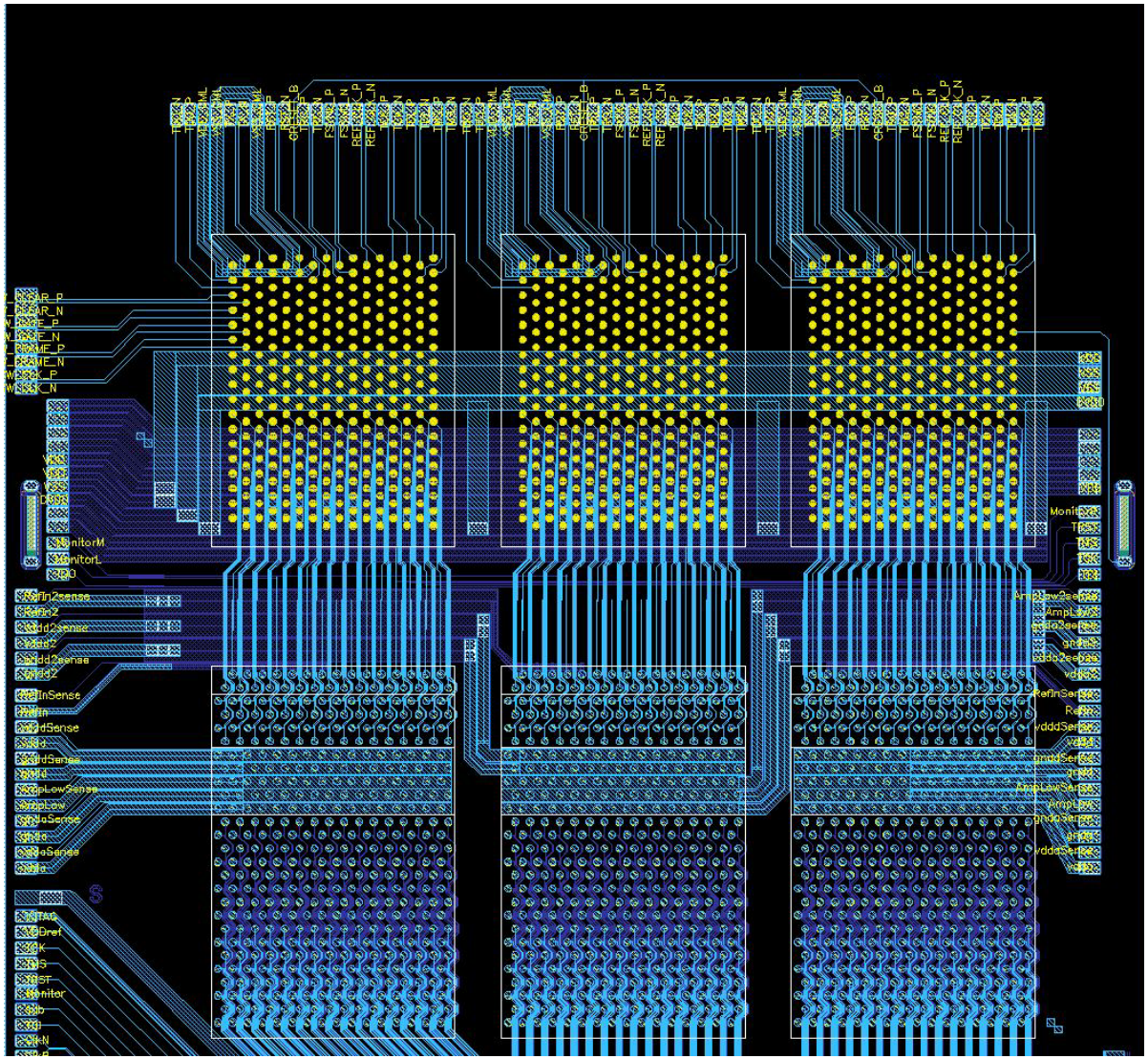
- 1.8V supply
- Programmable output currents:
  0.6, 1.2, 1.8, 2.4 and 3 mA
- Tri-state output
- Active V_{CM} feedback
- V_{OS} = 1.2V

M. Karagounis, I Kishishita

Switcher Sequencer

- 4 output signals (LVDS)
  - sw_clock
  - sw_gate
  - sw_clear
  - sw_new_frame
- Programmable phase on all outputs
  - 3.125 ns step size
  - 32 steps
- **sw_clock**: 50% duty cycle, programmable rising edge
- **sw_gate** and **sw_clear**: programmable rising and falling edge
- **sw_new_frame:** synch with sw_clock (to falling edge) one clock cycle high (programmable distance to external frame_sync signal 0-255)

PXD6 Matrices with DHP 0.2



- 4 small matrices
  - 1 chip DCD/DHP
  - 1 Switcher
  - 128 x 16 pixels
- 4 large matrices
  - 3 chips DCD/DHP
  - 5 Switcher
  - 768 x 120 (180)
- Not all layouts usable
  - M2 only adapted for DHP 0.2 (M1 done last year already)

  → Some layout shorts

Conclusion

- Most of (digital) verification is done
- Data efficiency expected to satisfy requirements (< 1% loss @ 3% occupancy)
- Analogs blocks implemented & verified
- Footprint defined and implemented already on PXD6 matrices (batch 2)
- Everything prepared for submission on Mai 28

BUT:

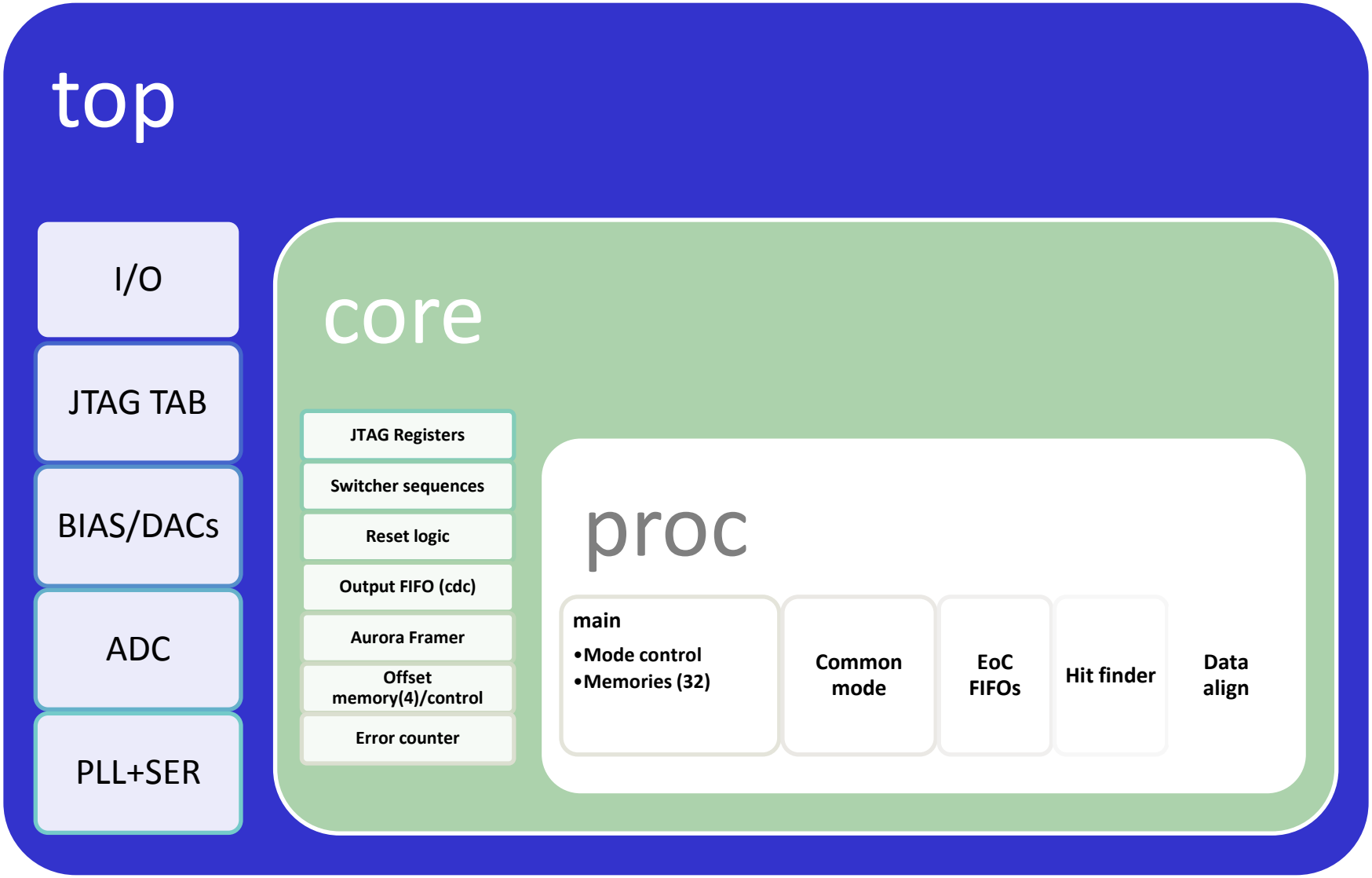
- MOSIS announced this IBM 90nm MPW run to be the last (not yet confirmed that it will even take place).
- We have to change technology for further designs
  - Vendor choice: TSMC, UMC, GF…
  - Stay with 90nm or go to 65nm (synergy with other projects)?
  - Test structures with analog blocks ➔ Delay in DHP development (~6-12 m)

- backup

Address Naming Convention

- Electronic channels:
  - 8 bit Switcher channel address: 0..191 (6 x 32 ch. Switcher)
  - 8 bit DCD channel address: 0..255 (one DCD → one DHP→ one Link)
- Physical (pixel) addresses:
  - 10 bit pixel row address: 0..767
  - 6 bit pixel column address: 0..63
  - this format is used for the hit data words inside the DHP

DHP 0.2 Block Diagram



Some more statistics

- General Design Information

Design Status	Routed
Design Name	dhp_top
# Instances	210162
# Hard Macros	120
# Std Cells	209631
# Pads	411
# Net	169040
# Special Net	6
# IO Pins	155
# Pins	631902
# PG Pins	419584
Average Pins Per Net(Signal)	3.738
Total Wire Length	13431638.51 um

- Floorplan/Placement Information

Total area of Standard cells	1327012.669 um^2
Total area of Standard cells(Subtracting Physical Cells)	1207224.995 um^2
Total area of Macros	5455043.037 um^2
Total area of Blockages	0.000 um^2
Total area of Pad cells	2110790.000 um^2
Total area of Core	9411868.186 um^2
Total area of Chip	12476000.000 um^2
Effective Utilization	3.2459e-01
Number of Cell Rows	1806
% Pure Gate Density #1 (Subtracting BLOCKAGES)	14.099%
% Pure Gate Density #3 (Subtracting MACROS)	33.537%
% Core Density (Counting Std Cells and MACROs)	72.059%
% Core Density #2(Subtracting Physical Cells)	70.786%
% Chip Density (Counting Std Cells and MACROs and IOs)	71.280%
% Chip Density #2(Subtracting Physical Cells)	70.319%