FPGA IMPLEMENTATION OF SHAPE FITTING ALGORITHM WITH ELECTRON-PHOTON AND HADRON COMPONENTS FOR RECONSTRUCTION IN THE BELLE II ELECTROMAGNETIC CALORIMETER

O. Radchenko, A. Kuzmin, M. Remnev, V. Zhulanov

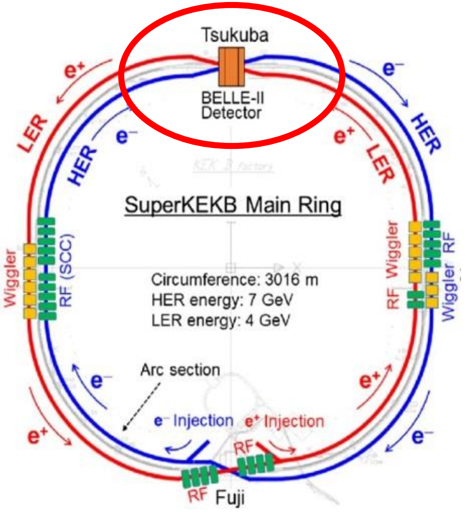
Budker Institute of Nuclear Physics

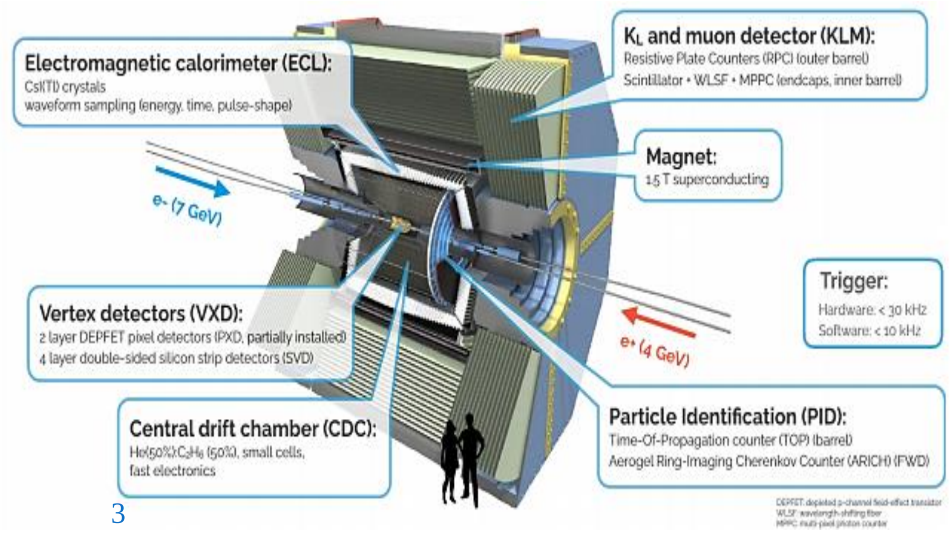
Presentation structure

1. SuperKEKB and Belle II detector
2. Electromagnetic calorimeter electronics
3. Motivation
4. Current fitting algorithm and fitting algorithm with two component
5. FPGA implementation and design optimization
6. Test environment
7. Conclusion

SuperKEKB and Belle II detector

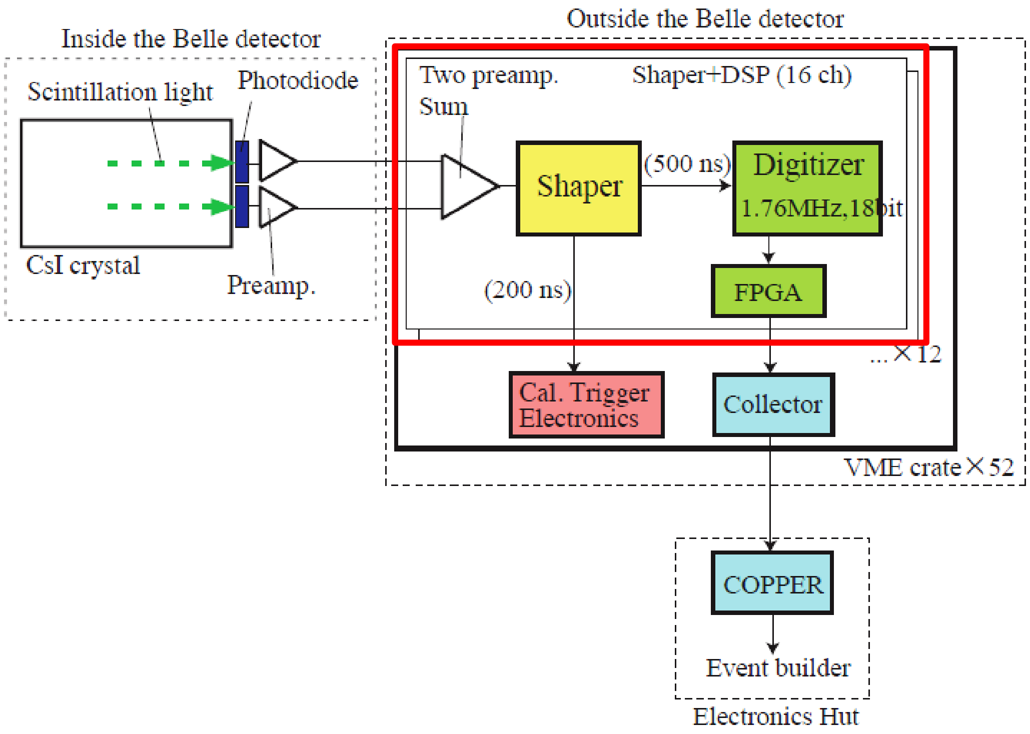
- SuperKEKB – electron-positron collider ($E_{e^-} = 7$ GeV, $E_{e^+} = 4\ GeV$).
- Project luminosity $6 * 10^{35} cm^{-2} s^{-1}$.
- Current peak luminosity $5.2 * 10^{34} cm^{-2} s^{-1}$.
- Physics program – study of rare B-, D-mesons, τ lepton, CP violation, search for new physics.





