

DHP emulator

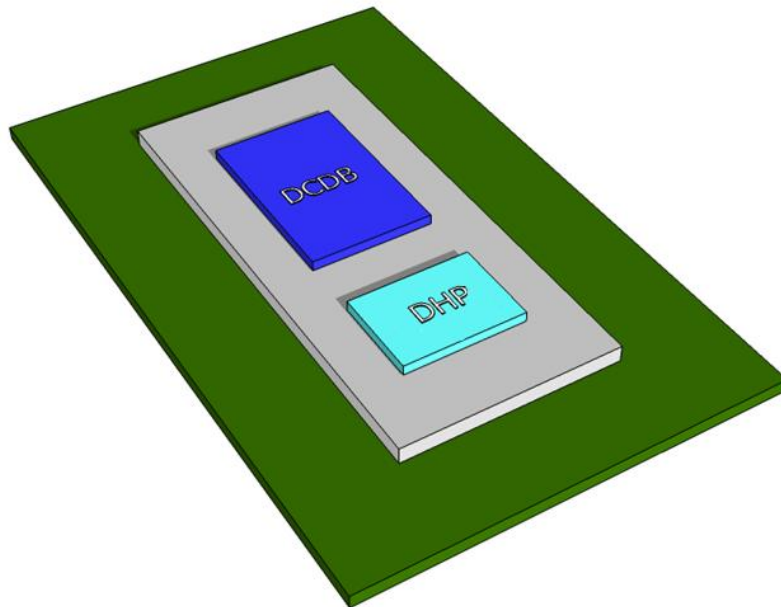
5th International Workshop on DEPFET Detectors and
Applications

from 29 September 2010 to 01 October 2010

Mikhail Lemarenko

DHP 0.1 chip

- First prototype of the Data Handling Processor chip



DHP emulator

- Processing unit implemented together with DCD-B readout firmware
- Exists as HDL code realized in FPGA (Virtex-4)

It is NOT

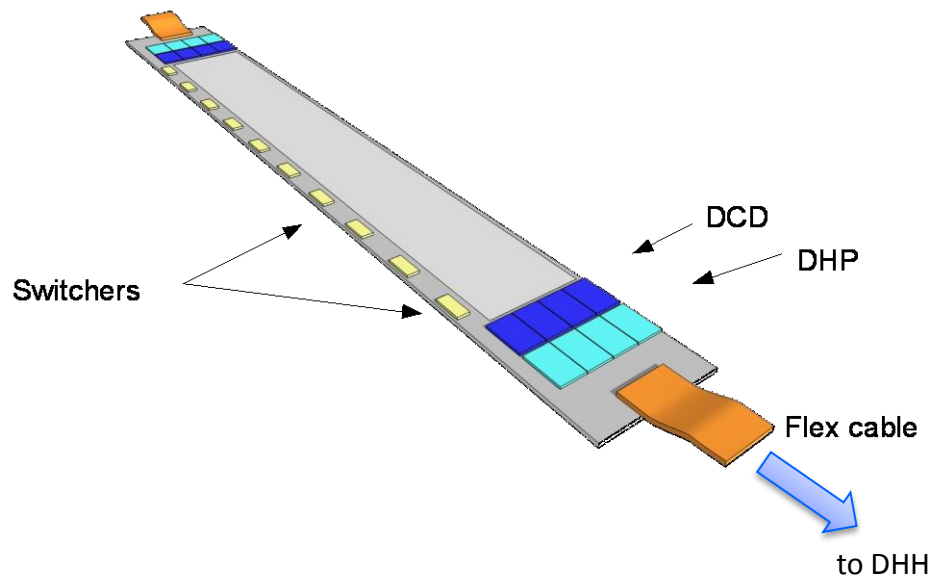
- FPGA replication of the DHP 0.1
- Alternative to the software simulation

...but it is about :

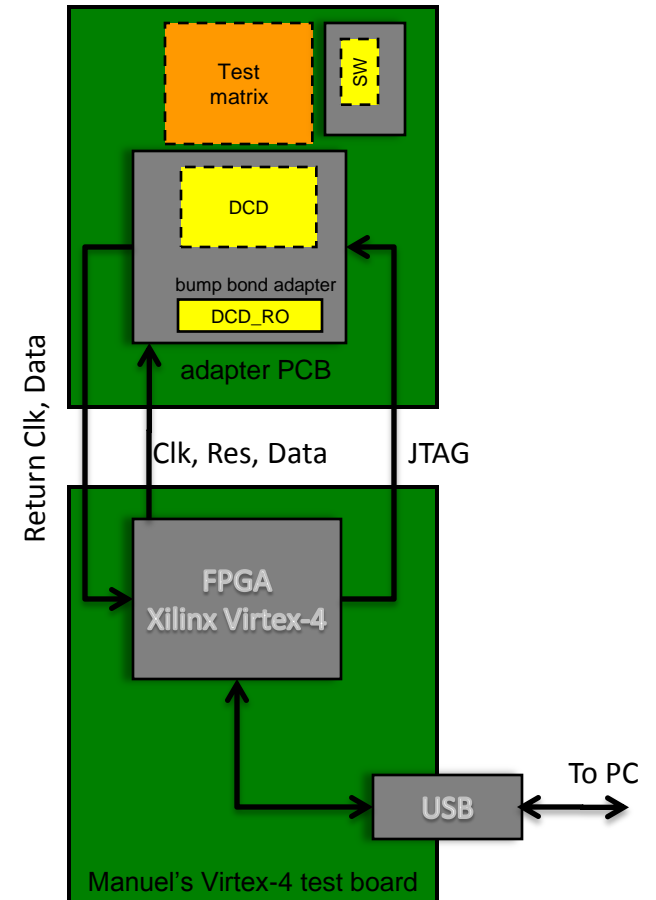
- Having a fast DCD-B readout system
- Checking the optimal DHP data processing configuration (what we think it is) on the real data
- Having a similar readout conditions as in the final experiment
- What is feasible in FPGA is also likely to be easily done in the real chip

- DEPFET module

(Still a long way to go)

