



Введение в ССД

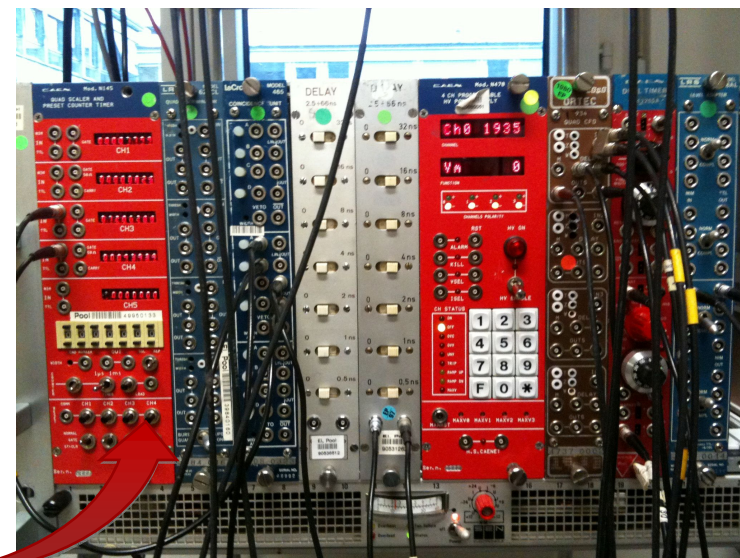
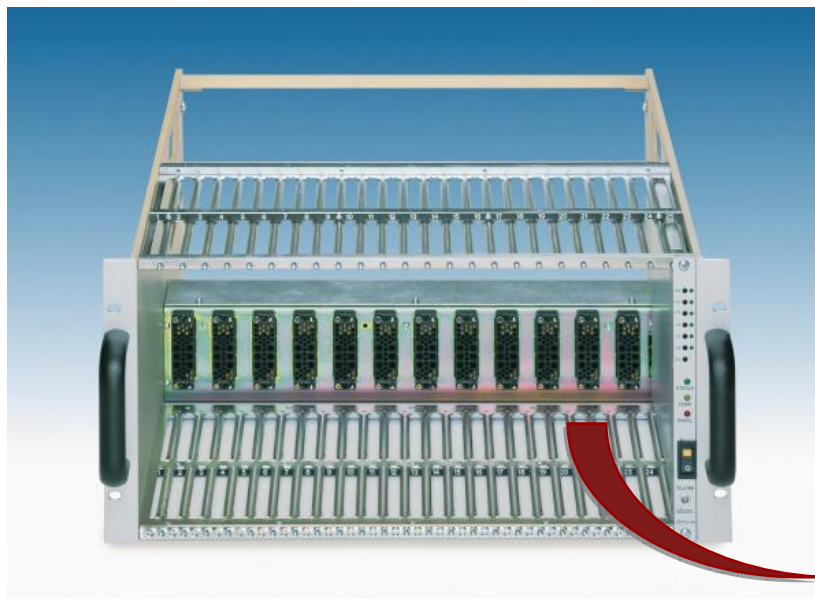
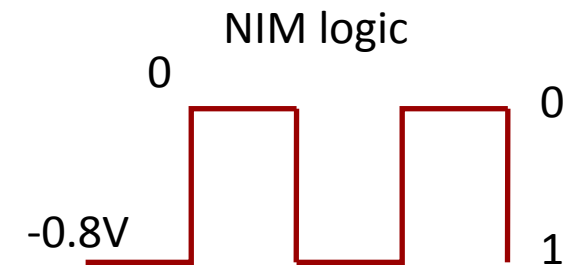
Триггерная электроника

О.Соловьянов (на основе материалов F.Pastore CERN)



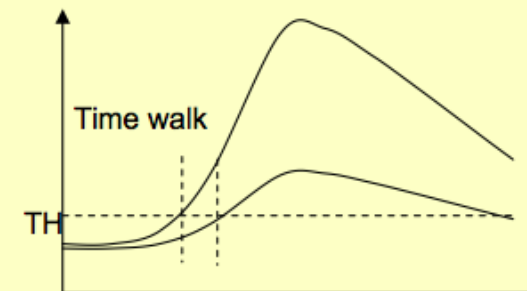
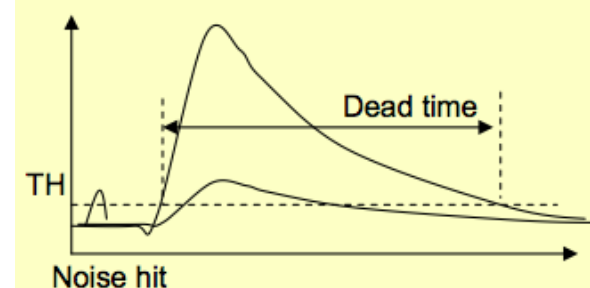
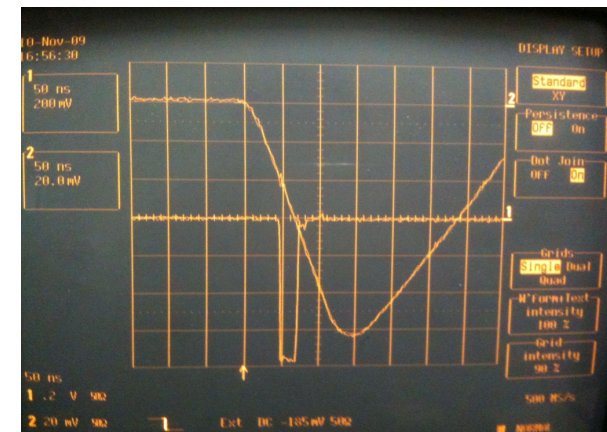
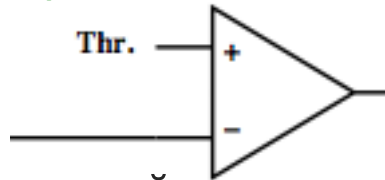
Простейшая триггерная система

- Простой триггер можно сделать на основе крейта и модулей NIM
- Электронные модули со стандартным размером, разъемом и логическими уровнями сигнала (-16 мА на 50 Ом = -0.8 Вольт)



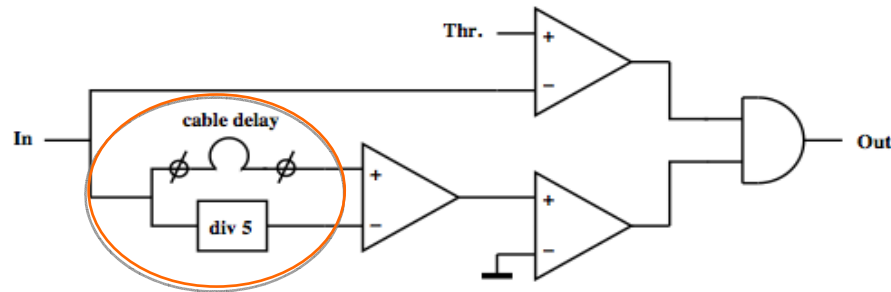
Триггер как двоичная система

- Исходные сигналы
 - Двоичные трекары (pixels, strips)
 - Аналоговые сигналы от трекаров, времяпролетных детекторов, калориметров
- Простейший триггер: пороговый
 - Посмотреть на сигнал
 - Поставить как можно более низкий порог, сигналы от детекторов имеют большой разброс по амплитуде
 - Компромисс между эффективностью и ложными срабатываниями от шума
- Ширина импульсов
 - Мертвое время ограничивает частоту
- Временной разброс
 - Время пересечения порога зависит от амплитуды сигнала



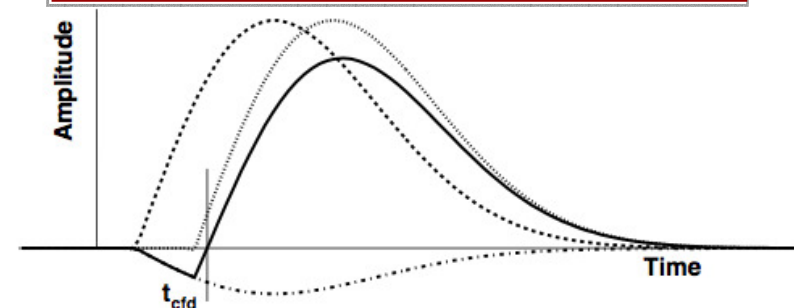
Дискриминатор со следящим порогом (CFD)

- Сцинтилляционные детекторы и ФЭУ имеют оптимальное положение срабатывания на определенной части амплитуды, обычно 10-15% от максимума
- Но есть временной разбег для сигналов с разной амплитудой (но одинаковым временем нарастания)

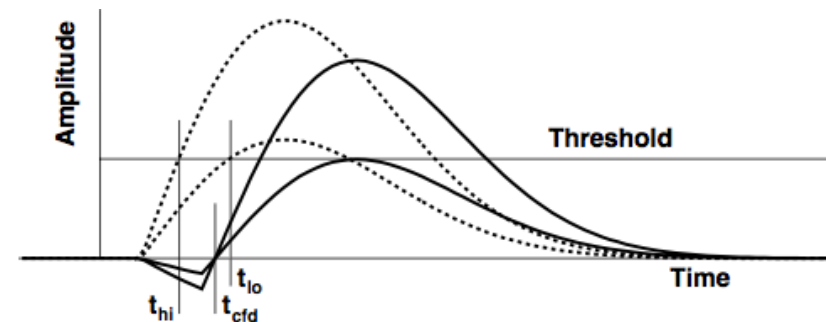


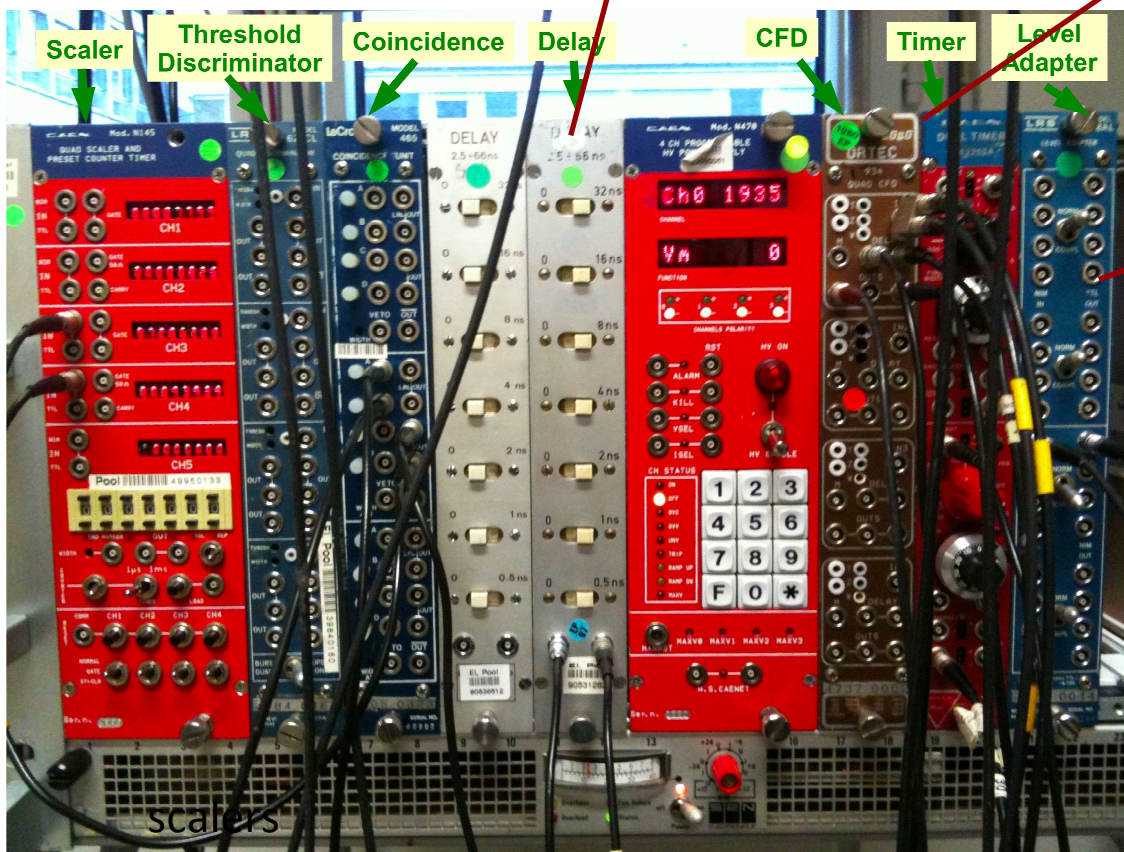
- Входной сигнал с амплитудой V распараллеливается по двум плечам.
- В одном он задерживается на время t_d , в другом инвертируется и ослабляется ($-fV$).
- Сигналы складываются.
- Формируется биполярный сигнал, пересечение нуля которым служит для получения временной отметки
- Коэффициент ослабления f определяет долю амплитуды (fraction) оптимальную для сигналов данного детектора.