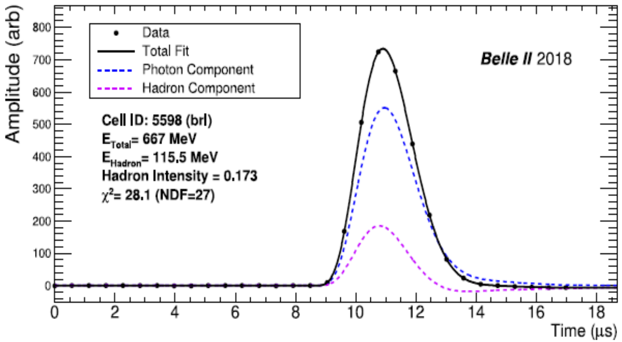
Electromagnetic calorimeter electronics

- 8736 CsI(Tl) counters (30 cm, 16.1X0);
- Trigger rate – 30 kHz;
- ShaperDSP board – shaper and digitizer, digital signal processor (waveform fitting);
- 18 bit ADC;
- ADC sampling frequency – 1.76 MHz;



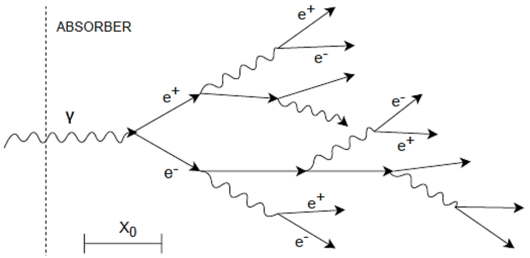
Motivation

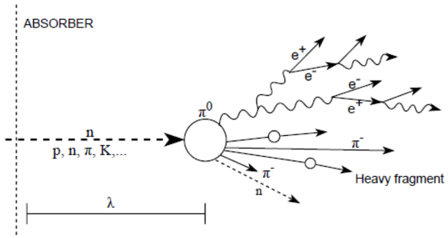
- In the CsI(Tl) crystals decay time depends on the ionization density.
- For hadron interacting in the calorimeter there are particles with low momenta having high ionization density.
- The pulse shape discrimination method using CsI(Tl) crystals is a novel approach recently used at the Belle II electromagnetic calorimeter. Fitting the signal by sum of two components allows to separate fraction of the signal with short decay time (hadron component). *[1] NIM, A 982 (2020) 164562, arXiv:2007.09642*
- This pulse shape discrimination procedure is used in offline data processing. But we need to save raw data (energy more than 50 MeV).
- Online reconstruction of hadron contribution to the signal is required to be implemented in current FPGA (Xilinx Spartan 3).
- Xilinx Spartan 3 is an outdated FPGA and Xilinx Integrated Synthesis Environment software tool for HDL designs has been discontinued in 2012.



hadron-like pulse shape

Electron-photon and hadron approximation algorithm





Fit function:

$y_i = AF(t_i - t_0) + P,$

A – signal amplitude, t_0 - signal start time, P – ADC pedestal value

Minimization:

$$\chi^2 = (y_i - AF(t_i - t_0) - P)S_{ij}^{-1}\left(y_j - AF\left(t_j - t_0\right) - P\right) \rightarrow min$$

