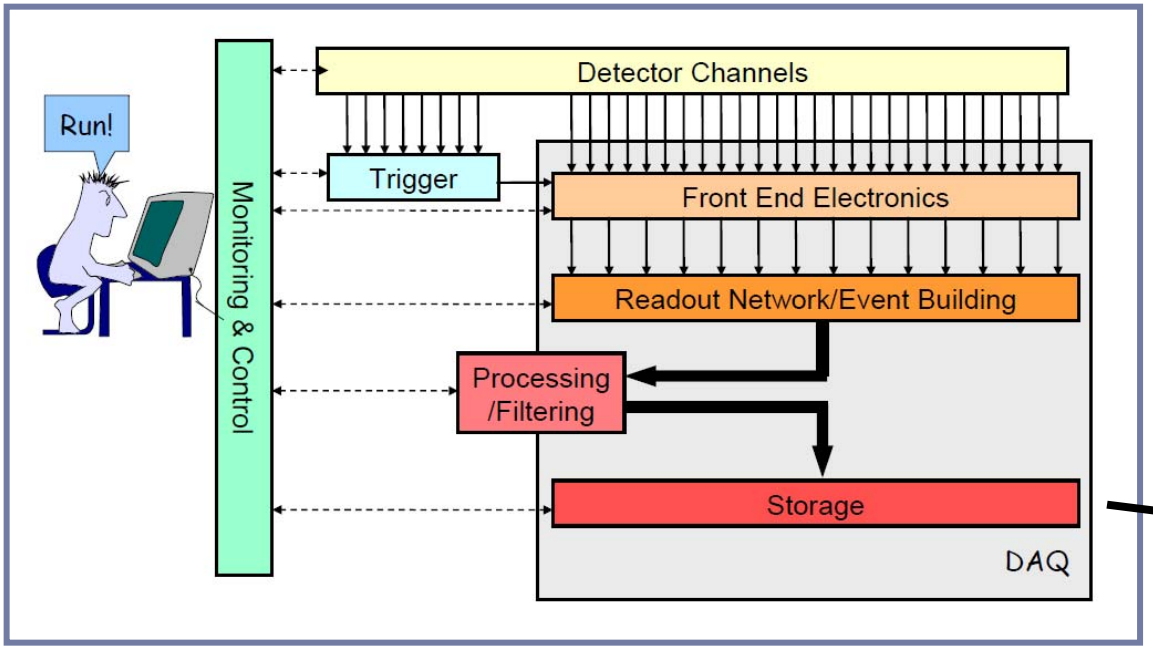
Компьютерные технологии в физике элементарных частиц.

Системы сбора данных

Задача системы сбора данных



Система сбора данных (DAQ, T/DAQ, DAS)

Моделирование

Обработка данных

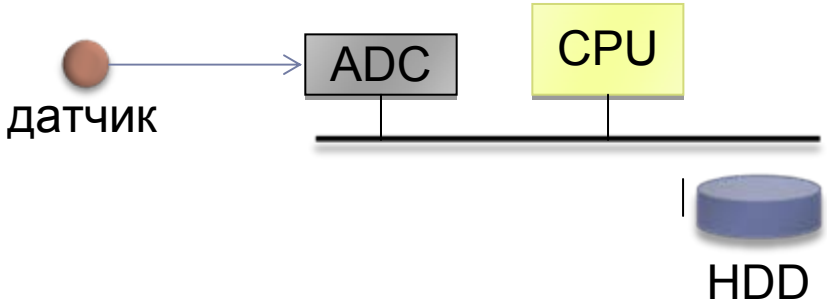
РЕЗУЛЬТАТ

Основная задача: обработка сигналов, возникших в системах детектора, и сохранение полезной информации в системе хранения