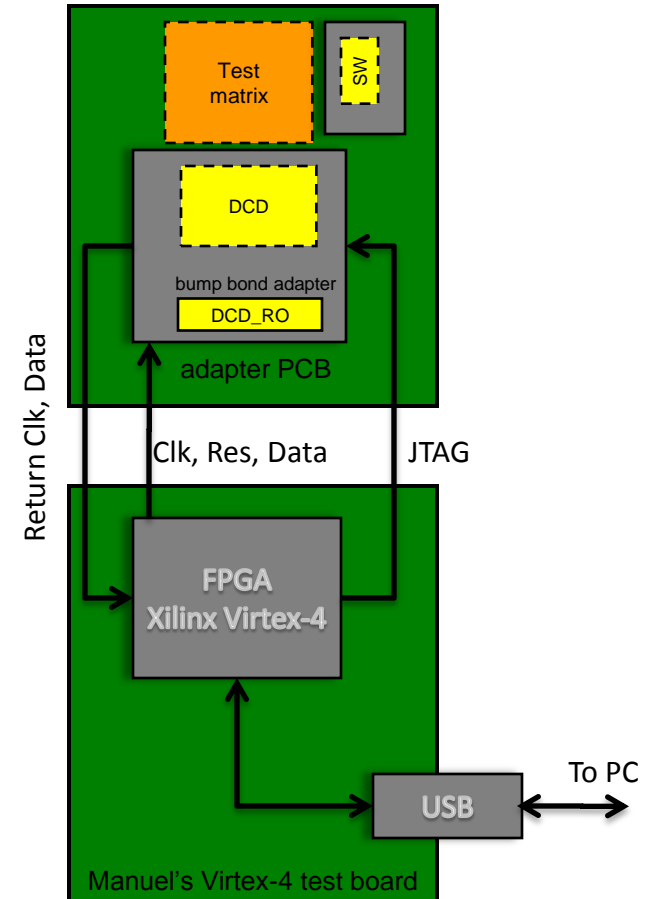


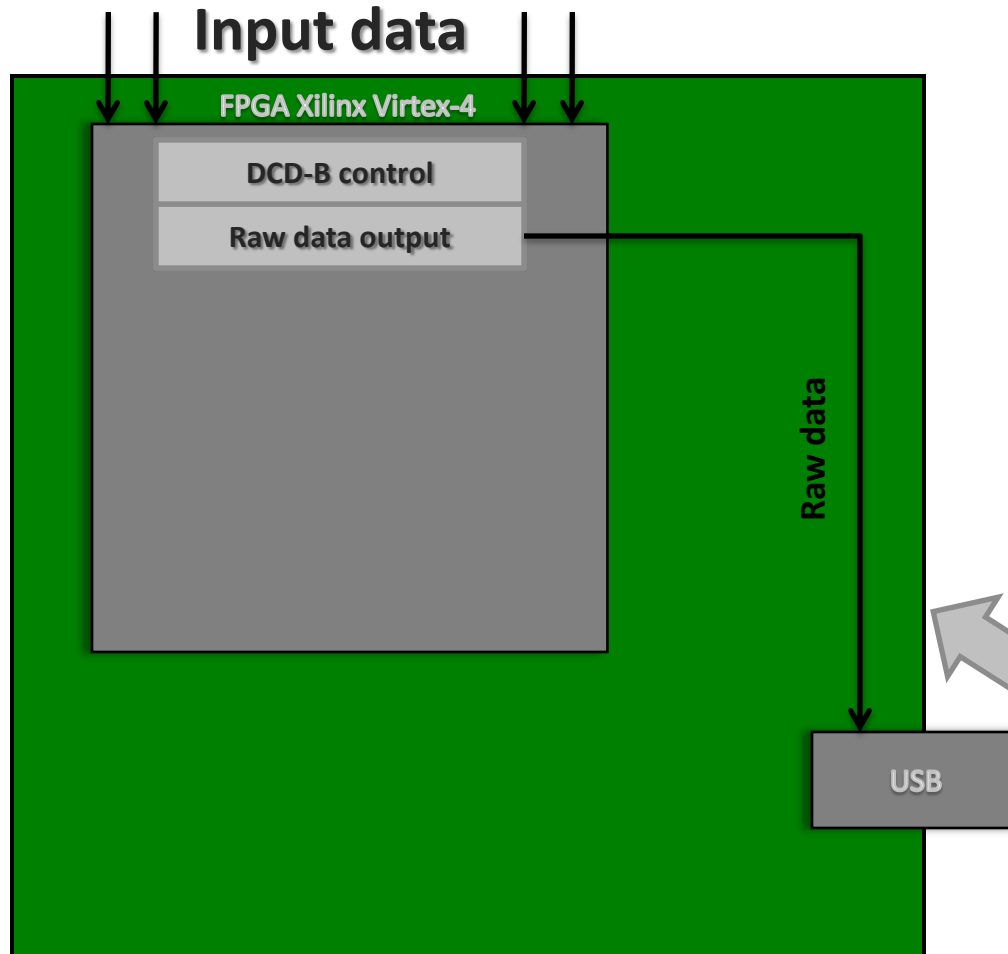
- DCD-B Test system



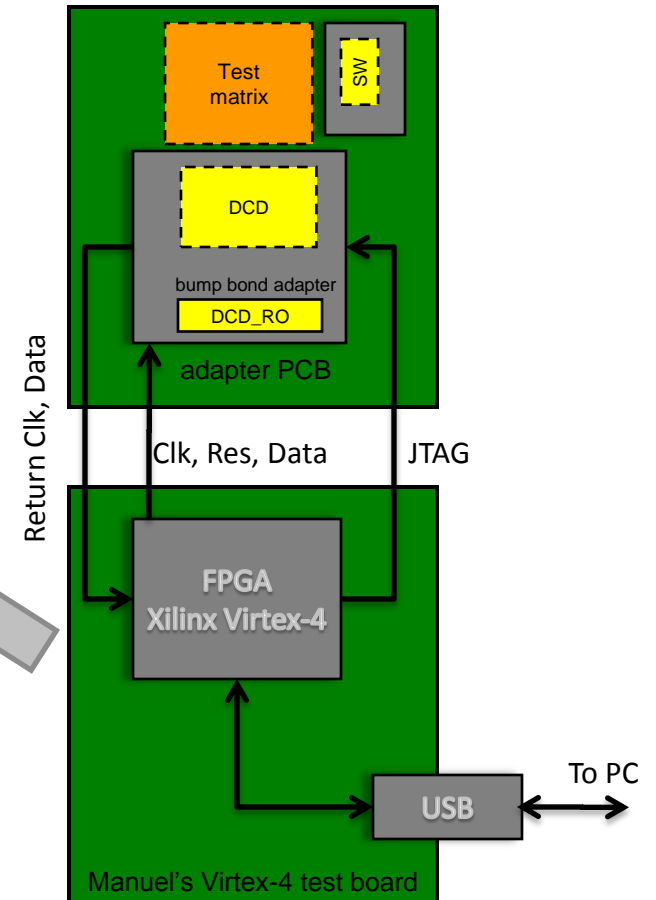


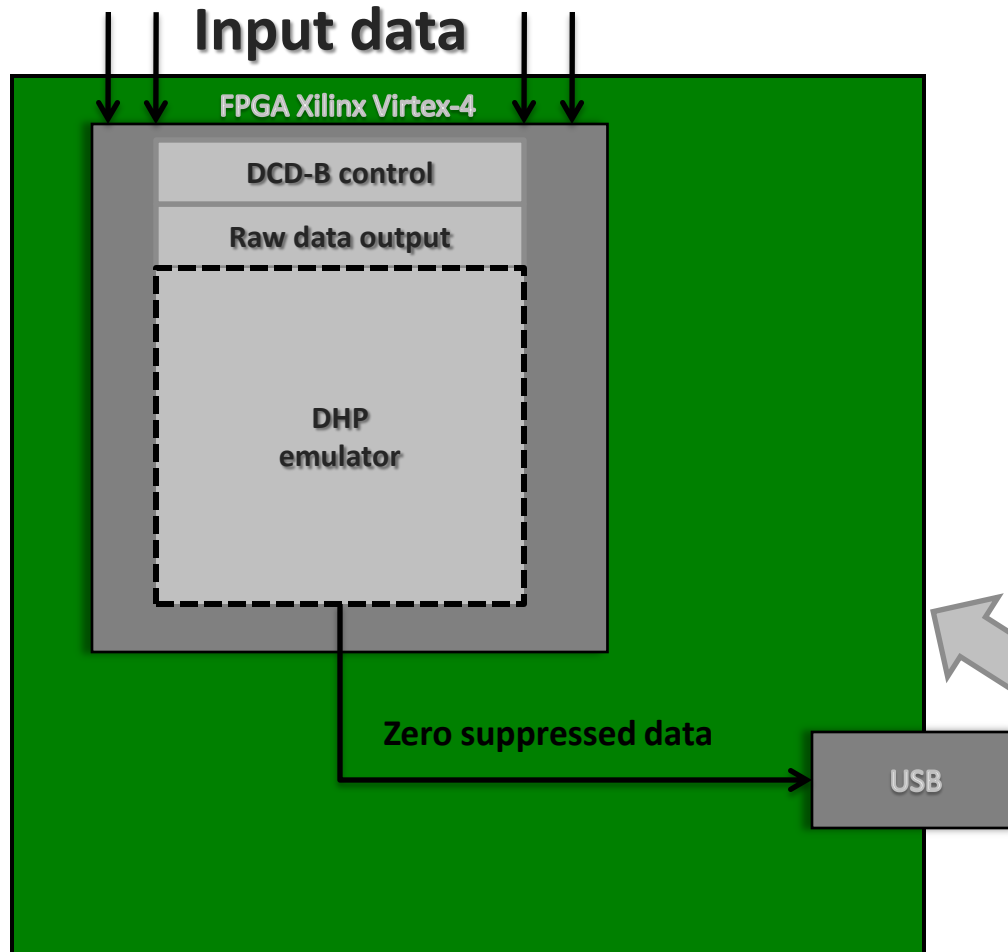
- DCD-B Test system



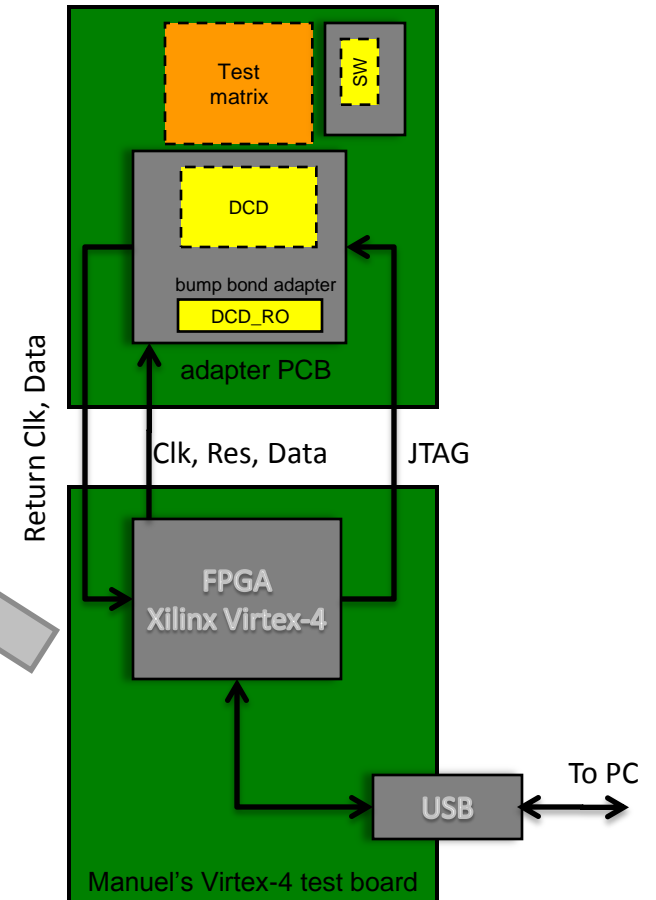


- DCD-B Test system