System of linear equations:

for small amplitudes:

$$\begin{pmatrix} A \\ P \end{pmatrix} = \boxed{\begin{pmatrix} \Sigma_i FG41_i^k y_i \\ \Sigma_i FG43_i^k y_i \end{pmatrix}}$$

$$\begin{pmatrix} A \\ B \\ P \end{pmatrix} = \boxed{\begin{pmatrix} \Sigma_i FG31_i^k y_i \\ \Sigma_i FG32_i^k y_i \\ \Sigma_i FG33_i^k y_i \end{pmatrix}}$$

Fit function:

$$F(t) = A_{photon}R_{photon}(t - t_0) + A_{hadron}R_{hadron}(t - t_0) + P$$

where $A_1 = \frac{A_{photon} + A_{hadron}}{2}$, $A_2 = \frac{A_{photon} - A_{hadron}}{2}$

Minimization:

$$\chi^2 = (y_i - F(t))S_{ij}^{-1}(y_j - F(t_j)) \rightarrow min$$

Systems of linear equations:

$$\begin{pmatrix} A_1 \\ A_2 \\ B \\ P \end{pmatrix} = \boxed{\begin{pmatrix} \Sigma_i FG51_i^k y_i \\ \Sigma_i FG52_i^k y_i \\ \Sigma_i FG53_i^k y_i \\ \Sigma_i FG33_i^k y_i \end{pmatrix}}$$

DSP coefficients for pulse reconstruction

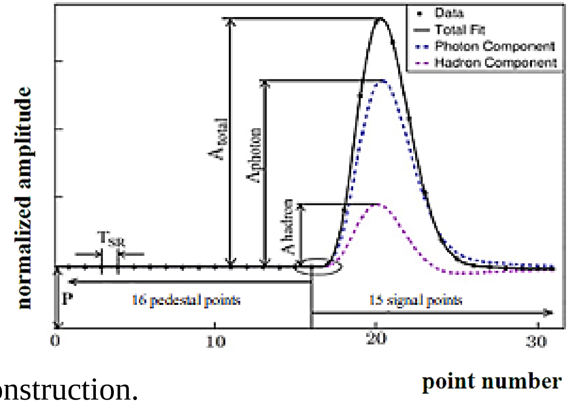
Shape fitting algorithm in FPGA

- To implement in FPGA the fitting algorithm we need to perform the following steps:

  1. Original fitting algorithm (electron-photon component) – shape fitting for the specific time $t_0 = t_{tr}$, getting amplitude A, B $= A\Delta t$.
  2. electron-photon component + hadron component algorithm. Calculating parameters A, A_1, A_2, B, P, $t_{new} = t_0 - \frac{B}{A_1}$;
  3. Define data quality using the calculated χ^2 function.
  4. Define hadron fraction:

$$A_{photon} = A_1 + A_2,\ A_{hadron} = A_1 - A_2,\ A_{total} = A_{photon} + A_{hadron};$$

$$\boxed{\text{Hadron Intensity} = \frac{A_{hadron}}{A_{total}}}$$



- If the amplitude A calculated at the first step is too small then it is decided to omit time reconstruction.

FPGA design goals and design optimization

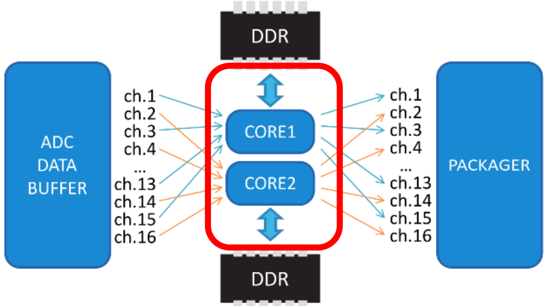
- The ShaperDSP should run two algorithms in parallel: the original signal processing and the new algorithm for hadron identification;
- Fit logic to the current chip => the bottleneck of implementation was limited resources that has left.
- Adding of hardware resources (BRAM, MULT) increase significantly time for simulation model behavior.

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	14,508	26,624	54%	
Number of 4 input LUTs	19,363	26,624	72%	
Number of occupied Slices	11,930	13,312	89%	
Total Number of 4 input LUTs	19,699	26,624	73%	
Number used as logic	16,284			
Number used as a route-thru	336			
Number used for Dual Port RAMs	2,658			
Number used as Shift registers	421			
Number of bonded IOBs	167	333	50%	
IOB Flip Flops	228			
IOB Master Pads	18			
IOB Slave Pads	18			
Number of RAMB16s	30	32	93%	
Number of MULT18X18s	20	32	62%	
Number of BUFGMUXs	7	8	87%	
Number of DCMs	3	4	75%	
Number of RPM macros	1			
Average Fanout of Non-Clock Nets	3.20			

estimated FPGA utilization with logic (for about 28% of free logic resources)