- входной импульс
- задержанный
- .- инвертированный
- биполярный

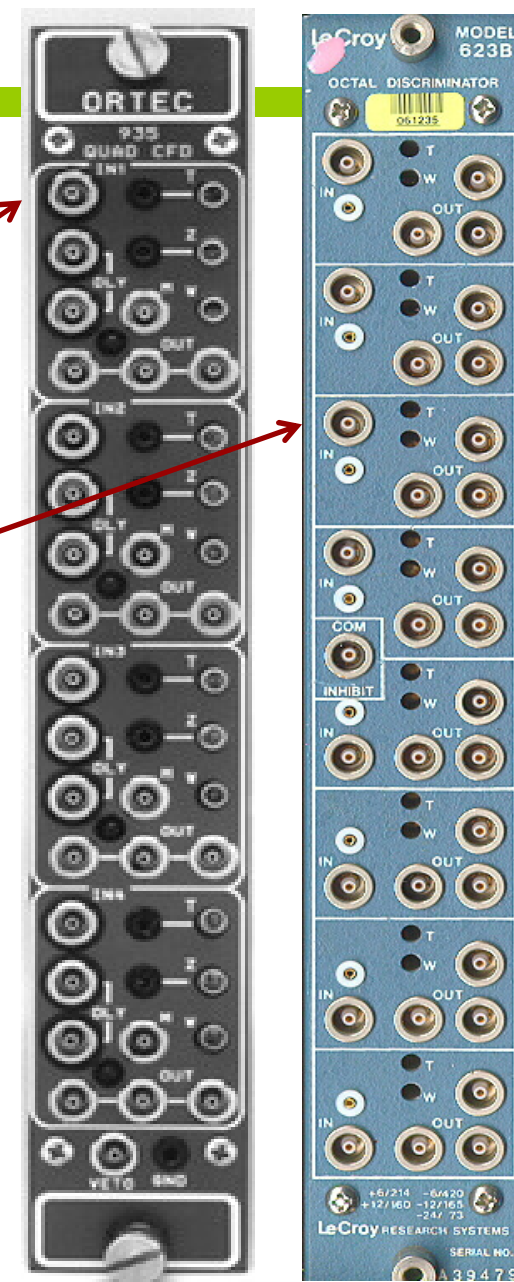


Выход CFD срабатывает когда биполярный импульс изменяет знак





➤ Пороги настраиваются отверткой



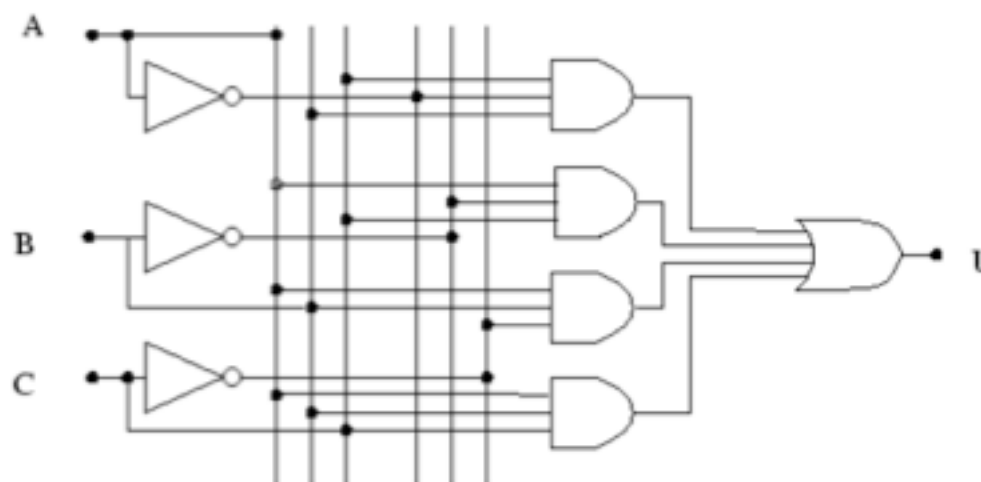
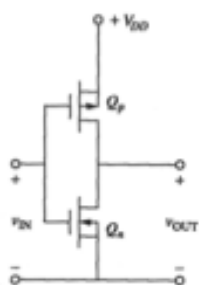
ORTEC CDF

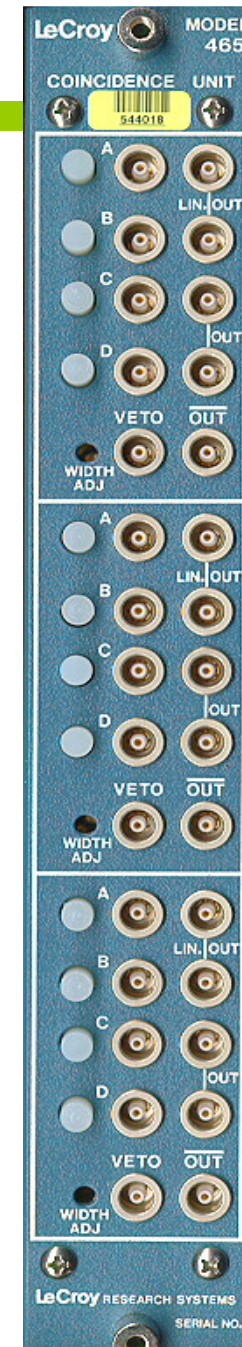
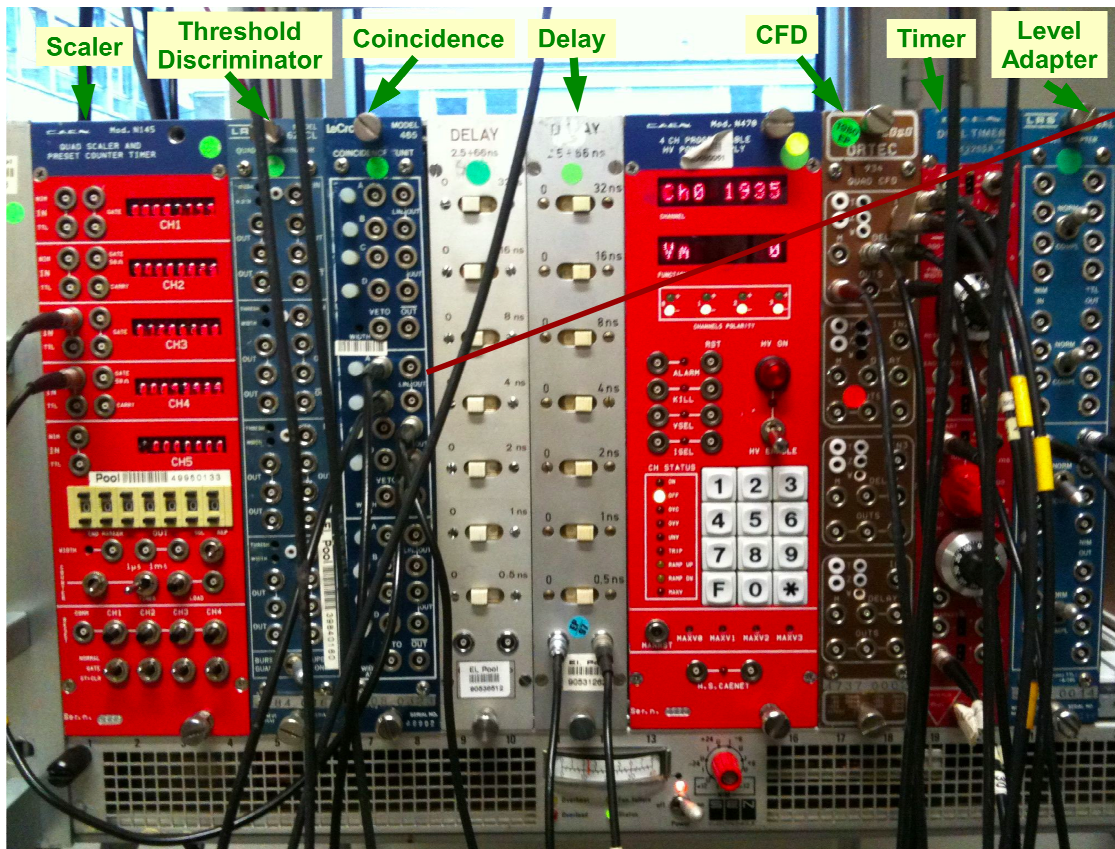
LeCroy
дискриминатор

Триггерная логика

- Аналоговые системы: усилители, фильтры, компараторы ...
- Цифровые системы:
 - Комбинаторные: сумматоры, дешифраторы, мультиплексоры, ...
 - Последовательные: флип-флоп, регистры, счетчики, ...
- Преобразователи: АЦП, ВЦП, ...