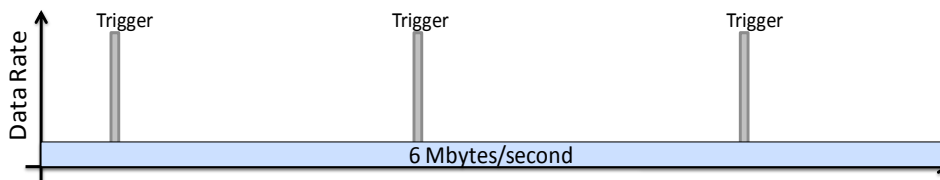


- DCD-B Test system



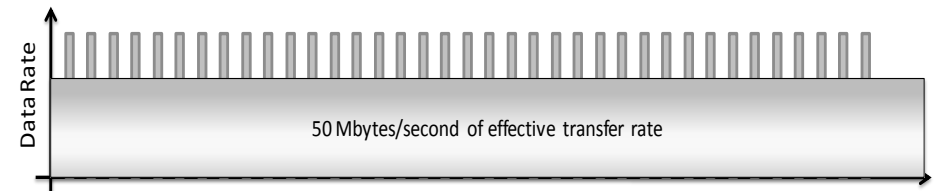
Raw Data DCD-RO output

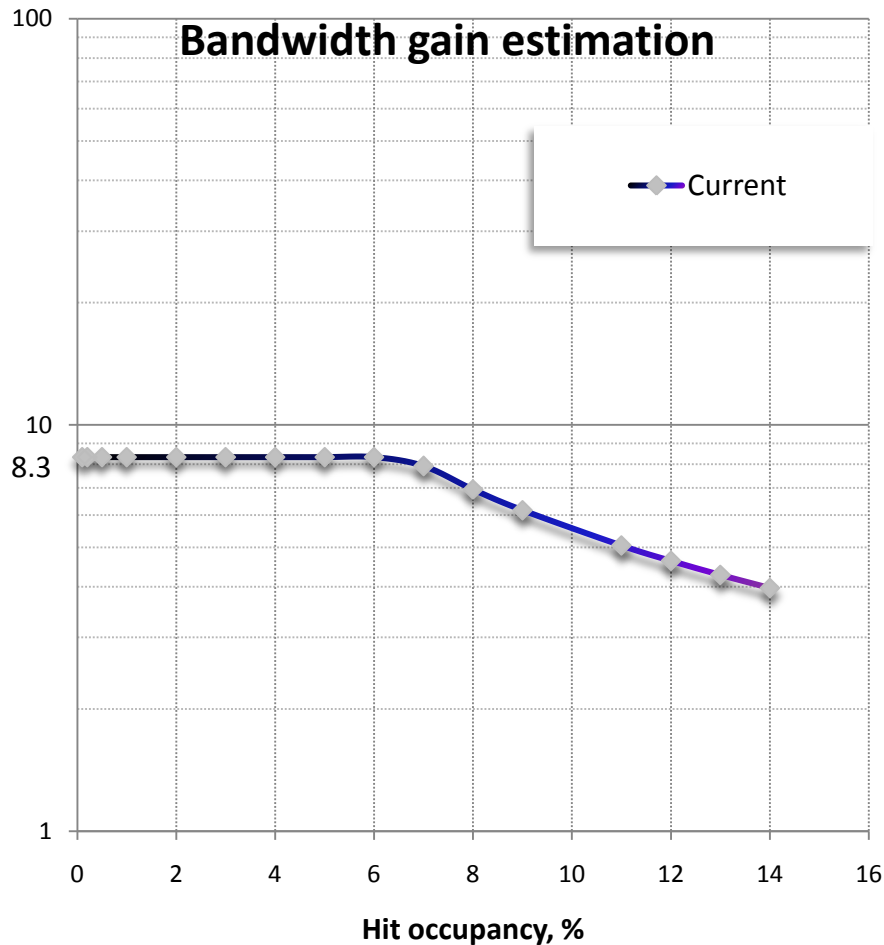
- Frame-per-frame-wise operation equal to 4104bytes of data for a 64x64 currently connected matrix
- Best possible estimated Frame rate ~ 1.5kHz