limited and precious hardware resources

MULT on LUTs

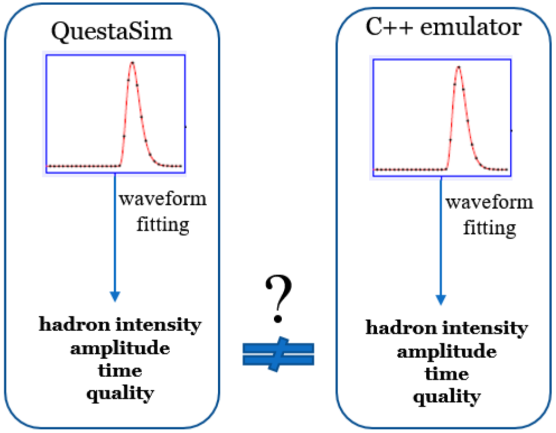


initially two cores for computations

x2

BRAM, MULT

Test environment

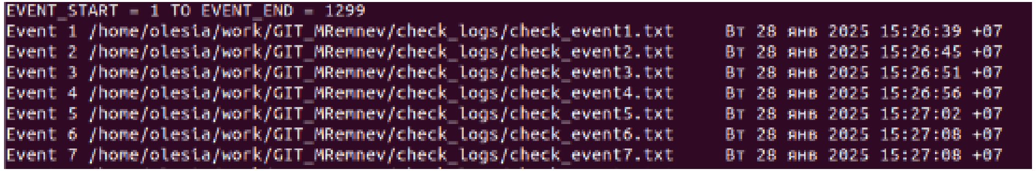


- Test environment of ShaperDSP design consists of:
  - QuestaSim with raw ADC data;
  - C++ Shaper electronics emulator;
- Verification of FPGA fitting results. It has to be consistent!

- Bash script was written for automatic simulation of big number of events with saving reports in log or text files.

- The calculation results were checked using two files with a set of events from the detector:

C++ emulator/ events	Number of events	Comment
exp0012_run02621_testpulse_data_new_time_algo.dat	498	A set of events with original + hadron algorithm
GoodEvents.dat	30000	Different set of events – boundary check (too big or too small amplitude), original + hadron algorithm



Final output result of fitting algorithm

- **Results of FPGA simulation and software emulator are consistent.**
- As well as fitting result, firmware output data also contain waveform data, trigger phase, information regarding active channels of calorimeter and etc.

QuestaSim:

fit quality	Time	Amp	Hadron intensity
2 bit	11 bit	14 bit	5 bit

trigger time

final firmware output fitting result

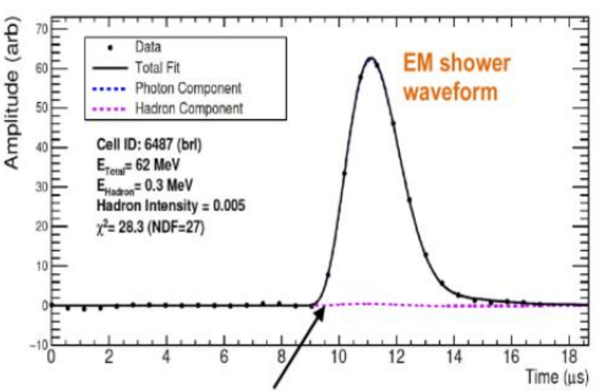
C++ emulator:

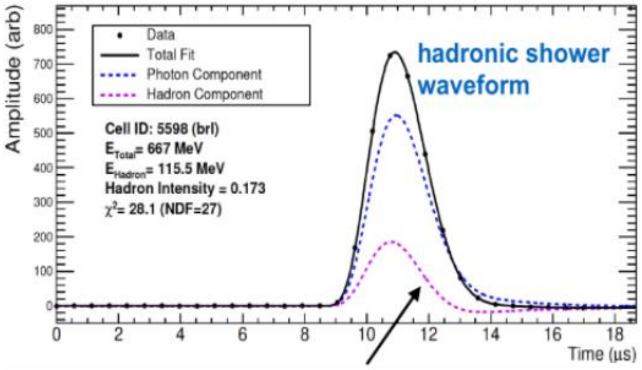
```
Expected output packet: E29C.226C
```

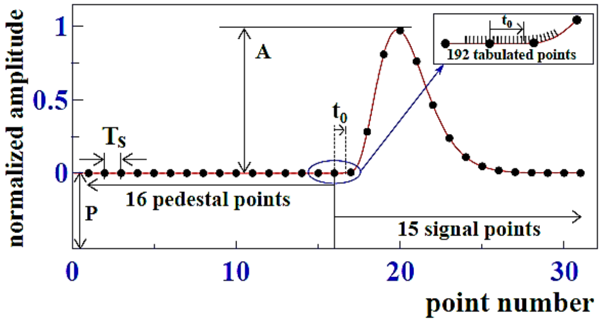
Conclusion

- FPGA implementation of shape fitting algorithm with electron-photon and hadron components is ready.
  - ✓ *Preparation and uploading of required algorithm parameters;*
  - ✓ *Encoders for calculated values of reconstructed amplitude and time;*
- Firmware was tested with software emulator. Output results are consistent.
- The next stage - work on the test stand has started.