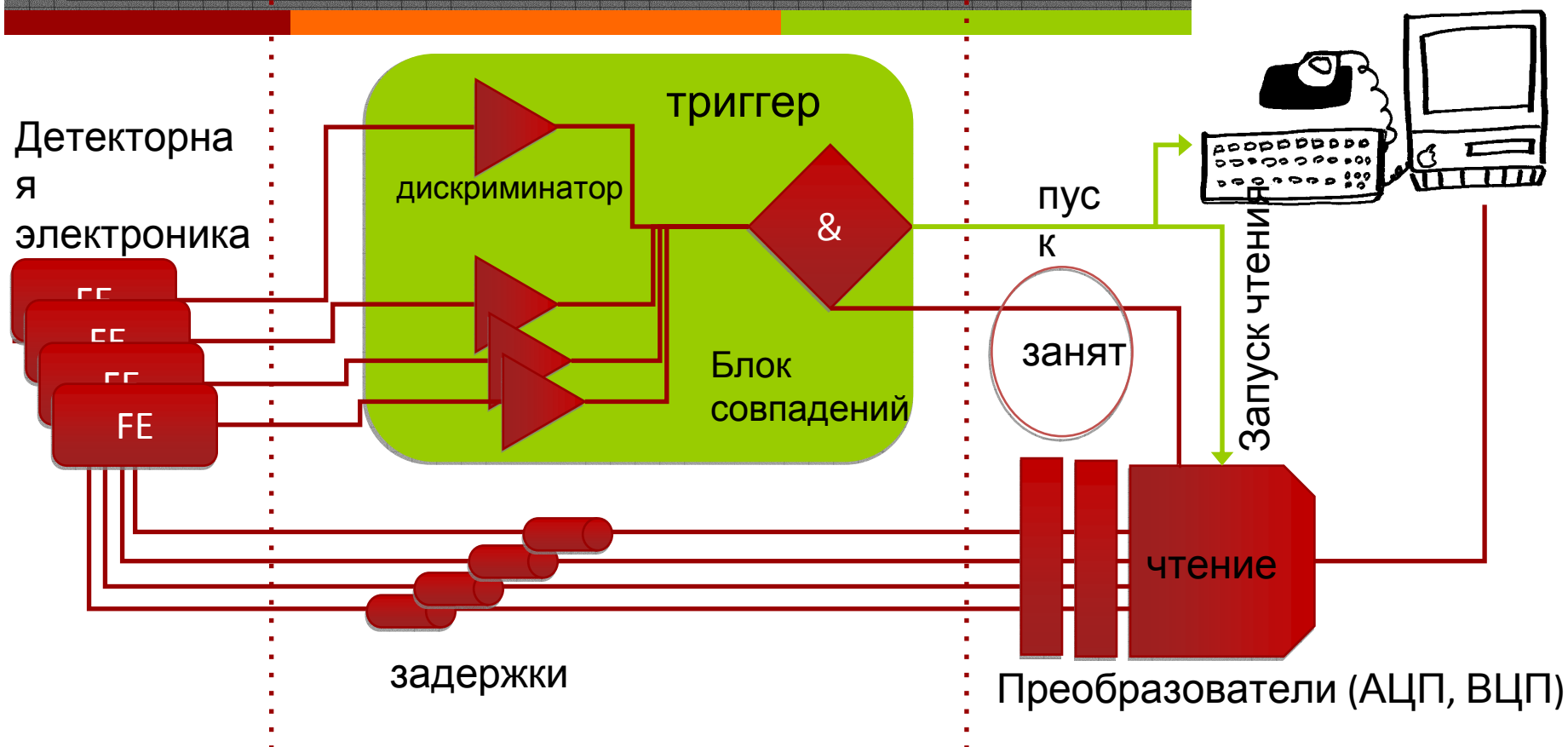
Поведение нашей системы может быть логически описано последовательностью математических операций





LeCroy блок совпадений

Простая триггерная система



- Сигнал от детектора может быть использован как для чтения так и для триггера, информация идет по двум разным путям:
 - Быстрый путь для триггерных данных, идет в ССД для запуска чтения
 - Медленный (задержанный) путь для самого чтения данных

Требования к триггеру

- Высокая эффективность
 - Малое мертвое время
 - Быстрое решение
- Надежность и устойчивость
- Гибкость

Мертвое время

- Dt = часть от общего времени набора данных в течение которого события не могут быть записаны, обычно до 10%

- **Мертвое время чтения:**

- Пока событие не будет прочитано полностью другие не могут приниматься

- **Мертвое время триггера:**

- время выработки триггерного решения

- **Операционное мертвое время:** время когда ССД не работает (сломалась, перезапускается)

По аналогии с фотоаппаратом, если мы хотим фотографировать как можно чаще, пределом будет являться скорость обработки/записи



Фото/события не могут



перекрываться
Быстрая реакция



Замена батарей



Мертвое время = потерянные события = источник неэффективности

Проблема частоты данных

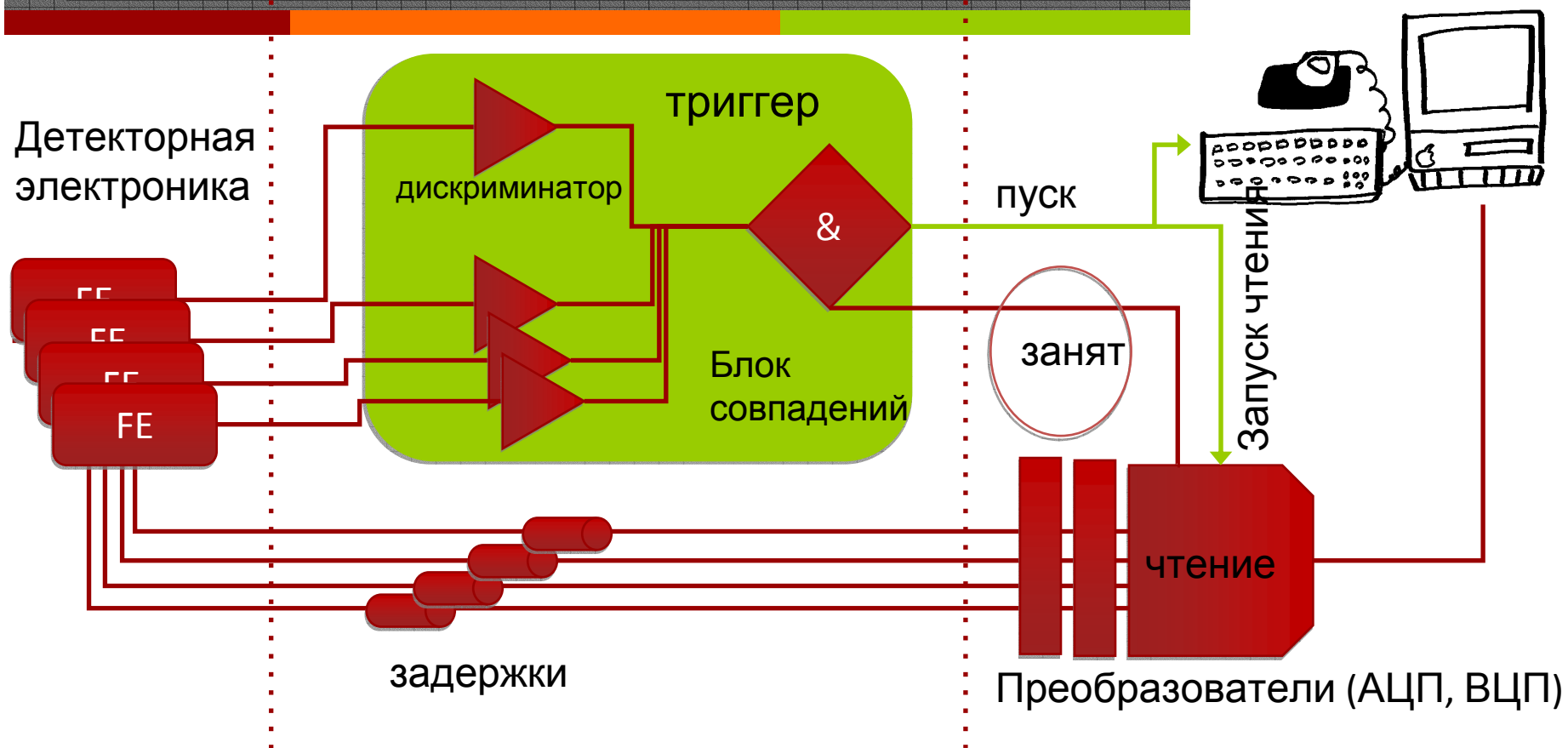
- R = число читаемых событий в секунду (частота ССД)
- R_T = частота триггера
- T_d = мертвое время
- Долевое мертвое время = $R \times T_d$, живое время = $1 - R \times T_d$ таким образом частота прочитанных событий $R = (1 - R \times T_d) \times R_T$:
- Доля триггеров не потерянных из-за мертвого времени:

$$R / R_T = 1 / [1 + R_T T_d] \quad \text{Мы хотим 100%!!!}$$

- Трюк состоит в том чтобы сделать R_T и T_d как можно меньше
- Если T_d велико, оно ограничивает максимальную частоту ССД, независимо от частоты триггера:
 - $R = 1/T_d$: если время чтения 1с, максимальная частота = 1 Hz
 - Если $R_T = 1/T_d \rightarrow$ мертвое время 50%

