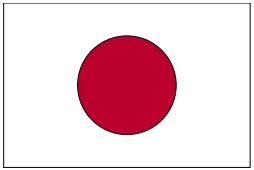
Monte Carlo processor for calibrating air shower experiments

Marcos Anzorena (ICRR) for the ALPACA collaboration
anzorena@icrr.u-tokyo.ac.jp

計測システム研究会2024@東大

ALPACA collaboration



More than 50 collaborators.
24 academic institutions.

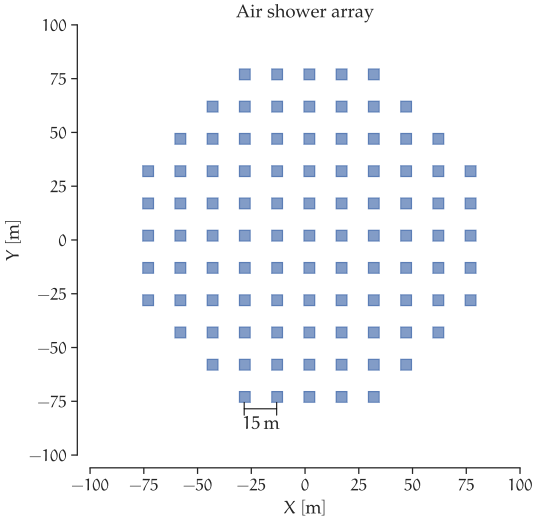
Andes Large Area PArticle detector for Cosmic ray physics and Astronomy

Site coordinates: 4740 m, 16°23′**S**, 68°8′**W**.

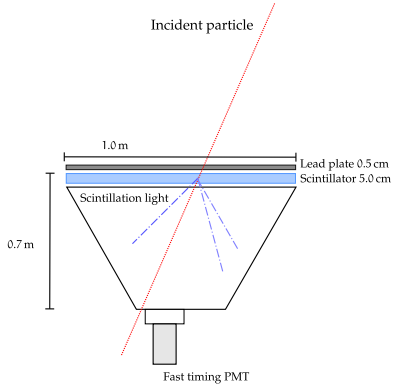
Current status: ALPAQUITA air shower array w/ 97 1 m^2 detectors.

ALPACA experimental site, June 2024

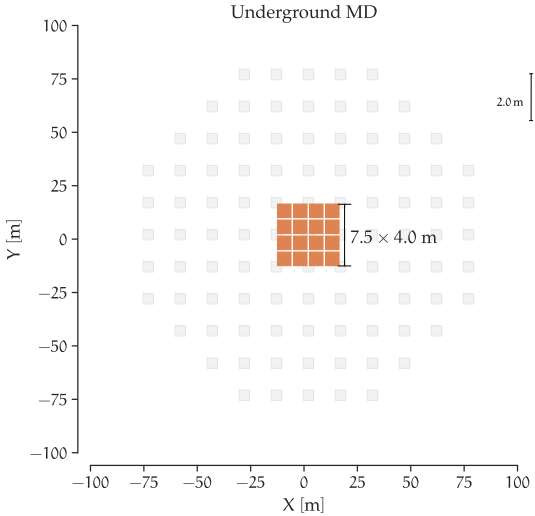
Experimental technique: Surface array detector

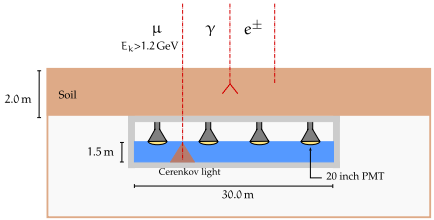


- Area coverage: 18 450 m^2
- Number of elements: 97
- Single-particle peak: 9.4 MeV