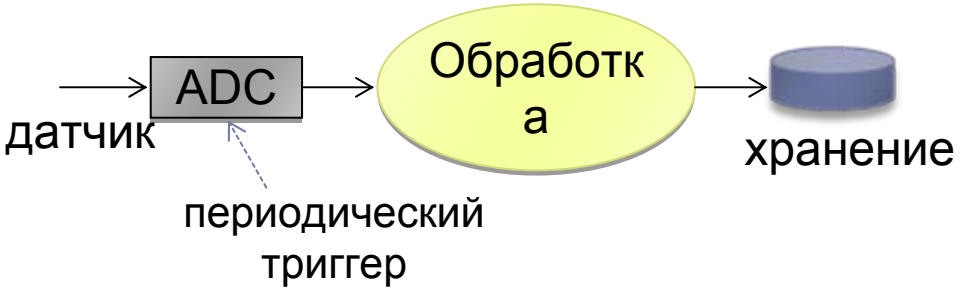
Простейшая система сбора данных

Внешний вид

Физическая модель



Логическая модель



- Простейшая система сбора данных – периодическое измерение температуры

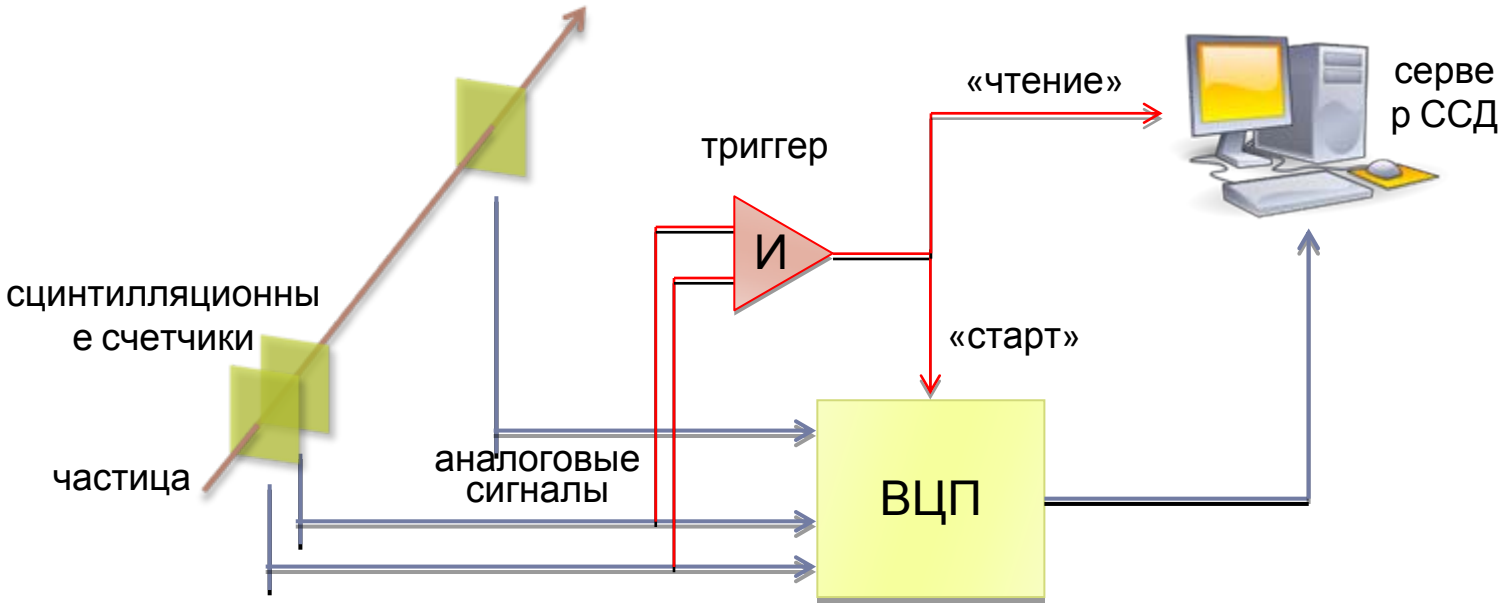
- Максимальная частота измерения определяется: временем измерения + временем преобразования + временем обработки + временем записи (передачи)
Например, если полное время измерения 1 мс, то система сможет обрабатывать 1 кГц измерений (событий).