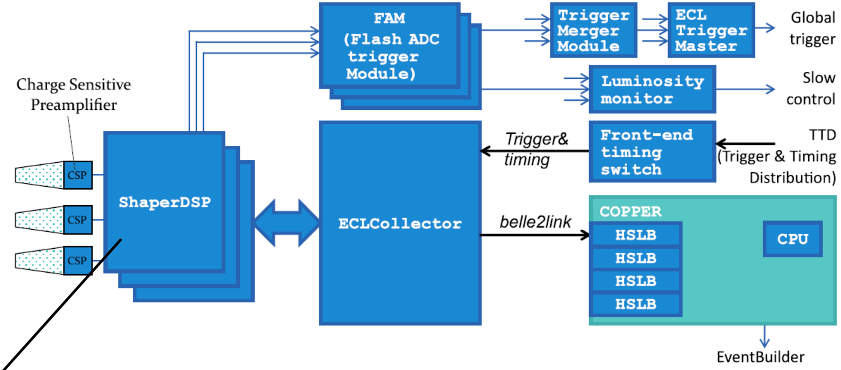
Spare slides

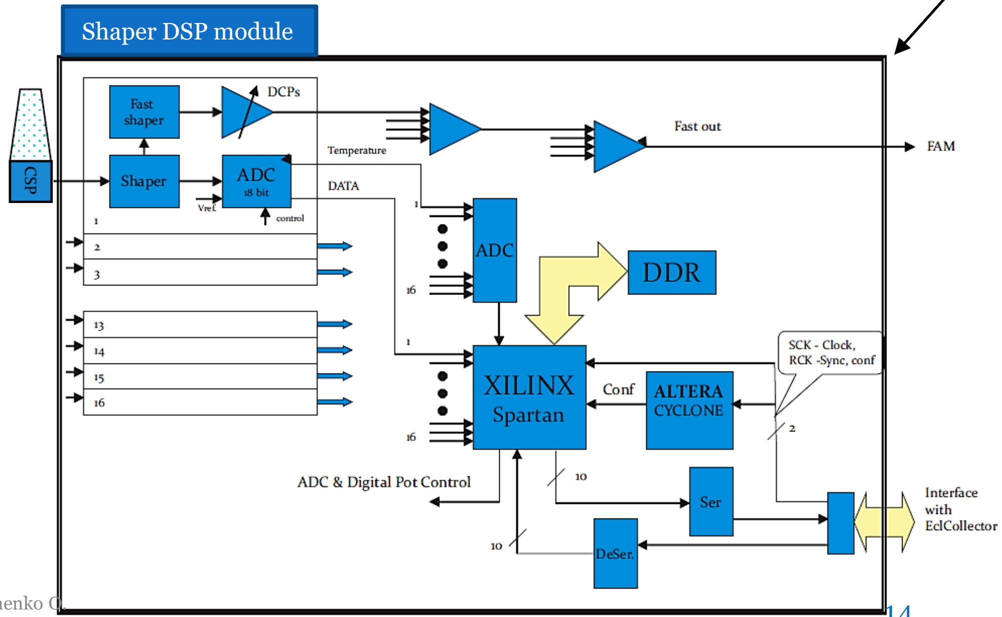

Amplitude (arb)
Data
Total Fit
Photon Component
Hadron Component
EM shower
waveform
Cell ID: 6487 (brl)
E_{Total}= 62 MeV
E_{Hadron}= 0.3 MeV
Hadron Intensity = 0.005
χ^2= 28.3 (NDF=27)
Time (μs)

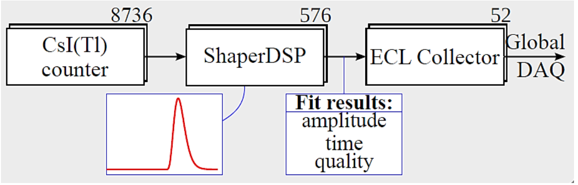

Amplitude (arb)
Data
Total Fit
Photon Component
Hadron Component
hadronic shower
waveform
Cell ID: 5598 (brl)
E_{Total}= 667 MeV
E_{Hadron}= 115.5 MeV
Hadron Intensity = 0.173
χ^2= 28.1 (NDF=27)
Time (μs)


normalized amplitude
A
t_0
T_S
P
16 pedestal points
15 signal points
192 tabulated points
point number