Experimental technique: Underground muon detector





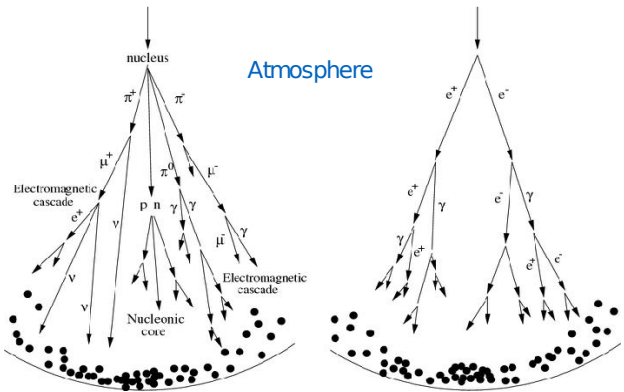
- Area coverage: 900 m^2
- Number of elements: 16 cells.
- Single-muon peak: 24 pe*

* S.Kato et al., Experimental Astronomy (2021) 52:85-107

Experimental technique: γ-ray/hadron discrimination

TeV-PeV proton, helium

TeV-PeV γ-ray

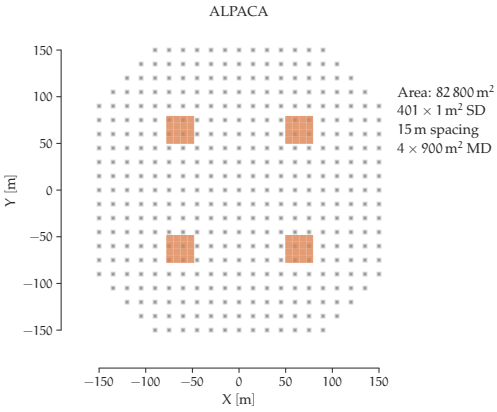


Number of μ's within < 100 m from the core

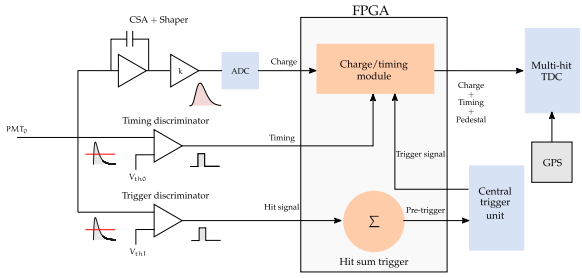
$\sim 50\mu$ for 100 TeV proton

$\sim 1\mu$ for 100 TeV γ-ray

ALPACA's DAQ system

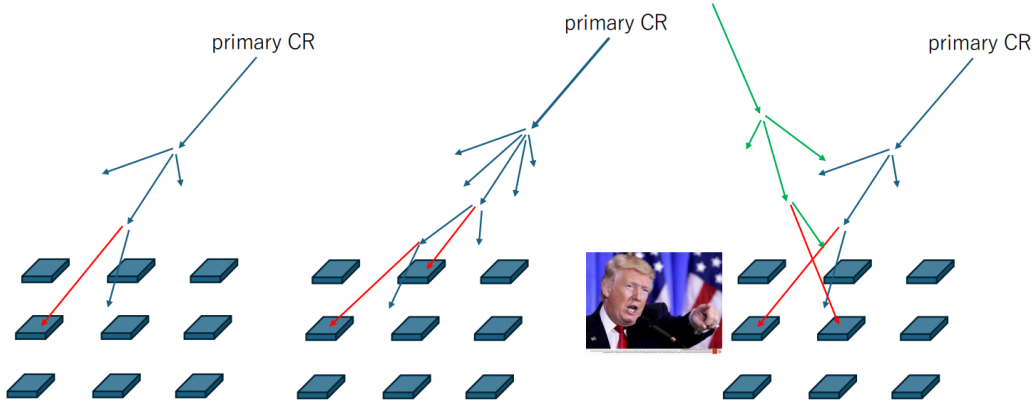


Full ALPACA array



ALPACA DAQ system

Contamination by accidental hits



Any1: events with signal in a single counter

Any2: events with signals in two counters, originating from a same origin (primary CR)

Fake Any2: events with signals in two counters originating from unrelated origins (two primary CRs)