- Системы с периодическим измерением характерны для систем мониторинга

Регистрация случайных событий

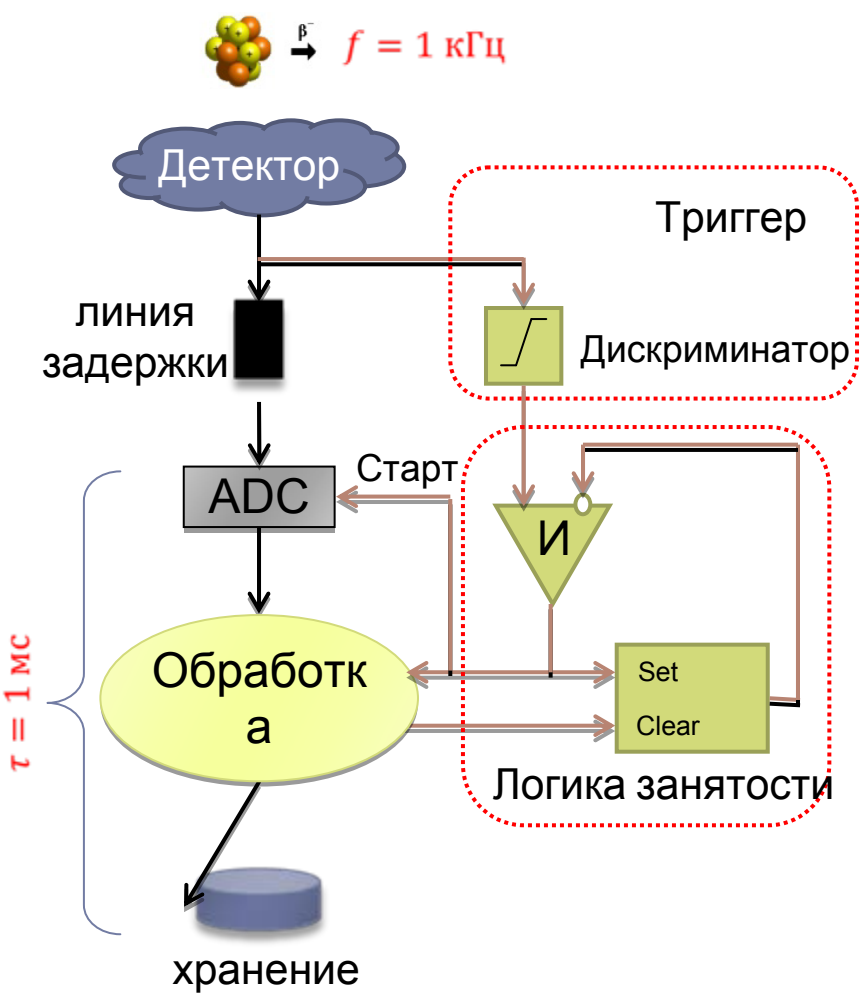
В экспериментах часто возникает задача регистрации событий, происходящих в случайный момент времени.

Пример: измеряем время пролета частицы через ряд счетчиков. Как определить, когда нужно начать измерять время?



Триггер – специальное устройство, которое формирует сигнал для системы сбора данных о том, что в детекторе произошло интересное событие.

Система сбора данных с триггером



Особенности реализации ССД с регистрацией случайных событий:

- Измеряемые сигналы необходимо задерживать на время работы триггера
- Необходимо реализовать логику занятости триггера, чтобы новое событие не возникло, пока система обрабатывает текущее событие
- События, возникшие пока система занята, отбрасываются – «мертвое время»

Пусть средняя частота появления событий 1 кГц и время обработки событий 1 мс. С какой средней частотой мы будем регистрировать события?

Случайные события

Пусть события в системе возникают случайно со средней частотой

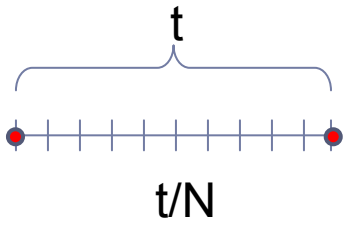
$$f = 1/T.$$

Пусть время обработки события (мертвое время) равно τ.
Тогда эффективность работы системы (доля зарегистрированных событий):

$$f_{out} = f \cdot (1 - f_{out}\tau)$$

$$\varepsilon = \frac{f_{out}}{f} = \frac{1}{1 + f \cdot \tau}$$

Свойства случайных событий



Вероятность того, что следующее событие появится через время t:

$$p(t) \sim \left(1 - \frac{t}{NT}\right)^N = e^{-t/T}$$

С учетом нормировки:

$$p(t) = \frac{1}{T} e^{-t/T}$$

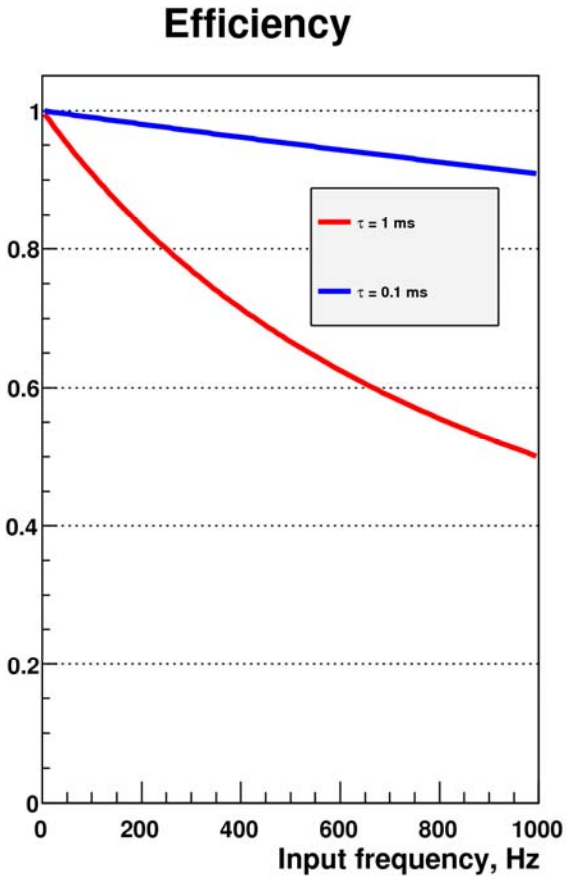
Среднее время между событиями:

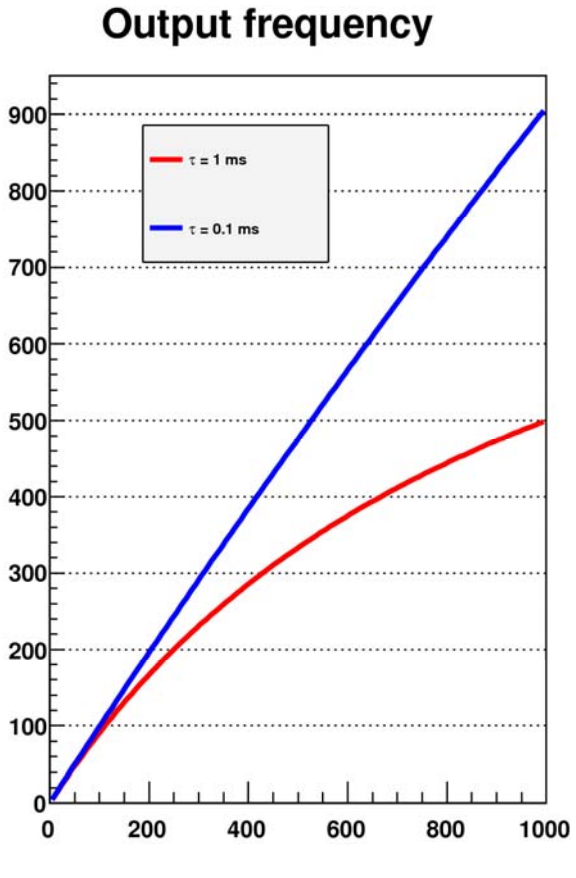
$$\langle t \rangle = \int_0^\infty t \cdot \frac{1}{T} e^{-\frac{t}{T}} = T$$

Вероятность прихода n событий за время Δt (распределение Пуассона):