ECL data acquisition system







Output data package

- Format of output data:

fit quality	Time	Amp	Hadron intensity
2 bit	11 bit	14 bit	5 bit

- Amplitude and time packing:

Amp

11 bit k	3 bit p

T

9 bit k	2 bit p

$$A = \begin{cases} k, & p = 0 \\ 2^{(10+p)} + k * 2^{(p-1)}, & p = 1..7, \end{cases}$$

k – mantissa with value from 0 to $2^{11} - 1$, p – order of number.

$$\text{Hadron Intensity} = \frac{A_{hadron}}{A_{total}} \in [-0.02:0.6]$$

$$T = \begin{cases} (-1)^s * k, & p = 0 \\ (-1)^s * \left(2^{7+p} + k * 2^{(p-1)}\right), & p = 1..3, \end{cases}$$

s – sign of a number, k – mantissa with value from 0 to $2^8 - 1$, p – order of number.

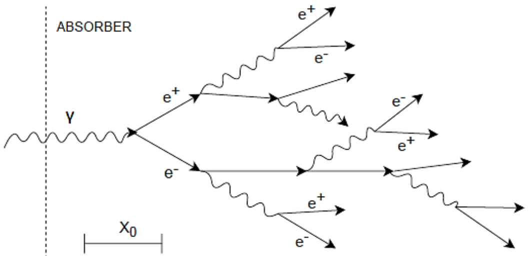
- Relative error δ_A for amplitude and δ_T for time:

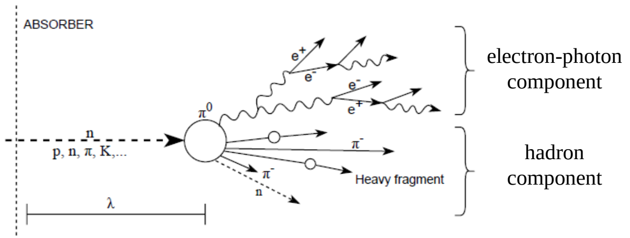
$$\delta_A = \pm 0.02\%$$

$$\delta_T = \pm 0.2\%,$$

=> significantly less compared to the energy and time resolution of the calorimeter.

Electron-photon and hadron approximation algorithm





Fit function:

$y_i = AF(t_i - t_0) + P,$

A – signal amplitude, t_0 - signal start time, P – ADC pedestal value

Minimization:

$$\chi^2 = (y_i - AF(t_i - t_0) - P)S_{ij}^{-1}\left(y_j - AF\left(t_j - t_0\right) - P\right) \rightarrow min$$

Final equation:

$$\chi^2 = \left(y_i - AF_i^k - BF_i'^k - P\right)S_{ij}^{-1}\left(y_j - AF_j^k - BF_j'^k - P\right) \rightarrow min,$$

where B = AΔt.

System of linear equations:

for small amplitudes:

$$\begin{pmatrix} A \\ P \end{pmatrix} = \boxed{\begin{pmatrix} \Sigma_i FG41_i^k y_i \\ \Sigma_i FG43_i^k y_i \end{pmatrix}}$$

$$\begin{pmatrix} A \\ B \\ P \end{pmatrix} = \boxed{\begin{pmatrix} \Sigma_i FG31_i^k y_i \\ \Sigma_i FG32_i^k y_i \\ \Sigma_i FG33_i^k y_i \end{pmatrix}}$$

Fit function:

$$F(t) = A_{photon}R_{photon}(t - t_0) + A_{hadron}R_{hadron}(t - t_0) + P =$$

$$= (A_{photon} + A_{hadron})(R_{photon} + R_{hadron})/2 + (A_{photon} - A_{hadron})(R_{photon} - R_{hadron})/2 + P$$

Minimization:

$$\chi^2 = (y_i - F(t))S_{ij}^{-1}(y_j - F(t_j)) \rightarrow min$$

Final equation:

$$\chi^2 = \left(y_i - A_1R_{+i}^k - A_2R_{-i}^k - BR_{+i}'^k - P\right)S_{ij}^{-1}\left(y_j - A_1R_{+j}^k - A_2R_{-j}'^k - BR_{+j}'^k - P\right) \rightarrow min,$$

where $A_1 = \frac{A_{photon} + A_{hadron}}{2}$, $A_2 = \frac{A_{photon} - A_{hadron}}{2}$, B $= -A_1\Delta t$.