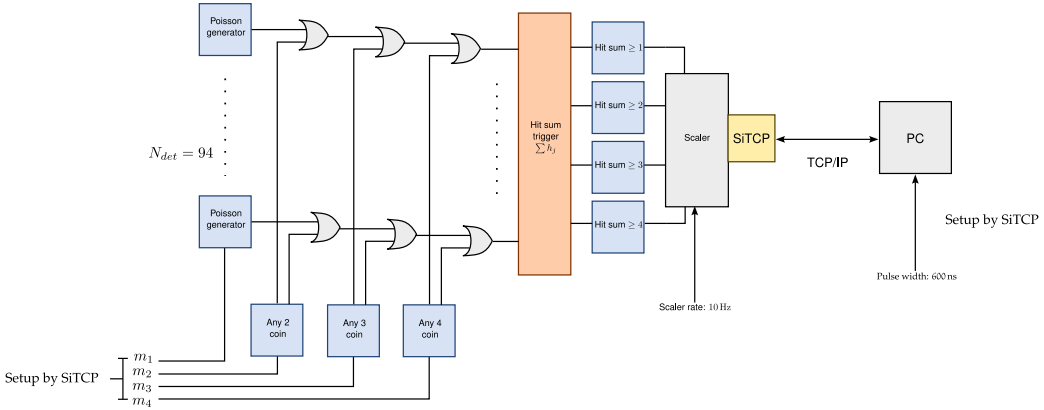
How to correct scaler data from accidental hits?

MC method

- 97 detectors independently generate *hits* randomly according to a Poisson distribution.
- The rate of the Poisson is: $T_{win} * m_0$.
- T_{win} is the time window of the coincidence 1200 ns.
- m_0 is the single rate from experiment 820 Hz.
- Count the number of *hits* in each iteration of the simulation.
- Repeat N times (considering 0.1 s, $N = 0.1/T_{win}$) — counts per 0.1 second.
- Repeat 50 times (that means $50 * N$) to achieve good statistics.
- Challenging to include higher order coincidences and time resolution — computer power.

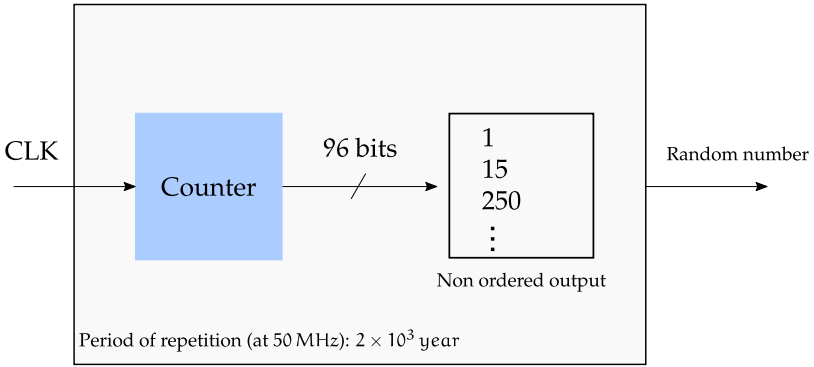
MC processor with FPGA

- 94 independent channels with rate m_1 — can be scale to larger number of channels.
- System includes double and triple coincidences.



How to produce random number generator with logic?

Principle of work: very large counter — Linear Feedback Shift register



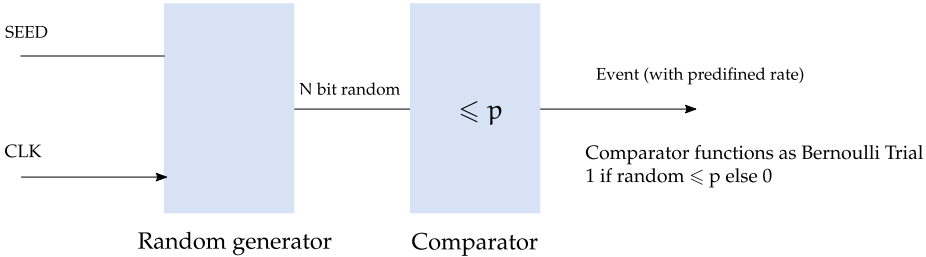
Uniform random number generator

LFSR with 32 bits output

```
process(gclk)
begin
 if rising_edge(gclk) then
  if rst='1' then
   SR<=seed(j);
  else
   SR(95 downto 32)<=SR(63 downto 0);
   for k in 0 to 31 loop
    SR(31-k)<=SR(95-k)xor SR(93-k)xor SR(48-k)xor SR(46-k);
   end loop;
  end if;
 end if;
end process;
```

How to produce Poisson signal with predefined rate?