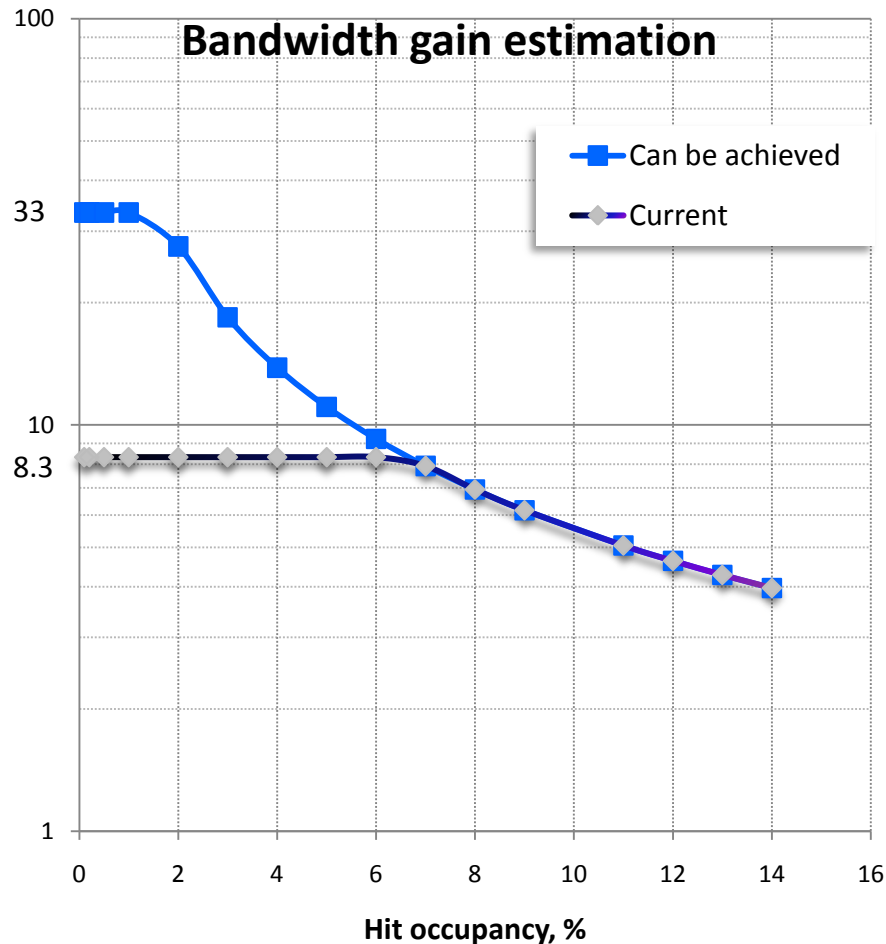
DHP emulator (Zero suppression mode)

- Only zero suppressed data are sent out using format:
 $Out = [Trig_i][V_1 X_1 Y_1][V_2 X_2 Y_2] \dots [Trig_{(i+1)}] \dots$
- Tested current frame rate ~12.5kHz (@ ~1% occupancy)



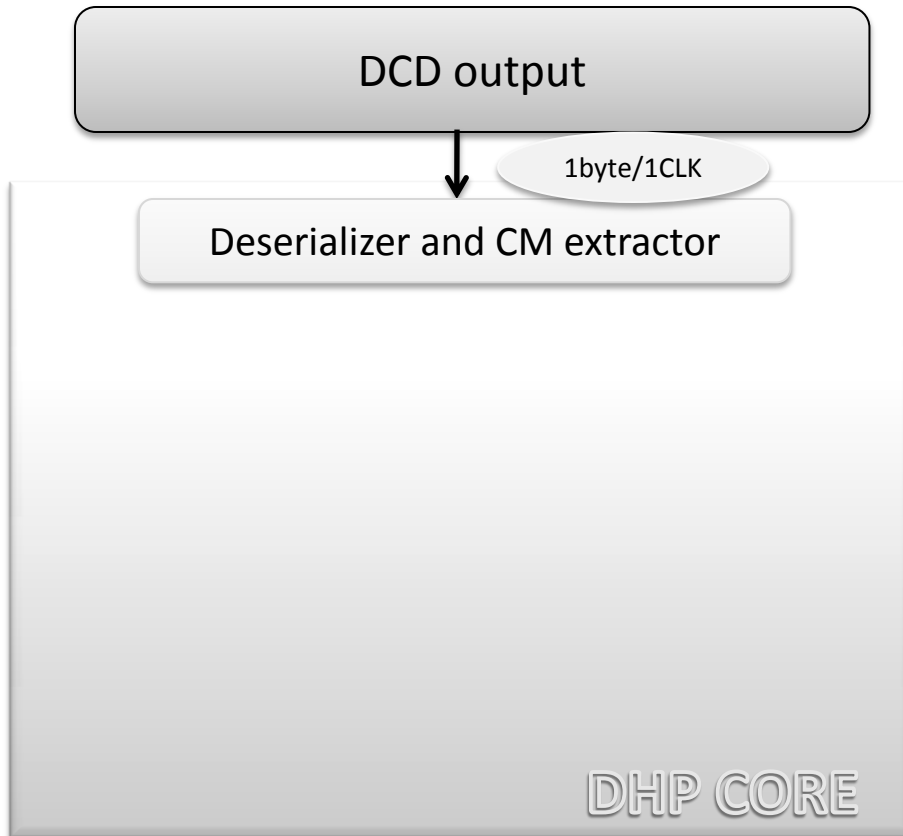


- How faster do we transmit data using zero suppression



- How faster do we transmit data using zero suppression
- Still can be faster by a factor of 4 or even better (bottleneck optimization)