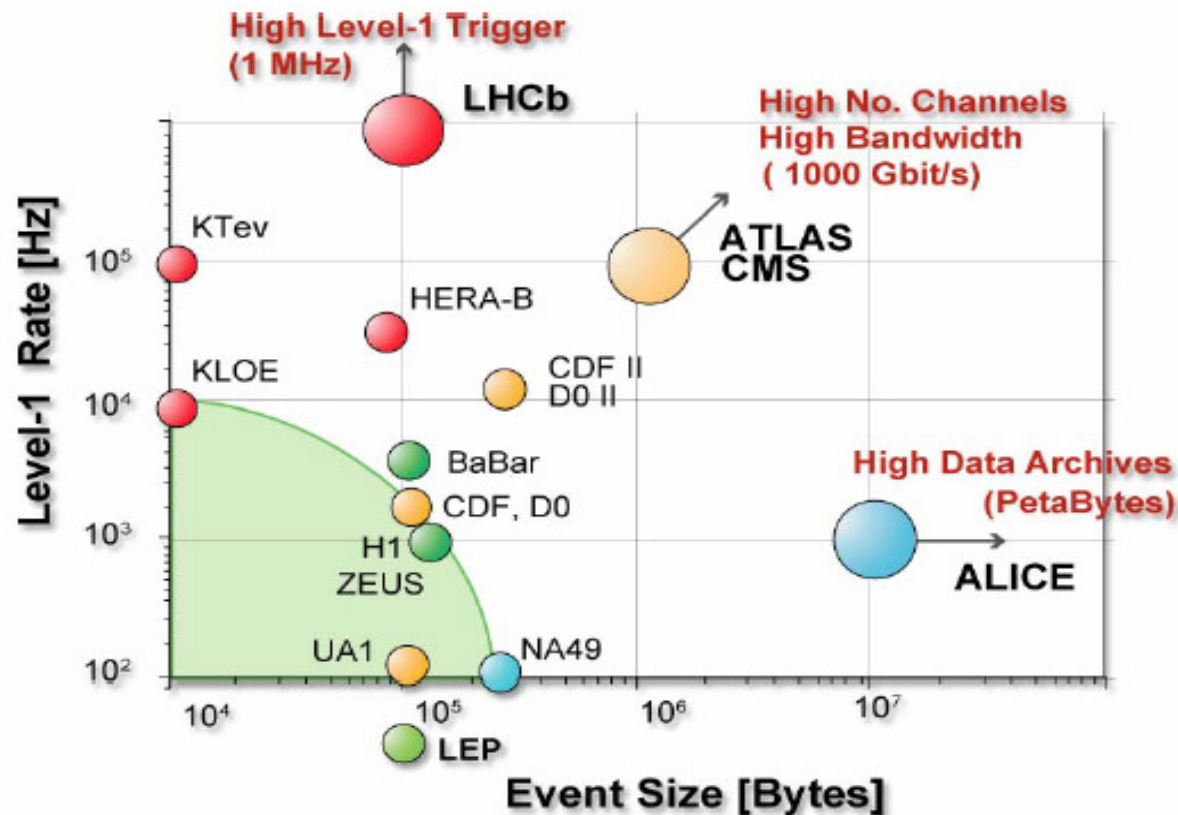
Мы всегда теряем события при $R > 1/T$

Простая триггерная система



$$D_t = R \times T_{RO}$$

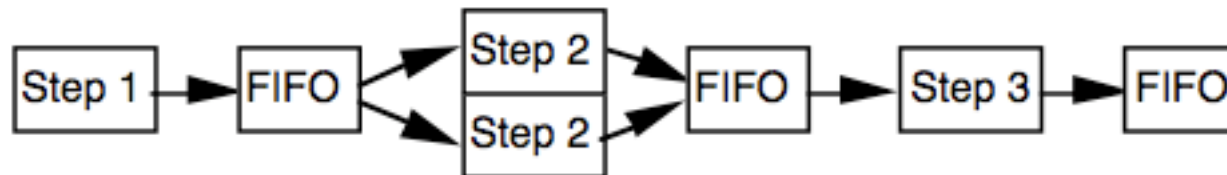
Тенденции ССД



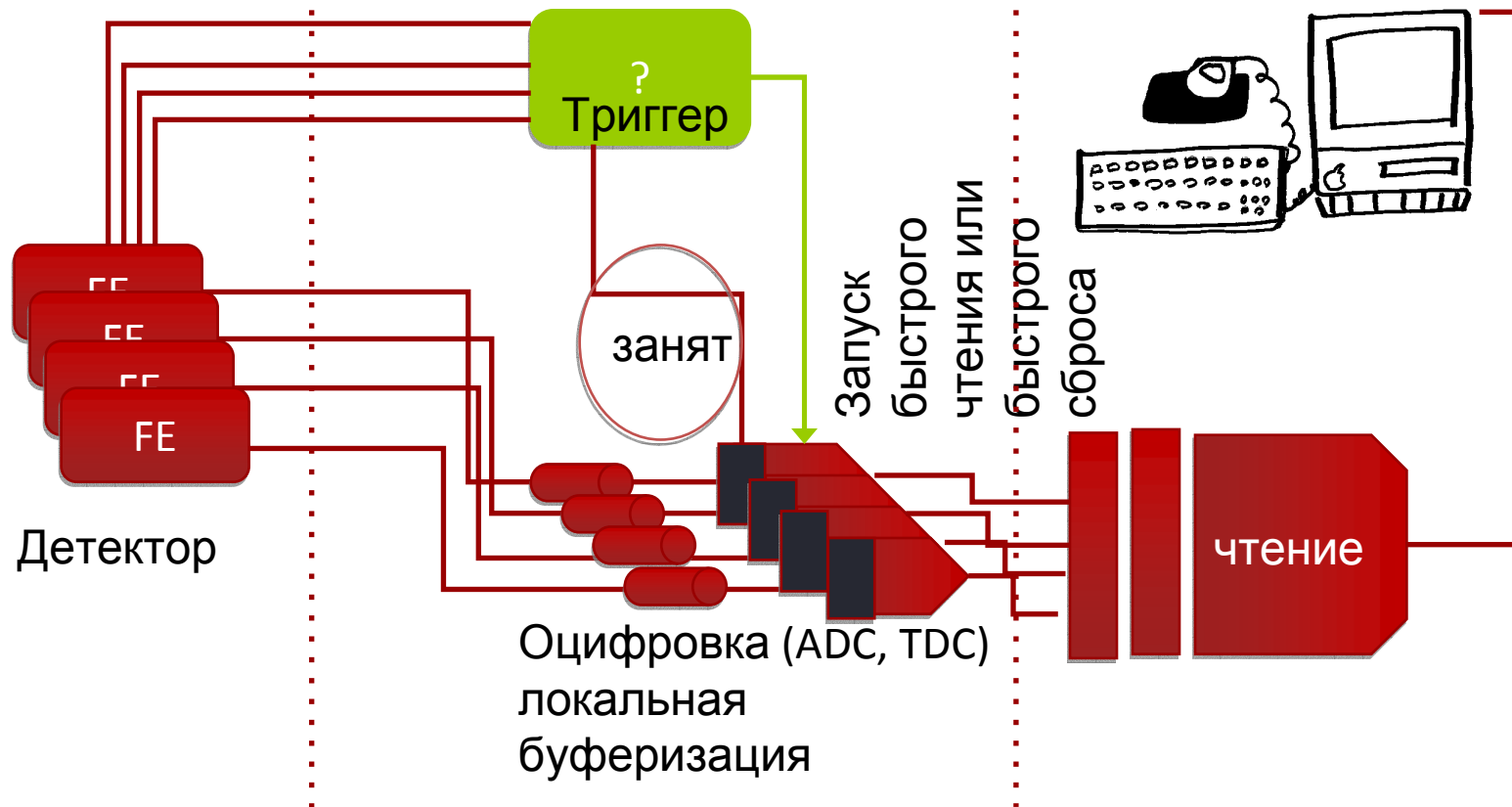
- Так как объемы данных и частоты триггеров многократно возросли, были разработаны новые архитектуры ССД

Минимизация мертвого времени чтения

- Для больших систем были разработаны два подхода
 - **Параллелизм:**
 - Независимые линии обработки триггера и сбора данных, для каждого элемента: оцифровка и триггер работают параллельно
 - **Конвейерная обработка**
 - Использование локальных буферов (FIFOs) между стадиями обработки позволяют иметь разное время обработки. Глубина локальных буферов ограничивает время обработки следующей ступени: лучше если шаг 3 быстрее шага 2



Минимизация мертвого времени чтения

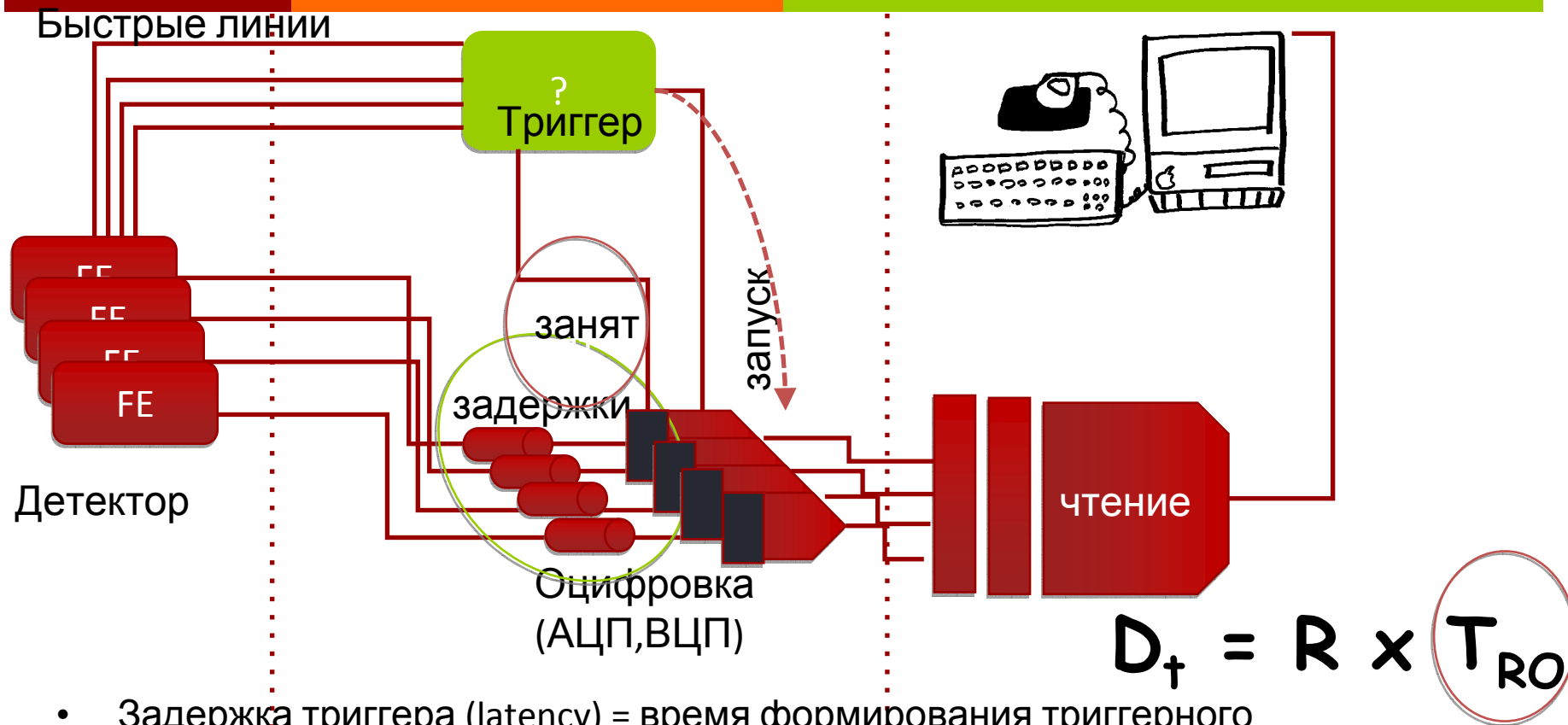


- Использование множества оцифровщиков: триггер посылает команду на быстрое чтение или быстрый сброс
- Разные стадии чтения: быстрое местное чтение и медленное общее чтение