Systems of linear equations:

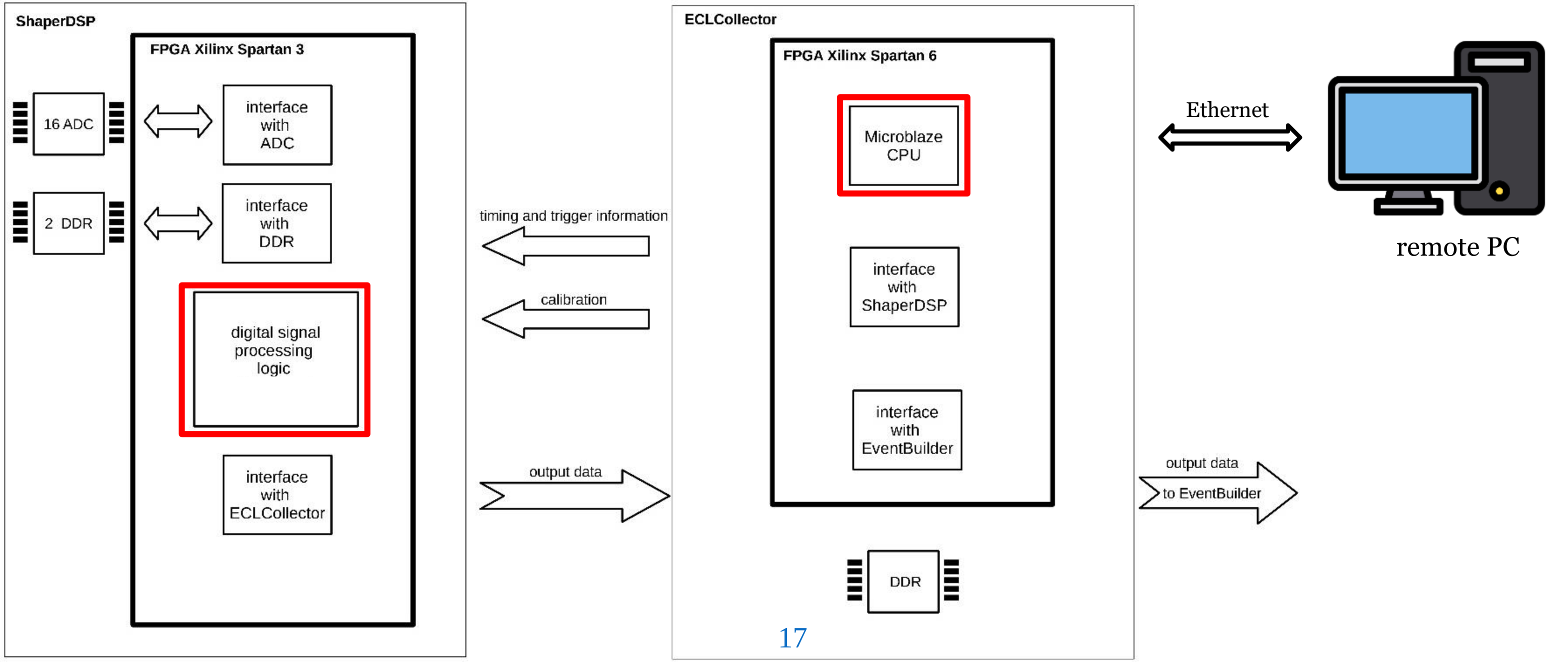
DSP coefficients for pulse reconstruction

$$\begin{pmatrix} A_1 \\ A_2 \\ B \\ P \end{pmatrix} = \boxed{\begin{pmatrix} \Sigma_i FG51_i^k y_i \\ \Sigma_i FG52_i^k y_i \\ \Sigma_i FG53_i^k y_i \\ \Sigma_i FG33_i^k y_i \end{pmatrix}}$$

ShaperDSP and ECLCollector

- ShaperDSP electronic module with FPGA Xilinx Spartan 3 on-board digitizes and fits the input signal coming from ECL counters.

- Interface with ECL Collector provides timing and trigger information and is also used for initial initialization and upload of DSP coefficients pre-calculated at the remote PC. These DSP coefficients are derived from pulse shape and noise covariance matrix of each channel.



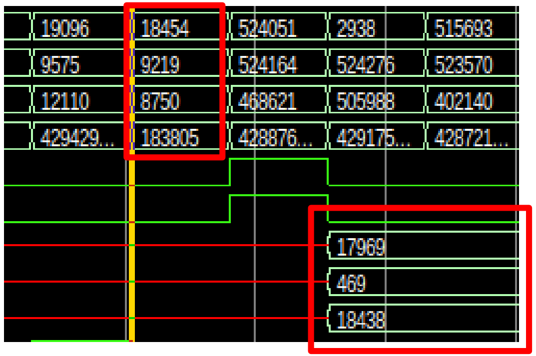
ShaperDSP design. Example of hadron component calculation

- The obtained amplitude value is recorded in ADC units (1 ADC channel ≈ 0.05 MeV).
- Example for 1 active channel:

$A = 18454;\ A_1 = 9219;\ A_2 = 8750;\ B = 183805;\ A_{photon} = 17969;\ A_{hadron} = 469;\ A_{photon} + A_{hadron} = 18438.$

QuestaSim:

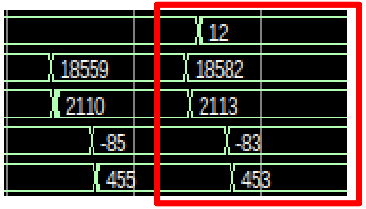
Amplitude A
Amplitude A_1
Amplitude A_2
Parameter B



A_{photon} component
A_{hadron} component
$A_{photon} + A_{hadr}$

Packed values:

integer hadron fraction
shifted amplitude A
encoded amplitude value
time estimation
encoded time value



Shaper emulator:

```
HEX_packAmplitude(18582) = 2113
HEX_packTIme(-83) = 0453
```

packed amplitude and time values

```
hadron_frac = 12
```

integer hadron fraction

```
h_amp / (double)A_total = 0.0254366
```

exact value of Hadron intensity

```
unpackHadronFraction(hadron_frac) = 0.03125
```

unpacked value of Hadron intensity

Uploading coefficients for each stage of calculations

it0
(fg41)
16 coefficients

it1
(fg31, fg32)
32 coefficients

it2,3
(fg31,fg32,fg51,fg52,fg53)
80 coefficients

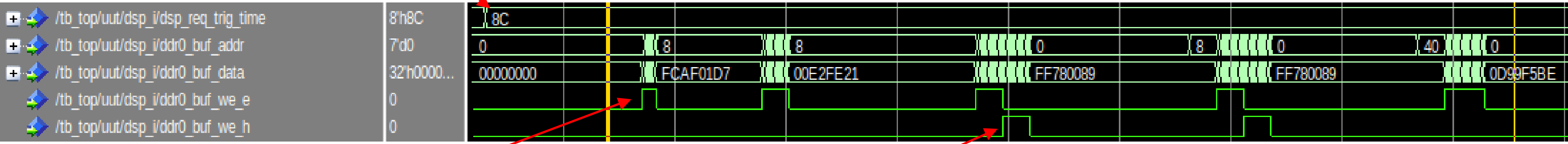
it4
(f, f1, fg33)
48 coefficients

trigger time

Amplitude calculation
with $t_0 = t_{tr}$

Amplitude calculation of a signal
with t_{new}, signal start parameter B

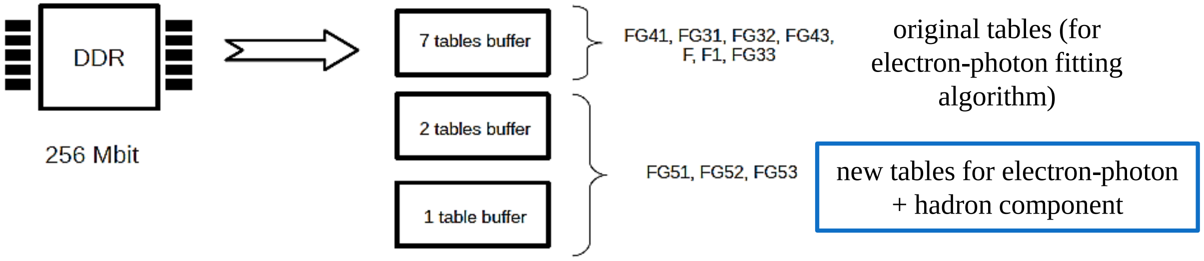
data quality (χ^2)



uploading coefficients for electron-photon
reconstruction algorithm

uploading coefficients for hadron component
reconstruction algorithm

Additional memory buffers. Rearrangement of tables.



Code of the table	Name of the table	
"0000"	FG41[16][24]	Array of $FG41_i^k$
"0001"	FG31[16][192]	Array of $FG31_i^k$
"0010"	FG32[16][192]	Array of $FG32_i^k$
"0011"	FG51[16][192]	Array of $FG51_i^k$
"0100"	FG52[16][192]	Array of $FG52_i^k$
"0101"	FG53[16][192]	Array of $FG53_i^k$
"0110" - "1011"	reserved	
"1100"	FG43[16][24]	Array of $FG43_i^k$
"1101"	F[16][192]	Array of F_i^k
"1110"	F1[16][192]	Array of F'^k_i
"1111"	FG33[16][192]	Array of $FG33_i^k$

- Implementation in ShaperDSP sequential writing to buffers and parallel reading of DSP coefficients;
- 2 independent memories were added in parallel mode to avoid increasing computation time also having more available data at the same time;
- Set of registers and its values were reinitialized in ECL Collector CPU;