Bernoulli trial aprox: urnd + comparator $\longrightarrow$ poisson generator



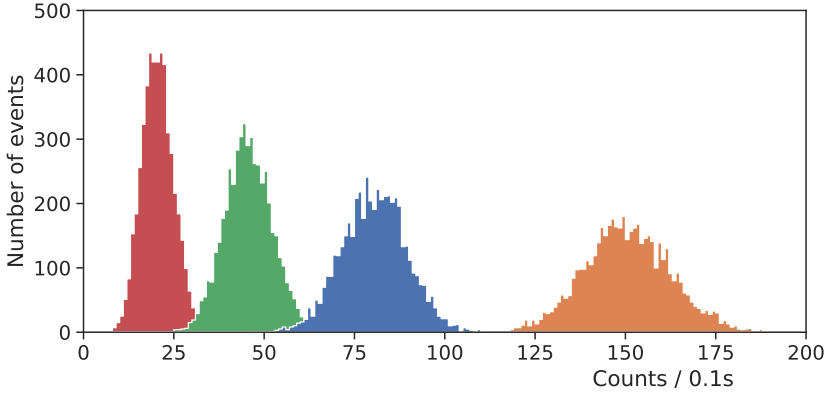
How to produce Poisson signal with predefined rate?

Bernoulli trial aprox: urnd + comparator $\longrightarrow$ poisson generator

- Rate is defined as $\lambda = N * p$
- N is the number of trials per second (clock frequency).
- Output of urnd gen and p are integers.
- p should be very small.

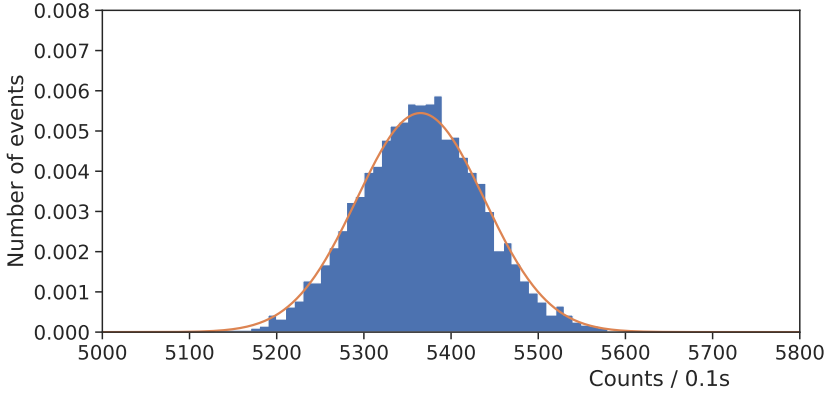
MC processor with FPGA — input parameters

- 800 events/s (blue), 1000 events/s (orange), 300 events/s (green) and 200 events/s (red).



MC processor with FPGA — results any 1 detectors

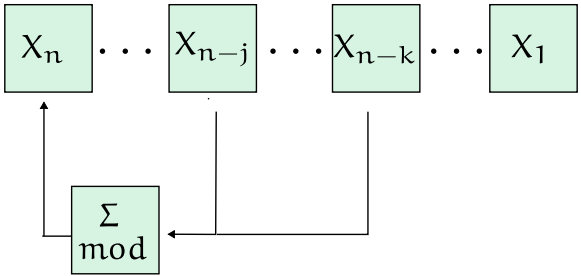
- Output rate should be close to the input rate $\longrightarrow$ 8000 events/s.
- Origin of problem — time correlation between channels.