➤ Мертвое время - произведение частоты триггера на мертвое время чтения

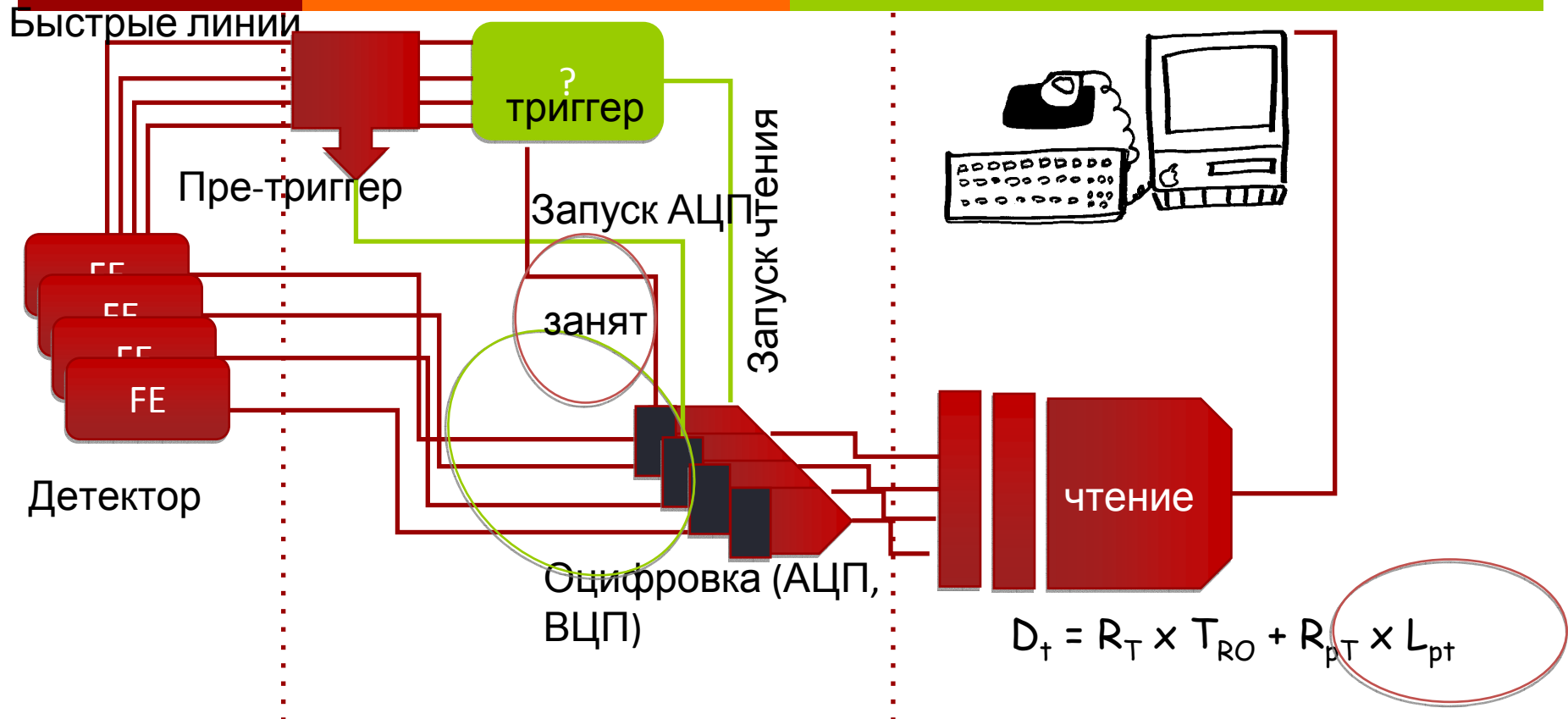
$$D_t = R \times T_{RO}$$

Задержка триггера



- Задержка триггера (latency) = время формирования триггерного решения и доведения его до потребителей
- Сигналы необходимо задержать до того как придет триггерное решение
 - Чем более сложный отбор событий тем больше задержка

Предварительный триггер



- Можно добавить очень быстрый триггер (пре-триггер) сигнализирующий о наличии минимальной активности в детекторах
 - Для запуска (медленной) оцифровки которая потом считывается после основного триггера
 - Пре-триггер должен быть доступен в момент прихода данных к оцифровщикам
- Основной триггер может придти позже (после оцифровки) -> может быть более сложным

Частота триггера и чтения

- Можно развить идею... **много уровней триггера**, каждый из которых уменьшает частоту, работает большее время
- Мертвое время будет равно сумме мертвого времени триггера (по всем уровням) и мертвого времени чтения

i=1 пре-триггер

$$\left(\sum_{i=2}^N R_{i-1} \times L_i \right) + R_N \times T_{LRO}$$

R_i = частота после уровня i

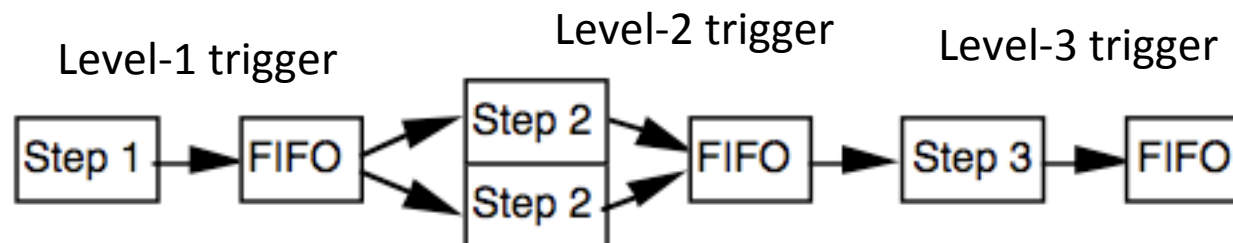
L_i = задержка триггера уровня i

T_{LRO} = время локального чтения

Мертвое время чтения

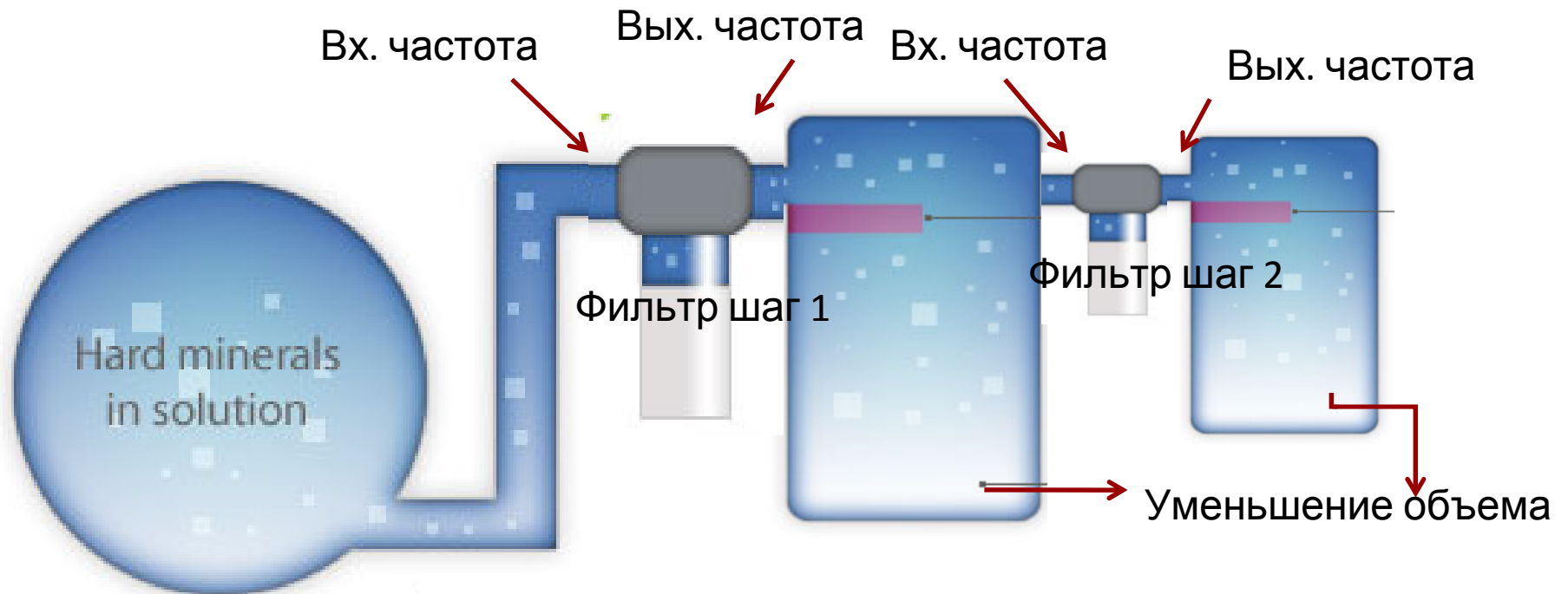
Многоуровневый триггер

- Используется в крупных экспериментах, где более сложные решения принимаются на более медленных частотах
 - Первый уровень с короткой задержкой и малой выбраковкой, работающий на больших частотах
 - Высокие уровни дальше отбирают события, уменьшая частоту, с более длинными временами решения (более сложные алгоритмы)
- Эффективность триггера для исследуемых физических процессов должна быть высокой, так как отброшенные события потеряны навсегда



Буферизация и фильтрация

- С каждым шагом уменьшается объем данных, и все более сложные фильтры могут использоваться на следующем шаге
- На время фильтрации данные буферизуются
 - Входная частота определяет время обработки и емкость буфера
 - Выходная частота ограничивает максимальную задержку триггера
 - Мощность фильтра ограничена емкостью буфера на следующем шаге



• До тех пор пока буфера не заполнились, нет дополнительного мертвого времени

- Если частота после фильтрации превышает возможности следующего шага:
- Добавить фильтр (более жесткий отбор)
 - Произвольный отсев (масштабирование)
 - Улучшить фильтр (более сложный отбор)