Structural schematic

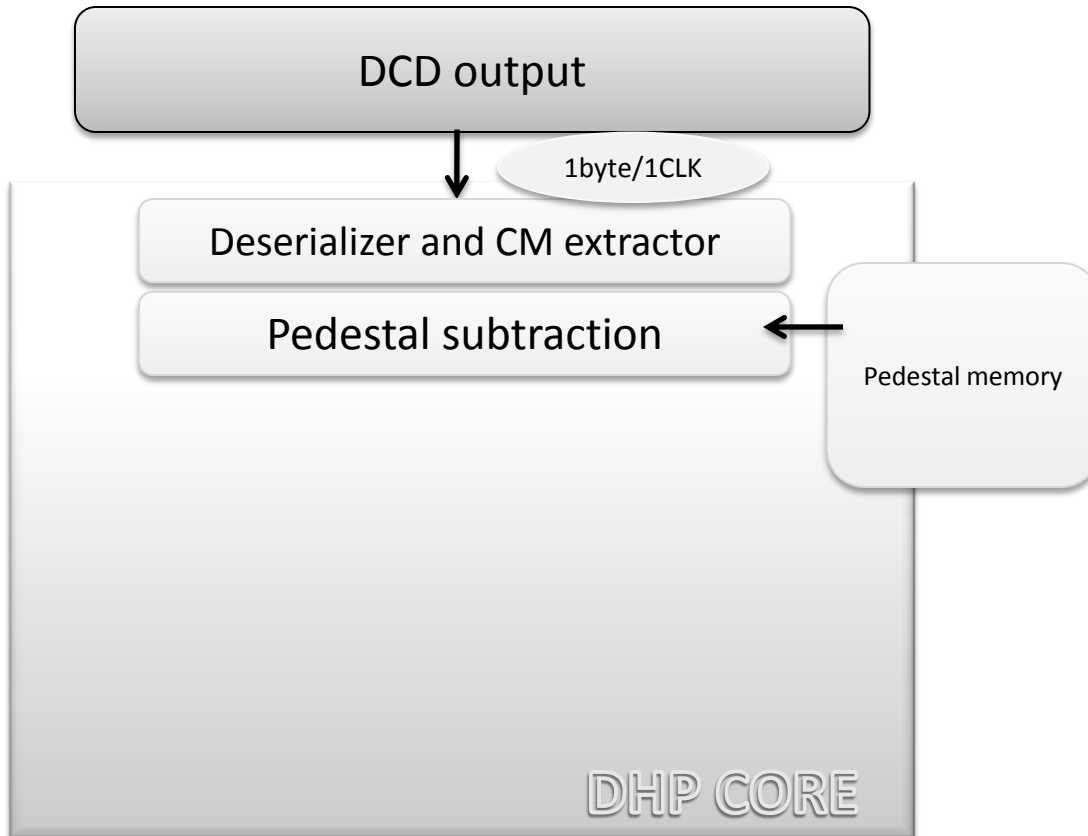


- CM subtraction and all other intermediate operations are done using 16bits precision to avoid introducing additional digitization error.
- Pedestals are stored in 16bits precision values, enough to do the dynamic pedestals update of averaging up to:

$$P_{N+1} = \frac{255P_N + I_N}{256}$$

(see DHP simulation talk)

Structural schematic

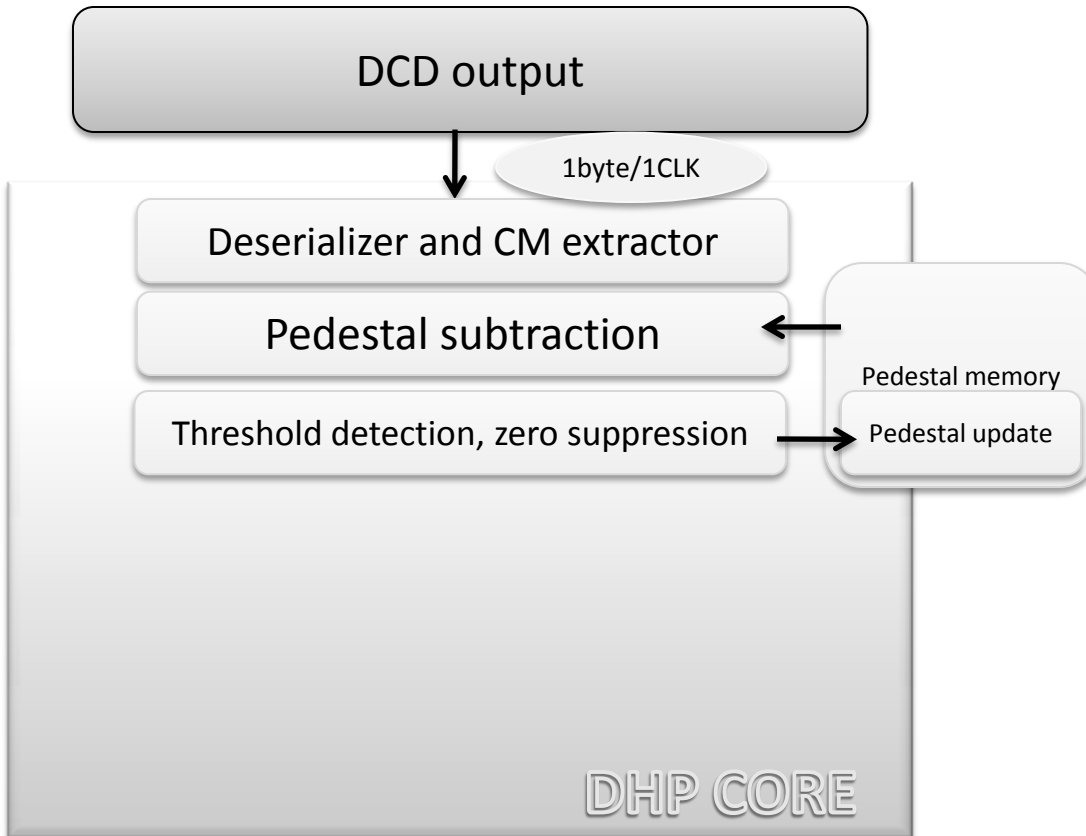


- CM subtraction and all other intermediate operations are done using 16bits precision to avoid introducing additional digitization error.
- Pedestals are stored in 16bits precision values, enough to do the dynamic pedestals update of averaging up to:

$$P_{N+1} = \frac{255P_N + I_N}{256}$$

(see DHP simulation talk)