MC processor with FPGA — solving time correlation

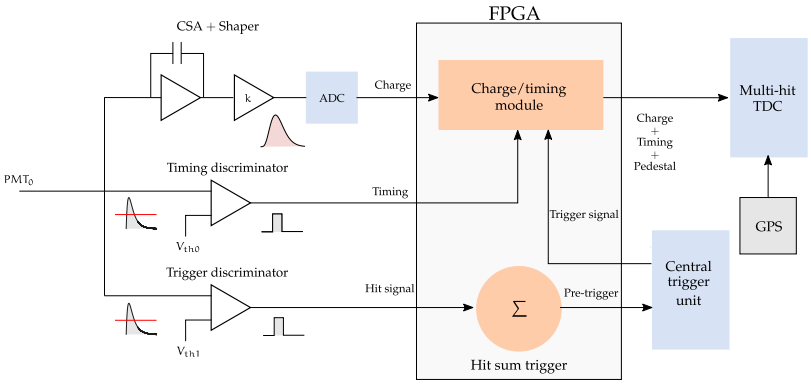
- Use Additive Lagged Fibonacci generator — large parallel output.
- Generate 127 random numbers — $j = 67$ and $k = 97$

ALFG needs to be seeded with random seed



ALPACA DAQ system revisited

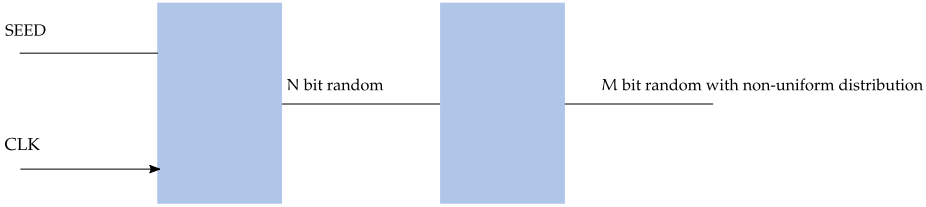
- Motivation: design advance trigger system.
- Pattern generator with timing and charge distribution is essential.



ALPACA DAQ system

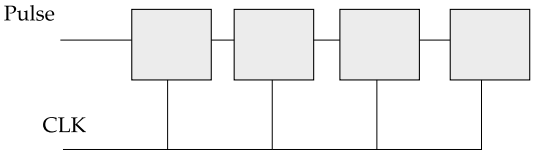
How to produce generate random with non-uniform distribution?

Inverse sampling: urnd+LUT



In general $N \neq M$. For our design $N = 16$ bits and $M = 10$ bits.
N is the quality of the random number generated.
M is the resolution of the pulse delay system.

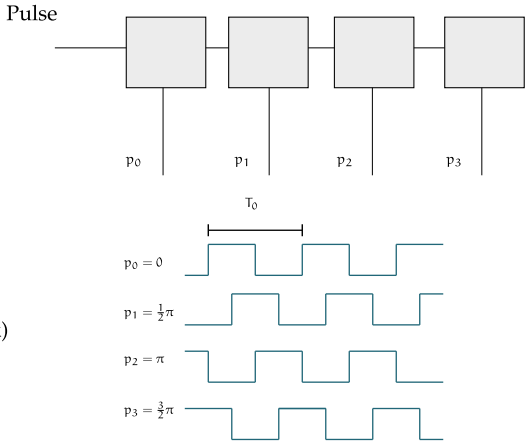
Generating delay for independent channels



Register bank

Pulse is delayed with T_0 units (total delay depends on size of bank)