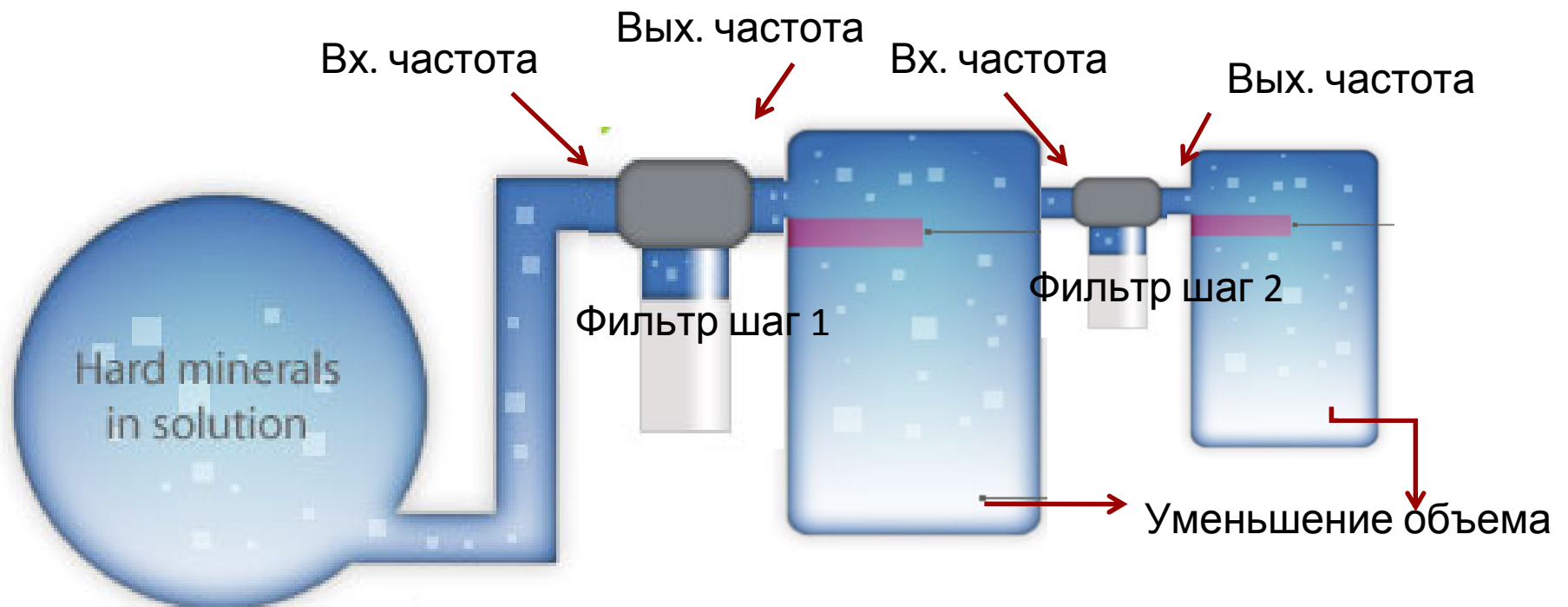
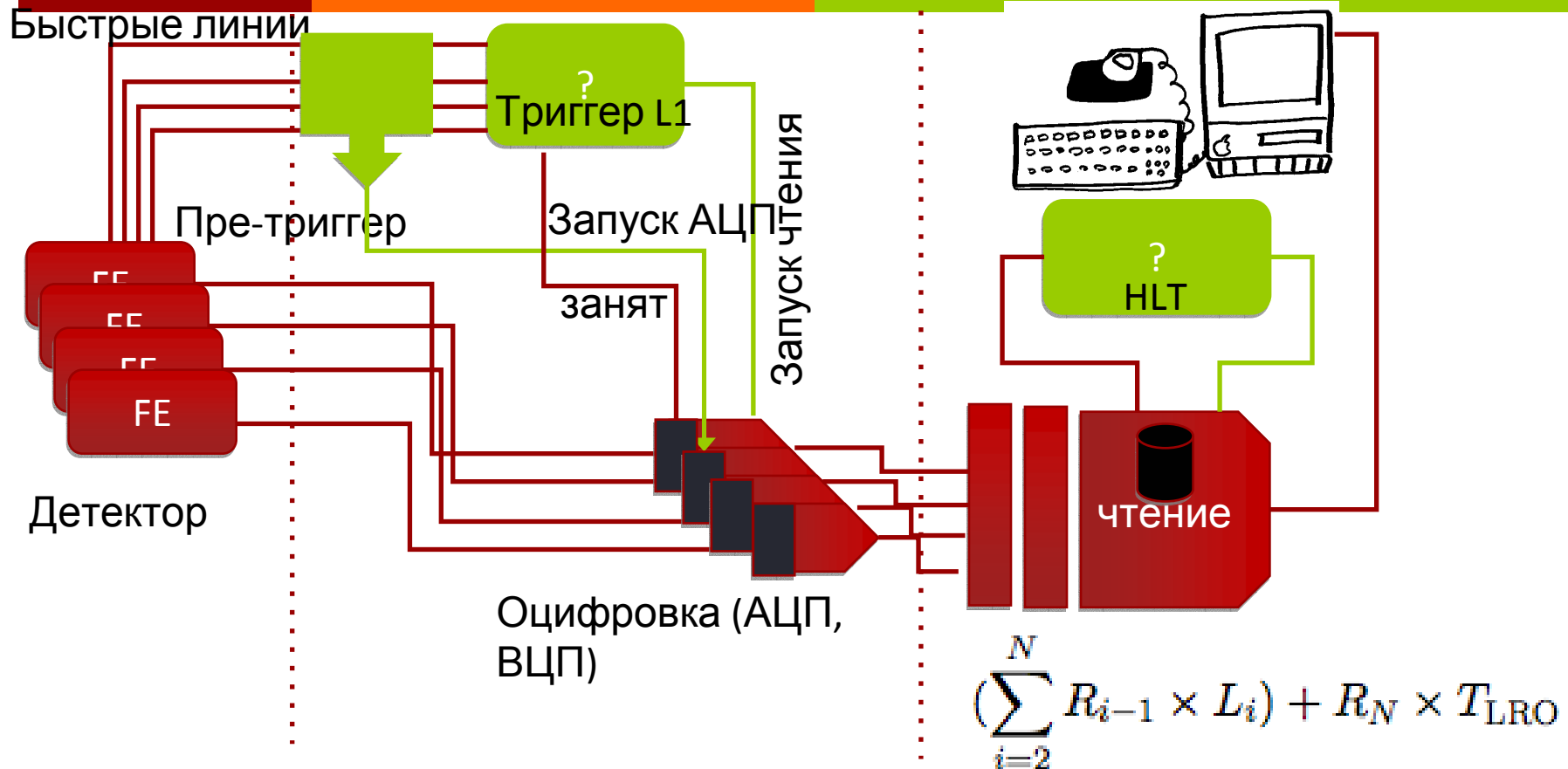


Схема многоуровневого триггера

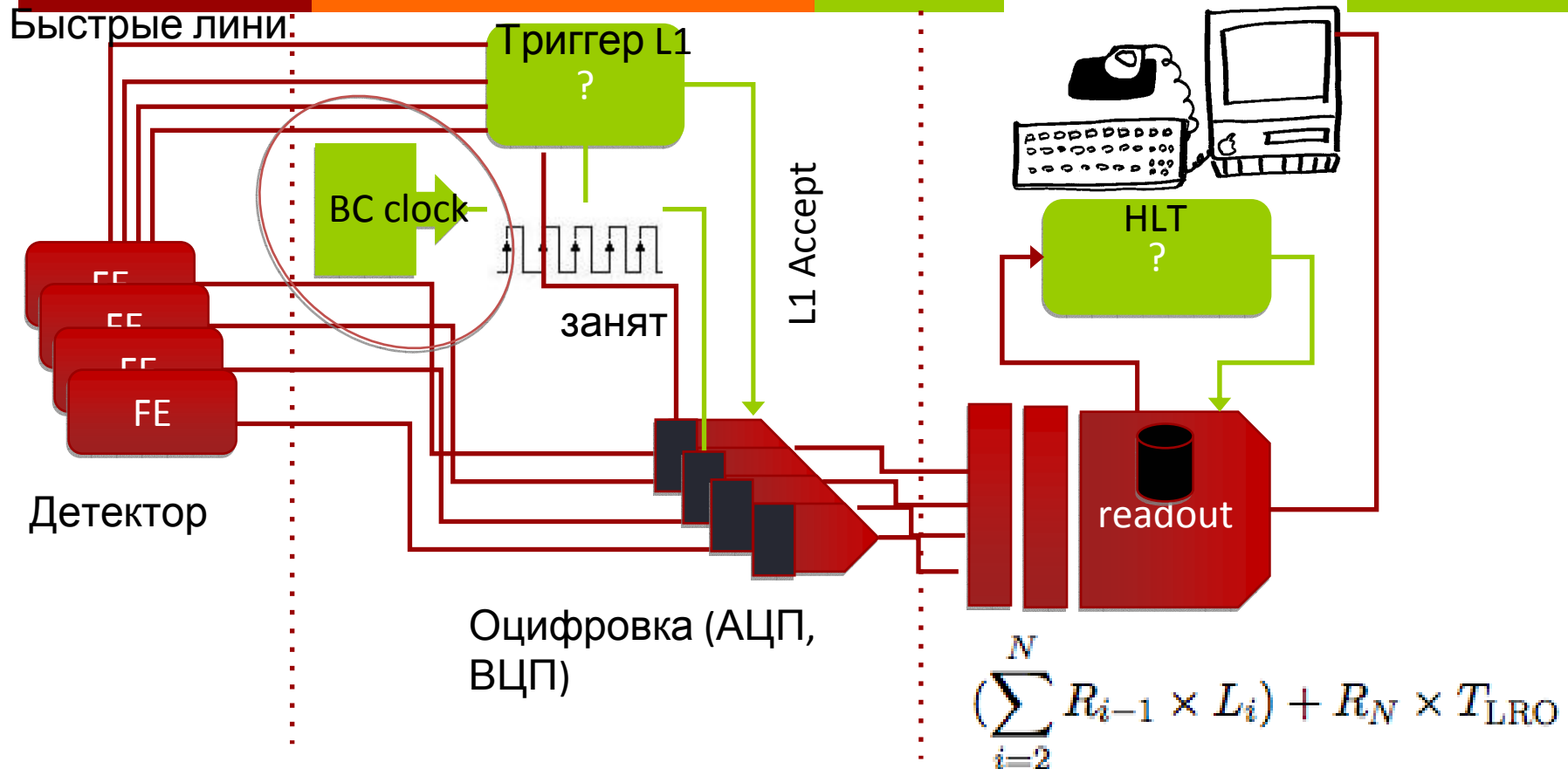


- Разные уровни триггера имеют доступ к разным данным
- Пре-триггер запускает оцифровку

Пре-триггер и синхронизация

- В коллайдерных экспериментах, частота банчей пучка (BC) может быть использована в качестве пре-триггера
 - Триггер первого уровня может использовать время между двумя BCs для принятия решения, без мертвого времени, если время между ними достаточно велико
 - Решение принимается между 2 BCs, когда не происходит взаимодействий
- L1 триггер является синхронной системой
 - Каждый BC событие оцифровывается и обрабатывается
 - Каждый BC принимается решение, называемое L1-Ассерпт, хотя задержка триггера больше
 - LHC: BC = 40 МГц, BC каждые 25 нс. Задержка триггера L1 может быть несколько микросекунд.

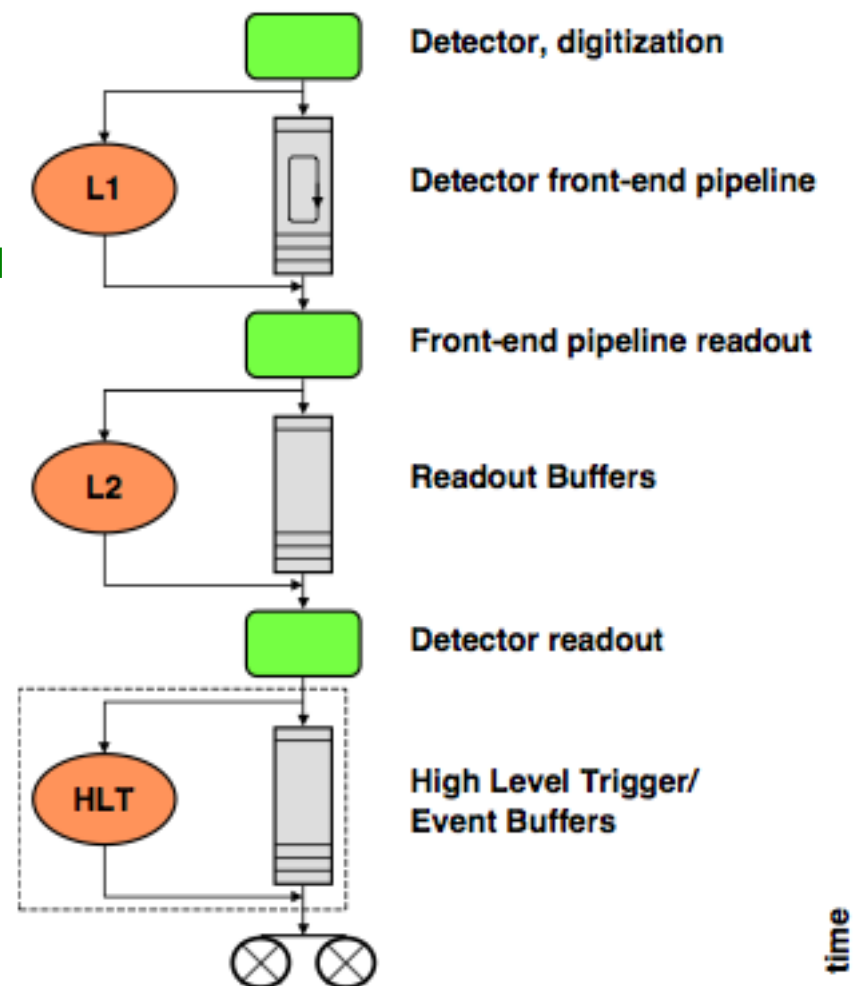
Схема многоуровневого триггера на коллайдерах



- L1: общая тактовая частота разводится по триггеру и детекторам
- Каждый уровень триггера запускает последующий уровень

Разные уровни – разные требования

- Чем более выборочен триггер
 - Меньше частота
 - Большой размер события
 - Лучше гранулярность
 - Больше сложность
 - Длиннее задержка
 - Больше буфера
- L1
 - Короткая задержка без мертвого времени
- Высокие уровни (HLT)
 - Выборочность, контроль частоты