Structural schematic

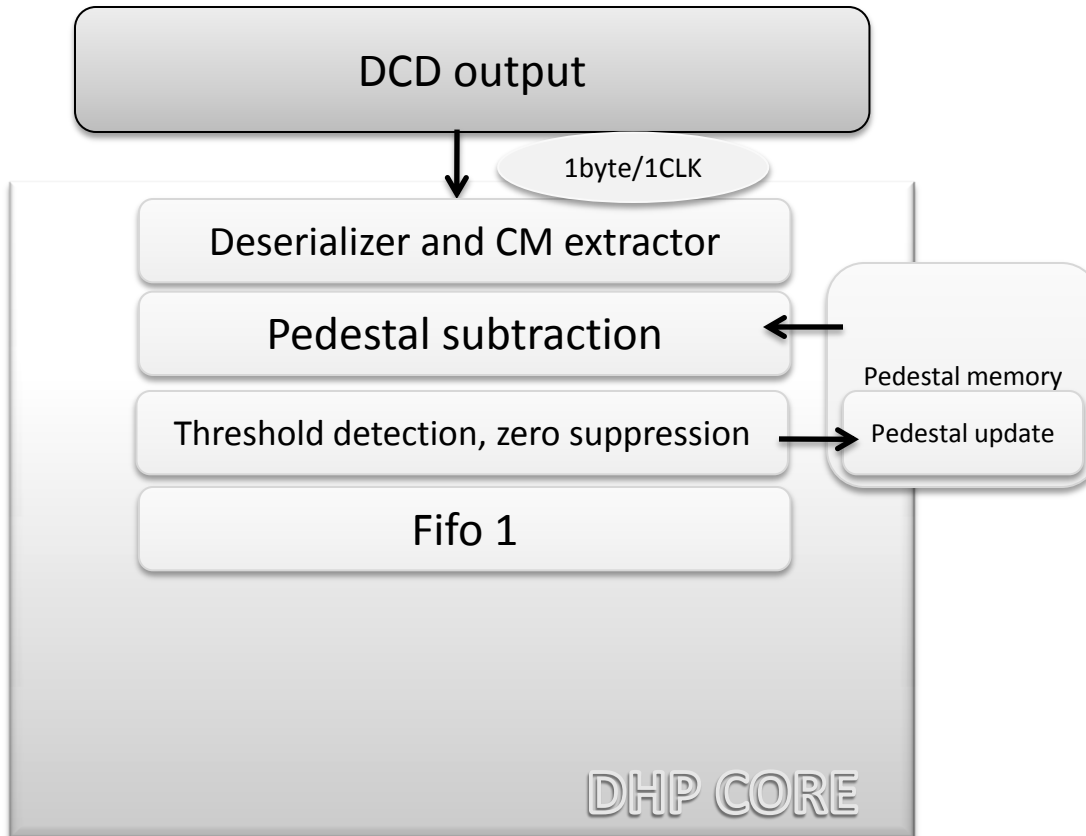


- CM subtraction and all other intermediate operations are done using 16bits precision to avoid introducing additional digitization error.
- Pedestals are stored in 16bits precision values, enough to do the dynamic pedestals update of averaging up to:

$$P_{N+1} = \frac{255P_N + I_N}{256}$$

(see DHP simulation talk)

Structural schematic

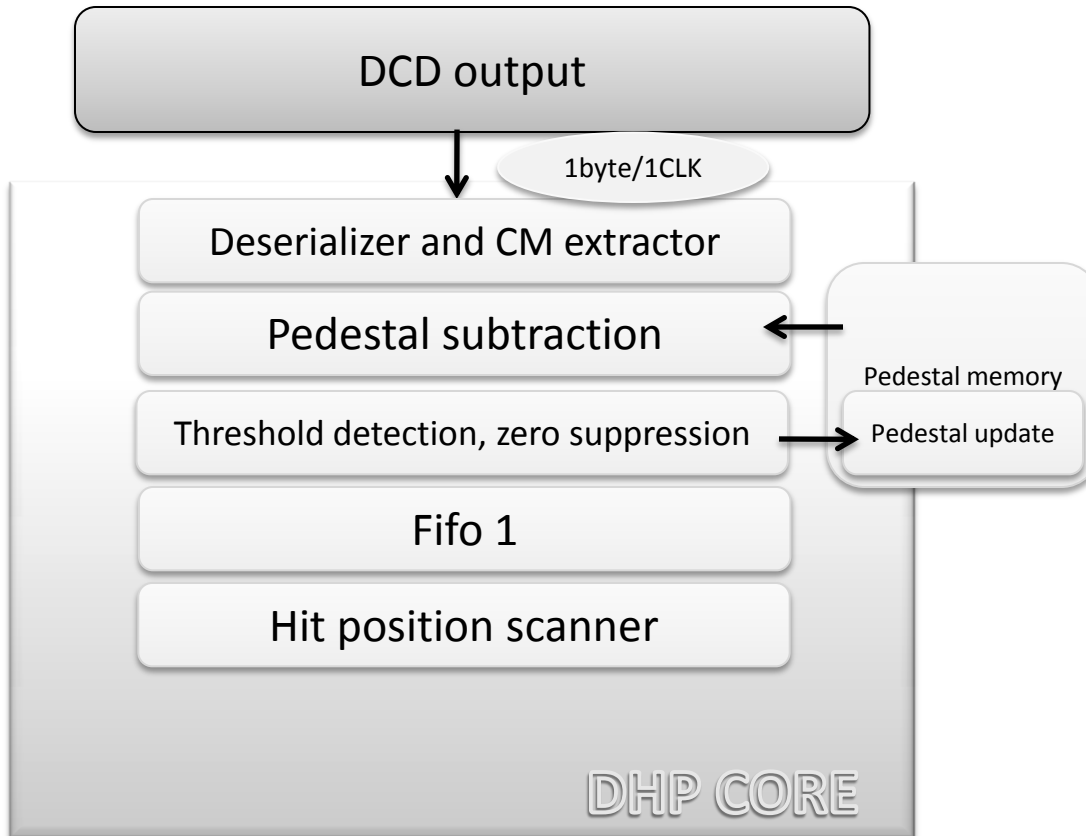


- CM subtraction and all other intermediate operations are done using 16bits precision to avoid introducing additional digitization error.
- Pedestals are stored in 16bits precision values, enough to do the dynamic pedestals update of averaging up to:

$$P_{N+1} = \frac{255P_N + I_N}{256}$$

(see DHP simulation talk)

Structural schematic

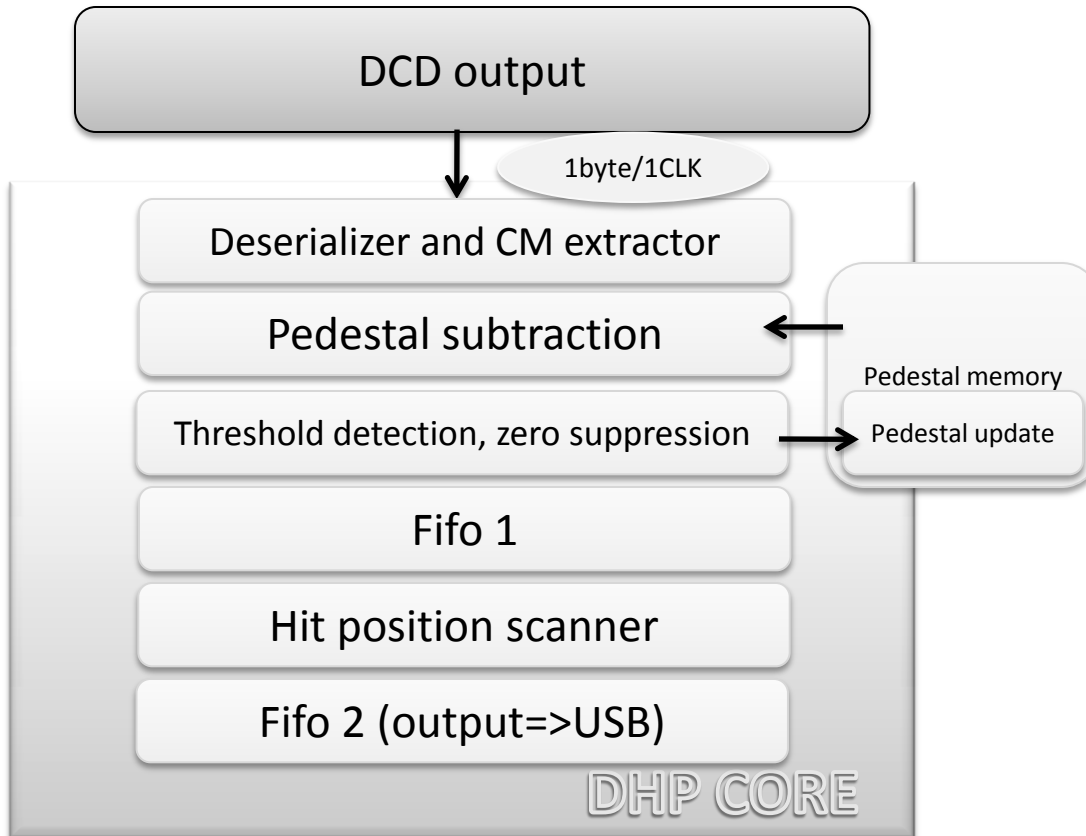


- CM subtraction and all other intermediate operations are done using 16bits precision to avoid introducing additional digitization error.
- Pedestals are stored in 16bits precision values, enough to do the dynamic pedestals update of averaging up to:

$$P_{N+1} = \frac{255P_N + I_N}{256}$$

(see DHP simulation talk)