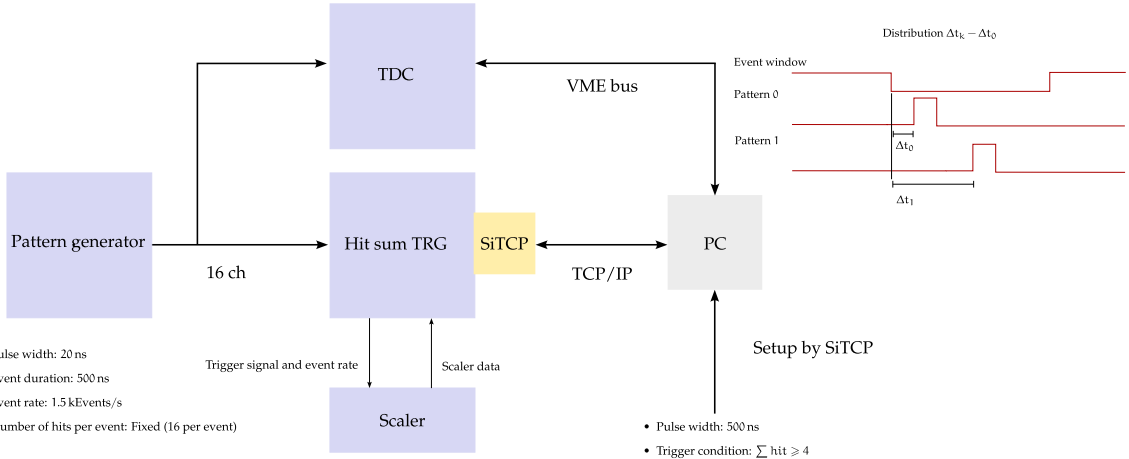
Delays with 1 ns required 1 GHz CLK



10 bit design with 250 MHz — resolution 0.5 ns, range 500 ns.

Circuit operating with both edges of the clocks

Test with 16 channel pattern generator

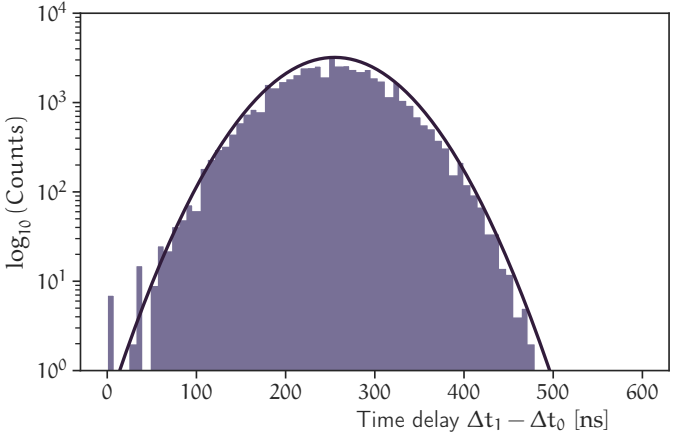


- Pulse width: 20 ns
- Event duration: 500 ns
- Event rate: 1.5 kEvents/s
- Number of hits per event: Fixed (16 per event)

- Pulse width: 500 ns
- Trigger condition: $\sum \text{hit} \geqslant 4$
- Trigger width: 1 µs

Test with 16 channel pattern generator

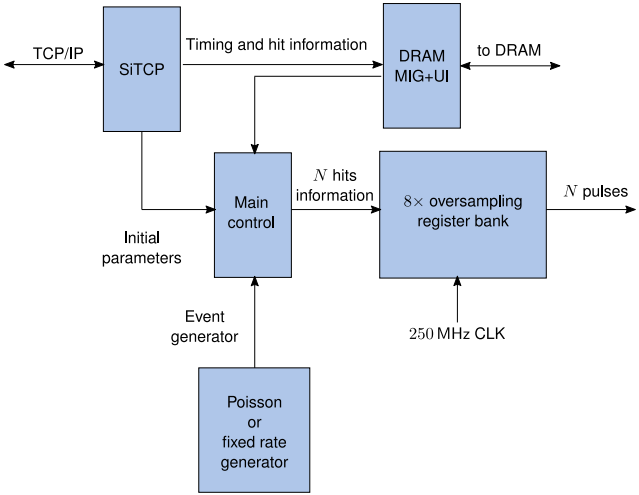
Input timing distribution: Gaussian μ =255 ns, σ =60 ns.



Good results, however ...

- Long delay banks are not good for timing performance.
- Initial length of bank: 1024 elements — 0.5 ns.
- Implemented solution: slow bank (23 elements) + fast bank (8 elements) + clever logic.
- LUT implementation in block RAM greatly affects routing and timing.
- Solution is to use external memory.

New architecture — capable of 64 channels or more



Future works

- Does MC processor capable of simulating full ALPACA detector?
- Is it possible to include higher order effects? (threshold dependence on temperature)
- Pattern generator with DRAM is in progress — use dedicated processor for control (Zynq ultrascale).
- Currently working on generating analog output considering charge distribution.
- Interesting option is to implement GAN (generative adversial network) on FPGA.