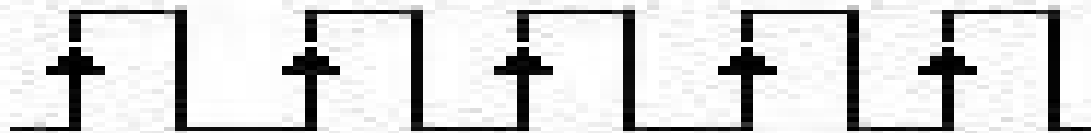
L1: уменьшение задержки

- Конвейер
- Быстрая обработка
- Быстрое перемещение данных

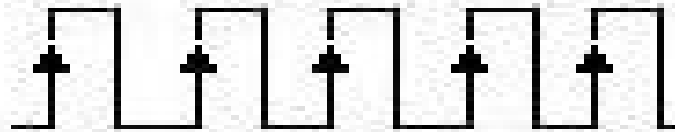
L1: время решения

- На LEP, интервал BC = 22 μ s давал возможность работать сложному триггеру с задержкой в несколько мкс.
- В современных коллайдерах, требования к светимости задают малое время между пересечениями пучка (25нс)
 - **За это время невозможно принять сложное решение!**

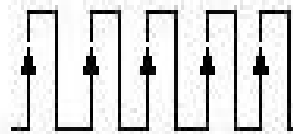
Tevatron: 396 ns



HERA: 96 ns



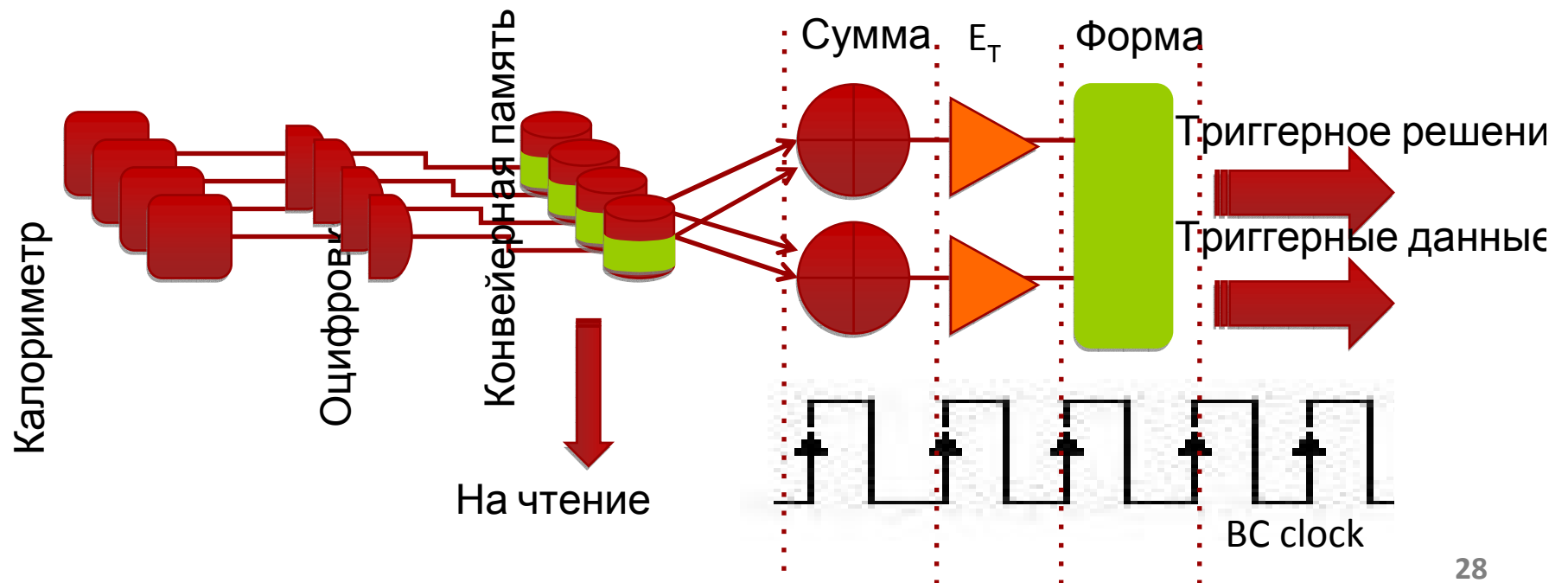
LHC: 25 ns



$$R = \mu \cdot f_{BC} = \sigma_{in} \cdot L$$

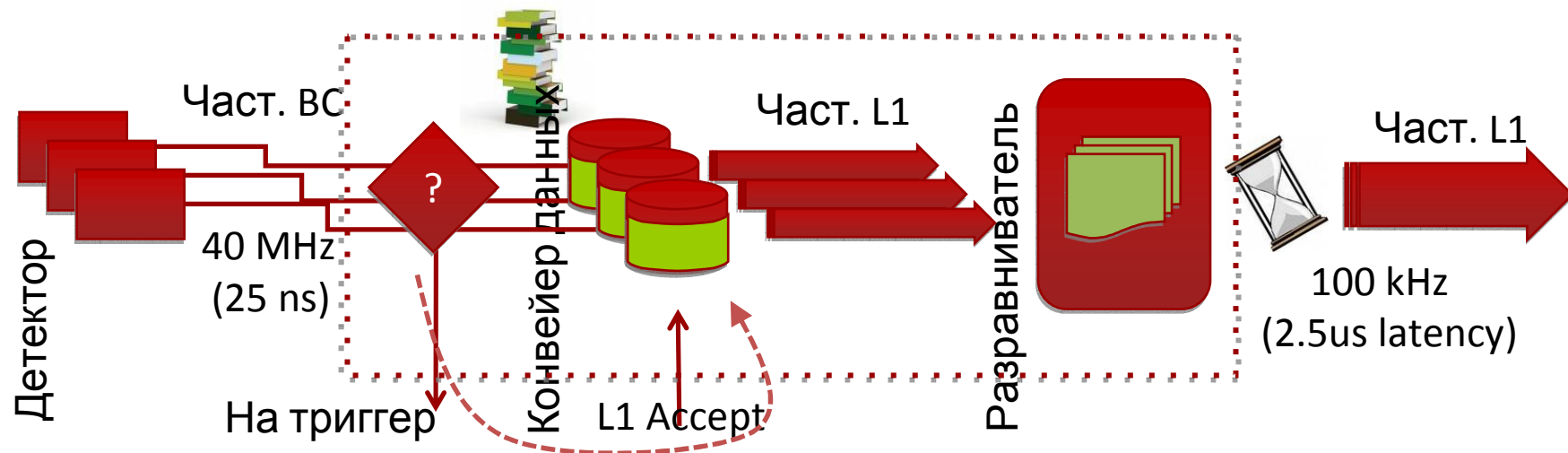
L1: конвейер

- Каждое пересечение пучка, необходимо выработать решение, но поскольку данные буферизуются, решение можно принять позже
- Тем самым необходимо чтобы триггер одновременно обрабатывал несколько событий
 - Параллельные операции
 - Разбиение обработки на шаги, каждый шаг в течении одного ВС
- Задержка фиксирована, исходя из суммы времен обработки и передачи данных



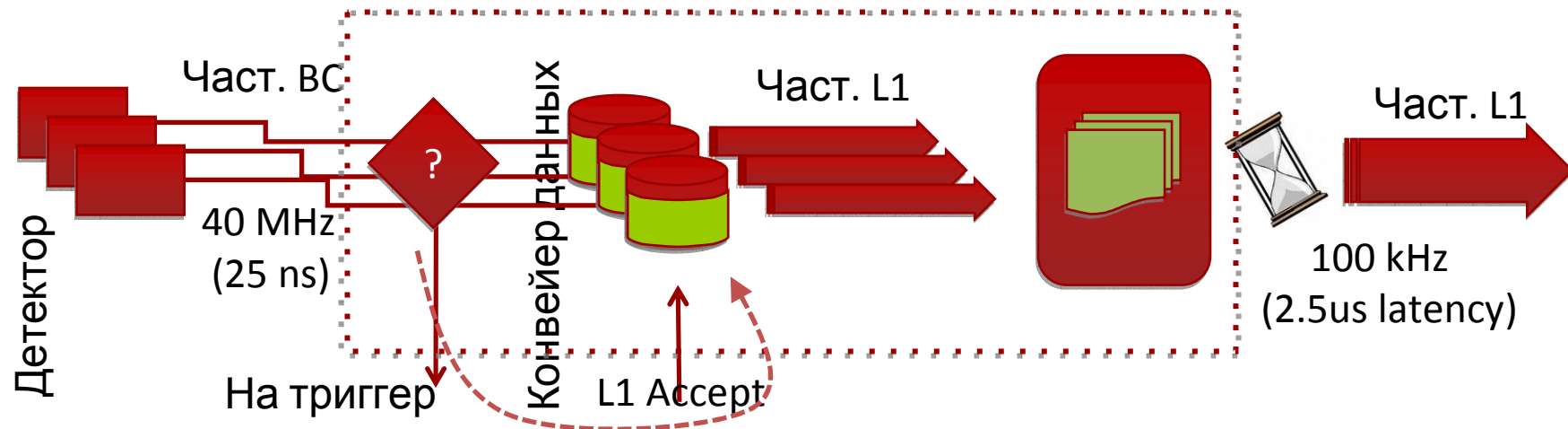
L1 триггер: чтение

- Конвейерное чтение L1:
 - Данные хранятся в логическом конвейере на время решения
 - Положительное решение (L1A) запускает чтение локальных буферов
- Локальные буфера должны быть необходимой глубины для ожидаемой задержки триггера, иначе данные соответствующие принятому триггеру уже будут потеряны (вытеснены новыми данными)
 - BC 25 ns, l1 latency 2.5 us -> буфер на 100 событий минимум (100 BCs)
- Фиксированная задержка позволяет найти данные соответствующие каждому триггеру



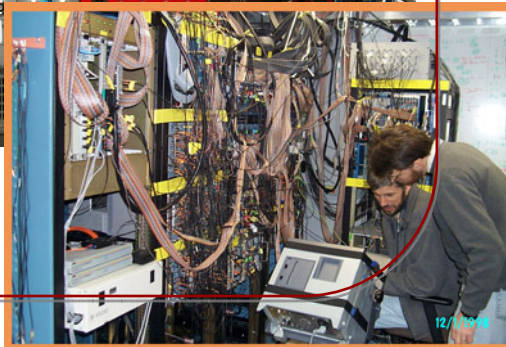
L1 триггер: чтение

- Из FIFO, по приходу L1-Ассерта данные передаются в разравнивающий буфер
- Добавляется небольшое мертвое время на входе (несколько BCs) чтобы избежать наложения данных
- Мертвое время добавляется на выходе чтобы предотвратить переполнение разравнивателя (если время между двумя триггерами слишком мало)
 - LHC: 5 BC мерт. время \times 100 kHz L1 \times 25 ns = 1.25 %