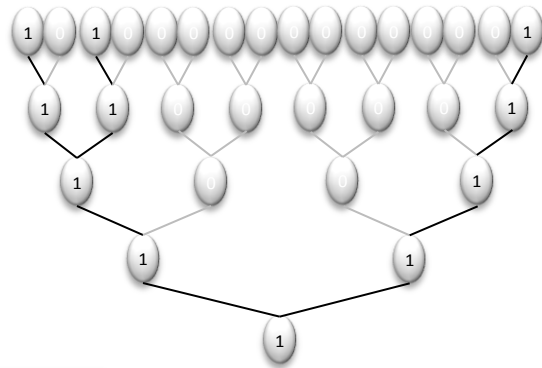
Structural schematic



- CM subtraction and all other intermediate operations are done using 16bits precision to avoid introducing additional digitization error.
- Pedestals are stored in 16bits precision values, enough to do the dynamic pedestals update of averaging up to:

$$P_{N+1} = \frac{255P_N + I_N}{256}$$

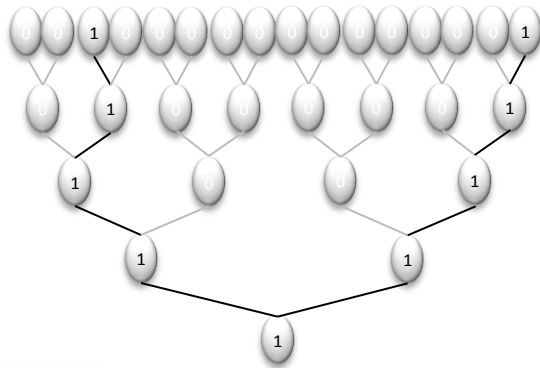
(see DHP simulation talk)



Left is 0
Right is 1

Hit found with path from bottom to top:
left-left-left-left \Leftrightarrow [0000] = position 0

- Although seemingly trivial task may be a serious bottleneck if not seriously taken in consideration
- Hit scanner is implemented using 'binary tree' technique, inspired from DHP predecessors (MEPHISTO, CURO etc...)
- Tree is updated asynchronously, giving one hit position per clock

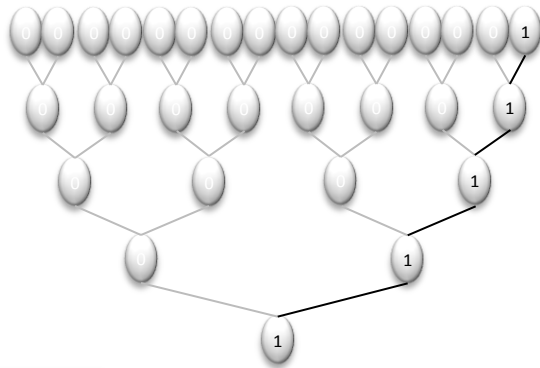


Left is 0
Right is 1

Hit found with path from bottom to top:
left-left-left-left \Leftrightarrow [0000] = position 0

Hit found with path from bottom to top:
left-left-right-left \Leftrightarrow [0010] = position 2

- Although seemingly trivial task may be a serious bottleneck if not seriously taken in consideration
- Hit scanner is implemented using 'binary tree' technique, inspired from DHP predecessors (MEPHISTO, CURO etc...)
- Tree is updated asynchronously, giving one hit position per clock



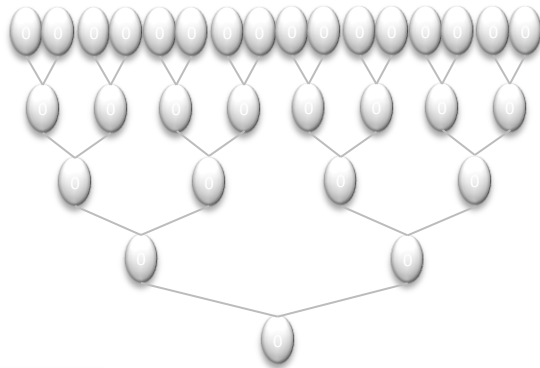
Left is 0
Right is 1

Hit found with path from bottom to top:
left-left-left-left \Leftrightarrow [0000] = position 0

Hit found with path from bottom to top:
left-left-right-left \Leftrightarrow [0010] = position 2

Hit found with path from bottom to top:
right-right-right-right \Leftrightarrow [1111] = position 15

- Although seemingly trivial task may be a serious bottleneck if not seriously taken in consideration
- Hit scanner is implemented using 'binary tree' technique, inspired from DHP predecessors (MEPHISTO, CURO etc...)
- Tree is updated asynchronously, giving one hit position per clock



Left is 0
Right is 1

Hit found with path from bottom to top:
left-left-left-left \Leftrightarrow [0000] = position 0

Hit found with path from bottom to top:
left-left-right-left \Leftrightarrow [0010] = position 2

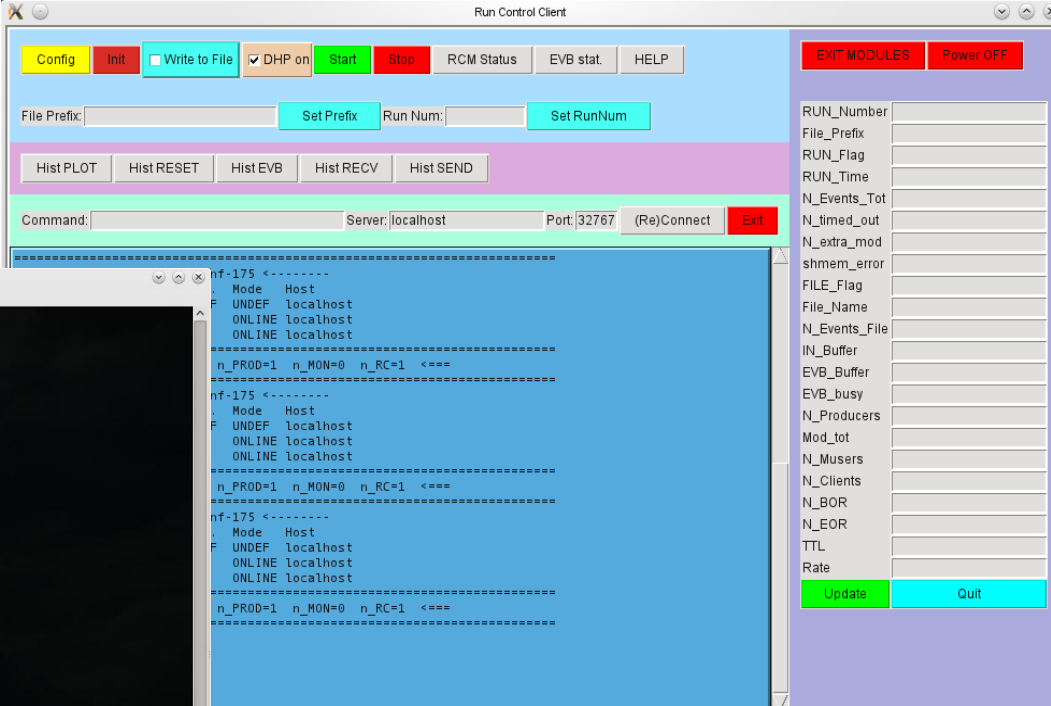
Hit found with path from bottom to top:
right-right-right-right \Leftrightarrow [1111] = position 15