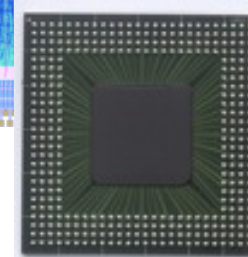
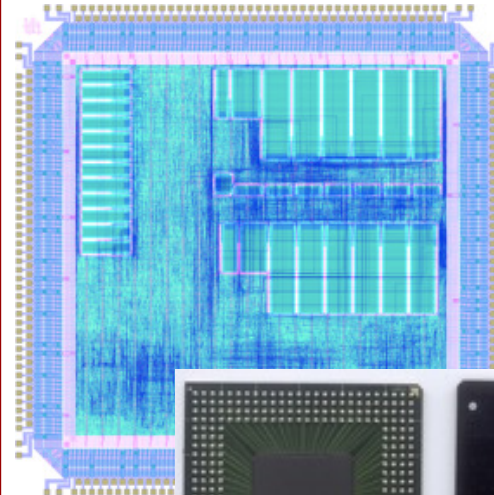


Выбор системы

- Модульная электроника
 - Простые алгоритмы
 - Низкая цена
 - Простота использования



- Цифровые интегрированные системы
 - Сложные алгоритмы
 - Быстрая обработка сигналов
 - Необходимы специальные знания и умения



Интегральные схемы и микропроцессоры

- Требования для больших скоростей
 - Сложные алгоритмы
 - Программируемые решения на языках высокого уровня
 - Форматирование и сжатие данных
 - Мониторирование и обнаружение сбоев

CPU Transistor Counts 1971-2008 & Moore's Law

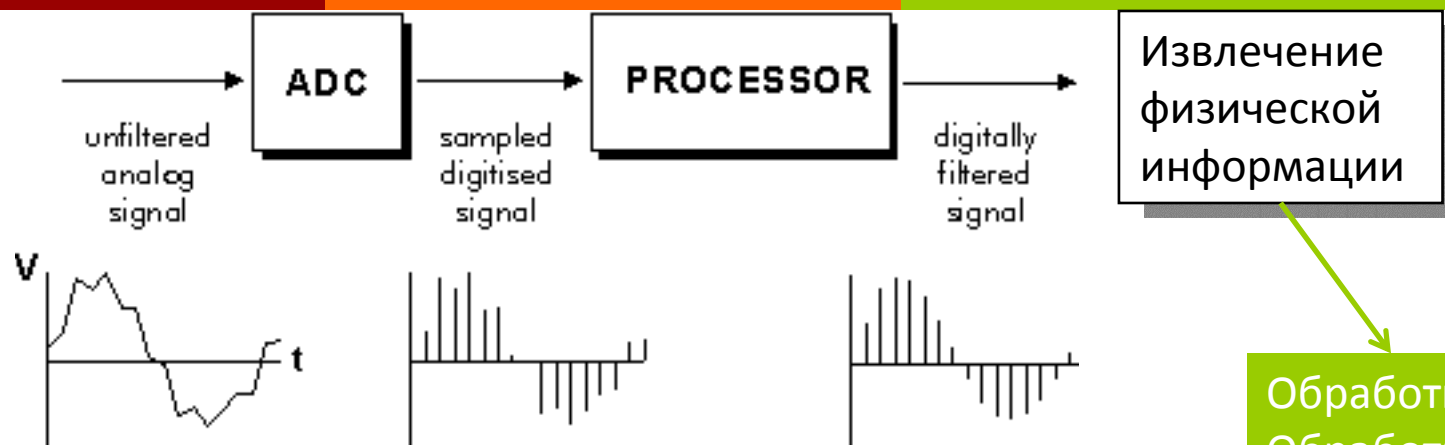


Закон Мура определяет развитие электроники

- Плотность упаковки
- Скорость

- Интегральные схемы дают преимущество в надежности, меньшей мощности, меньшего числа компонентов и лучшей производительности
- Микропроцессоры
 - Одна микросхема со всеми основными функциями компьютера: CPU, память, порты в/в, прерывания
 - Может быть встроена в ССД: чтение, буферизация и обработка данных непосредственно рядом с регистрирующей электроникой

Цифровая обработка сигналов (DSP)



- DSP – микропроцессоры производящие миллионы операций с плавающей запятой в секунду (над параллельными данными)
- Оцифровка:
 - Измерение чего-либо (заряд, амплитуда, время,...) что сравнивается с образцом (ADC, FADC,...)
- Обработка сигналов
 - Цифровые фильтры (выделение сигнала из шума)
 - Сжатие данных
 - Высокая производительность
 - Обработка сигнала со скоростью его получения

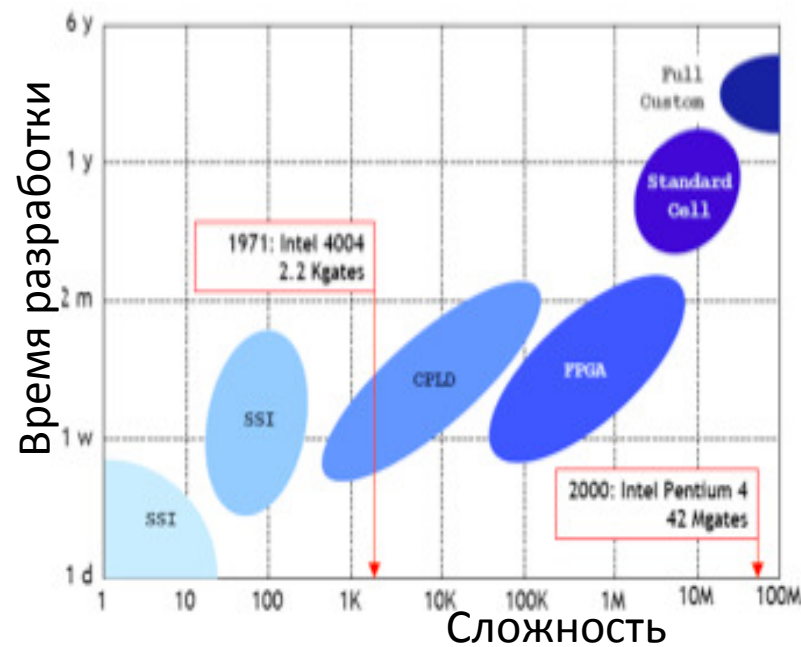
Обнаружение и
коррекция ошибок
Малый уровень
шума

DSP



Быстрые триггерные процессоры

- Специализированные интегральные схемы (ASIC): оптимизированные для быстрой обработки
 - Процессоры Intel, ~ GHz
- ASIC (почти как FPGA)
 - Широко доступны 100 MHz (1/10 скорости спец. ASIC)



В/В

Память

Цифровая
логика

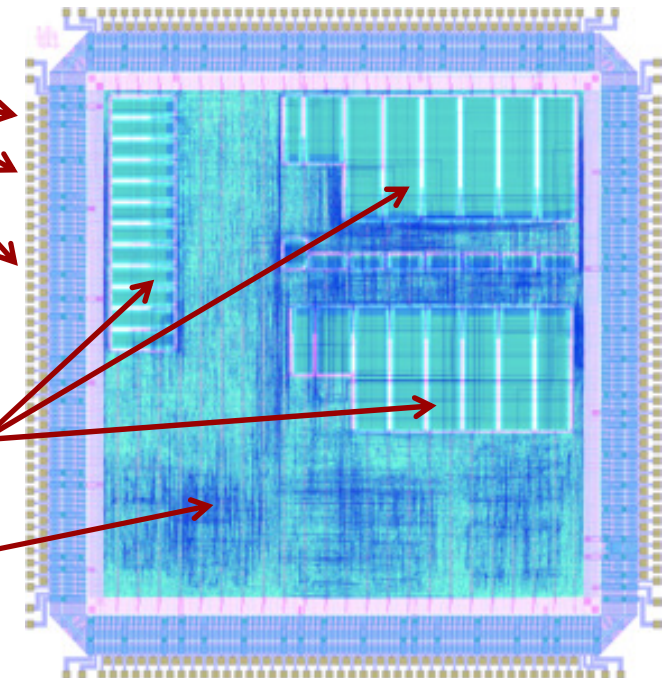
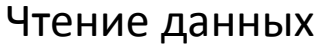
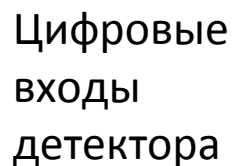


Схема CM ASIC

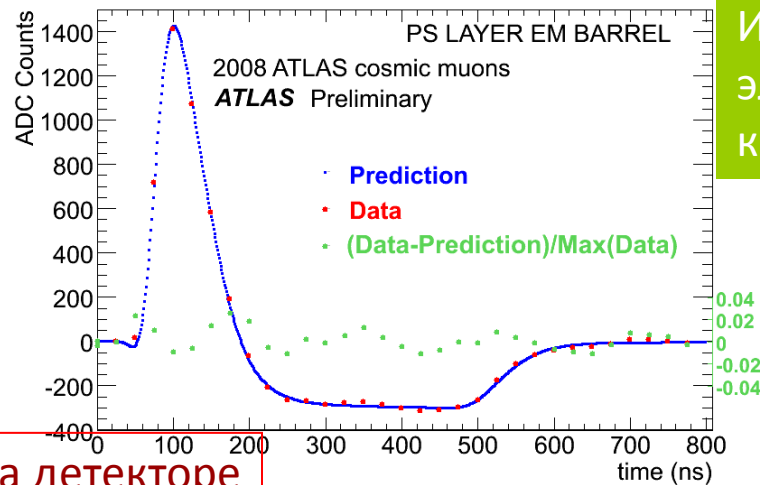


Выход для чтения

Входные
сигналы
синхронизации



Пример : калориметрический триггер АТЛАС



На детекторе

➤ Суммирование аналоговых сигналов

Пре-процессор

➤ Оцифровка сигнала: разрешение 10бит

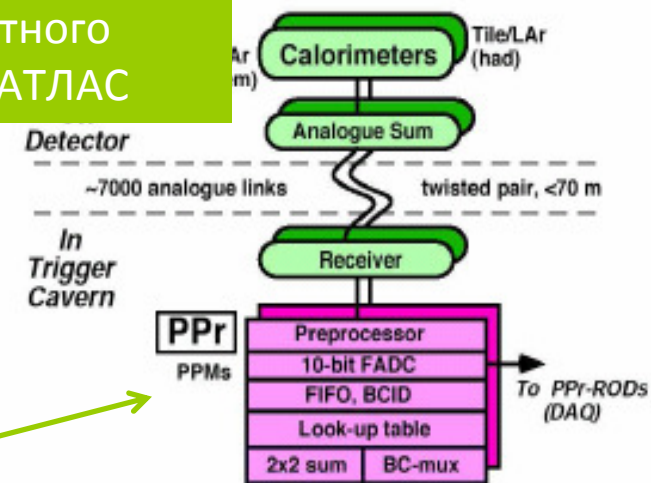
➤ Триггерный алгоритм

- Каждому бину придается значение энергии по таблицам соответствия
- Порог на значение поперечной энергии

➤ Сигнал длиной 8 BC

- Поиск пика сигнала для определения BC взаимодействия

Импульс
электромагнитного
калориметра АТЛАС



Пример: калориметрический триггер АТЛАС

- Кластерный процессор (CP)
- Процессор энергии/струй (JEP)
- Выполнен на FPGA, легко изменяемые параметры алгоритма
- 5000 цифровых линий соединяют Ppr, JEP и CP, 400 Mb/s

Кластеризация для АТЛАС е/γ триггерного алгоритма