All hits are found in 3 clock cycles instead of 16

- Although seemingly trivial task may be a serious bottleneck if not seriously taken in consideration
- Hit scanner is implemented using 'binary tree' technique, inspired from DHP predecessors (MEPHISTO, CURO etc...)
- Tree is updated asynchronously, giving one hit position per clock

In process...

```
DAQ/bib: depfet
File Edit View Scrollback Bookmarks Settings Help
[019| 8|61] [019|24|61] [020|15|62] [020|36|63] [020|30| 4] [020|24| 5] [020|61| 6] [019|37| 7]
[019|46| 7] [019|62|12] [020|22|13] [019|11|14] [019|59|14] [020| 8|16] [020|31|18] [019|10|22]
[019|51|23] [019|37|26] [019|41|28] [020|12|29] [020|15|30] [020|62|31] [020|21|33] [020|18|38]
[020|37|40] [020|18|41] [020| 4|42] [020|45|42] [020|19|43] [020|13|45] [019|11|46] [019|59|46]
[020|27|48]
Frame 30
Trigger: 65280
[019|54|33] [018| 6|34] [019|61|34] [020|25|35] [020|18|40] [020| 6|42] [019|23|43] [019|43|43]
[020|19|44] [020|26|45] [020| 6|46] [020|38|48] [018| 4|50] [019|20|50] [019|51|50] [020|54|52]
[020|48|54] [020|13|55] [020|33|60] [020|19|63] [019|23| 1] [020|44| 4] [020|27| 7] [020|64|10]
[019|19|13] [020|50|14] [020|57|18] [020| 9|21] [020|40|24]
Frame 31
Trigger: 65283
[018|64|60] [020|55|61] [019|47|64] [020|22| 1] [019|12| 3] [019|22| 4] [020|17| 6] [019|20| 8]
[019|32|11] [020|24|12] [020|14|13] [019|23|14] [019|30|14] [019|48|15] [019|64|15] [020|13|26]
[020|17|28] [020|64|30] [019|38|31] [020|39|33] [020|62|35] [019| 7|37] [019|58|37] [020|13|40]
[019|44|41] [020| 3|46] [019| 3|48] [020| 3|55]
Frame 32
Trigger: 65281
[020| 7|23] [020| 8|24] [020|12|28] [019|22|29] [019|50|29] [020|29|32] [019|60|34] [020|54|37]
[020|14|38] [020|30|39] [019| 5|40] [019|48|40] [019|64|40] [020|29|42] [020|13|51] [019|13|52]
[019|64|52] [020|38|56] [019|42|61] [019| 7|62] [019| 3|63] [019|40|63] [020|44| 2] [019| 3| 3]
[019|37| 3] [019|55| 3] [020|58| 4] [020|36| 6] [020| 3| 7] [020|28| 8] [020| 4|11] [020| 9|13]
[019|33|14] [019|23|16] [019| 8|20] [019|54|20]
Frame 33
Trigger: 65280
[020| 1|23] [020|38|26] [020| 5|27] [020|43|28] [019|46|30] [020|22|31] [019| 4|32] [019|23|33]
[019|44|33] [019| 7|37] [019|34|37] [019| 5|38] [019|23|38] [018|62|38] [019|41|39] [019|57|39]
[020|22|41] [020|40|43] [020|57|44] [020|36|47] [020| 6|50] [019| 6|51] [019|57|51] [020|31|55]
[020|51|56] [020|35|60] [019|64|60] [020|33|62] [020|61|63] [020|60| 1] [020|60| 5] [020|21| 7]
[020|61| 9] [019|26|13] [019|63|13] [020|47|14] [020|16|15] [019|31|18] [020| 1|19] [020|31|20]
Frame 34
Trigger: 65281
[020|41|56] [019|11|57] [019|37|57] [019|42|59] [020|17|41] [020|61|49] [019|40|50] [019|56|50]
[019|21|52] [020|39|54] [020|56|55] [020|49|56] [020|42|58] [019| 5|61] [019|21|61] [019|46|61]
[020|56|62] [020|25|63] [019|30| 2] [019|46| 2] [019|10| 6] [020|19| 6] [019|34| 7] [019|63| 7]
[019| 2| 9] [018|63| 9] [019|60|10] [020|21|11] [019|10|12] [019|52|12] [019|59|12] [019|28|16]
[019|11|19] [019|55|19] [019|53|23] [019|62|23] [019|45|24] [020|32|28] [020|30|31]
Tail data size 1578
and 1t contains: 255|03|20|29|46|20|47|49|20|03|53|20|02|54|19|20|56|19|36|56|19|19|60|20|29|62|19|22|64|20|49|02|20|01|03|20|
8|19|43|13|20|01|14|19|01|15|19|43|15|20|40|16|20|54|17|20|32|18|19|43|23|19]
--> TDepfetProducerLab::Read DHP: E v e n t b u i l d i n g <-----
Data dumping happened correctly
currently we have : 0 frames
```



The Run Control Client interface includes a top toolbar with buttons for Config, Init, Write to File, DHP on (checked), Start, Stop, RCM Status, EVB stat, and HELP. Below this are input fields for File Prefix and Run Num, with Set Prefix and Set Run Num buttons. A row of histogram control buttons (Hist PLOT, Hist RESET, Hist EVB, Hist RECV, Hist SEND) is present. The Command field is empty, and the Server is set to localhost with Port 32767. A (Re)Connect button and an Exit button are also visible. On the right side, there are two red buttons: EXIT MODULES and Power OFF. A list of parameters is shown on the right, including RUN_Number, File_Prefix, RUN_Flag, RUN_Time, N_Events_Tot, N_timed_out, N_extra_mod, shmем_error, FILE_Flag, File_Name, N_Events_File, IN_Buffer, EVB_Buffer, EVB_busy, N_Producers, Mod_tot, N_Musers, N_Clients, N_BOR, N_EOR, TTL, and Rate. At the bottom of this list are Update and Quit buttons.

...will be ready soon

- Although data processing is initially checked with the software simulation, everything have to be verified for its feasibility in hardware
- All features can be tested in real conditions
- DHP emulator can be used during the next test beam together with the regular DCD readout system
 - Read data with the zero suppressed mode
 - Faster or continuous readout allow us to check several DEPFET matrix properties impossible to be seen with the slow readout, such as complete clear check

(To be done)

- Clustering
 - Depends a lot on average cluster size
- Increase input bandwidth

THANK YOU