$$p_n(\Delta t) = \frac{\mu^n e^{-\mu}}{n!}, \text{где } \mu = \Delta t/T$$

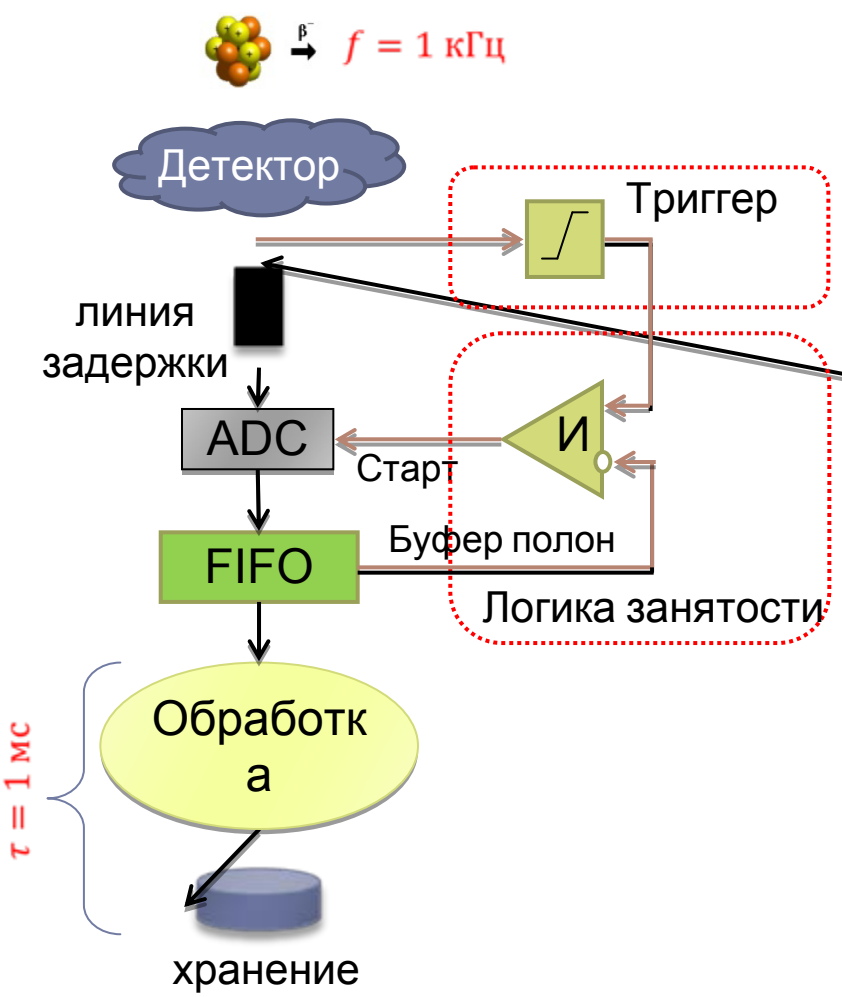
Эффективность работы простой системы сбора данных с триггером





Для достижения приемлемой эффективности (>90%) нужна система сбора данных с пропускной способностью, в 10 раз превышающей частоту появления событий!

Буферизация



Решение проблемы случайности событий – буферизация:

- Добавление FIFO буфера (First In First Out) позволяет отделить «быструю» оцифровку сигналов от «медленной» обработки.
- Буферизация позволяет увеличить эффективность системы. Мертвое время в приведенном примере определяется быстродействием «быстрой» части (АЦП) и средней пропускной способностью «медленной» обработки.
- Буферизация превращает случайный поток событий на входе в равномерный поток событий на выходе (de-randomizer)

Почему работает буферизация?

Пусть события в системе возникают случайно со средней частотой $1/T$, а обрабатываются со средней частотой $1/\tau$. Пусть в очереди $(N-1)$ слотов. Пусть P_n – вероятность того, что в очереди заполнено n-слотов. Как вероятность P_n меняется со временем?

$$P_n(t+\Delta t) = P_n(t)\cdot\left(1-\frac{\Delta t}{T}\right)\left(1-\frac{\Delta t}{\tau}\right) + P_{n-1}(t)\cdot\frac{\Delta t}{T}\left(1-\frac{\Delta t}{\tau}\right) + P_{n+1}(t)\cdot\left(1-\frac{\Delta t}{T}\right)\frac{\Delta t}{\tau}$$

В стационарном режиме вероятности не меняются, т.е. $P_n(t+\Delta t) = P_n(t)$.

Возникает система уравнений:

$$\begin{cases} \frac{1}{T}P_{n-1} - \left(\frac{1}{T}+\frac{1}{\tau}\right)P_n + \frac{1}{\tau}P_{n+1} = 0 \\ \frac{1}{T}P_0 - \frac{1}{\tau}P_1 = 0 \\ \frac{1}{T}P_{N-1} - \frac{1}{\tau}P_N = 0 \end{cases}$$

Решение:

$$P_n = \left(\frac{\tau}{T}\right)^n P_0 = \rho^n P_0$$

$$1 = \sum P_n = P_0(1+\rho+\cdots+\rho^N)$$

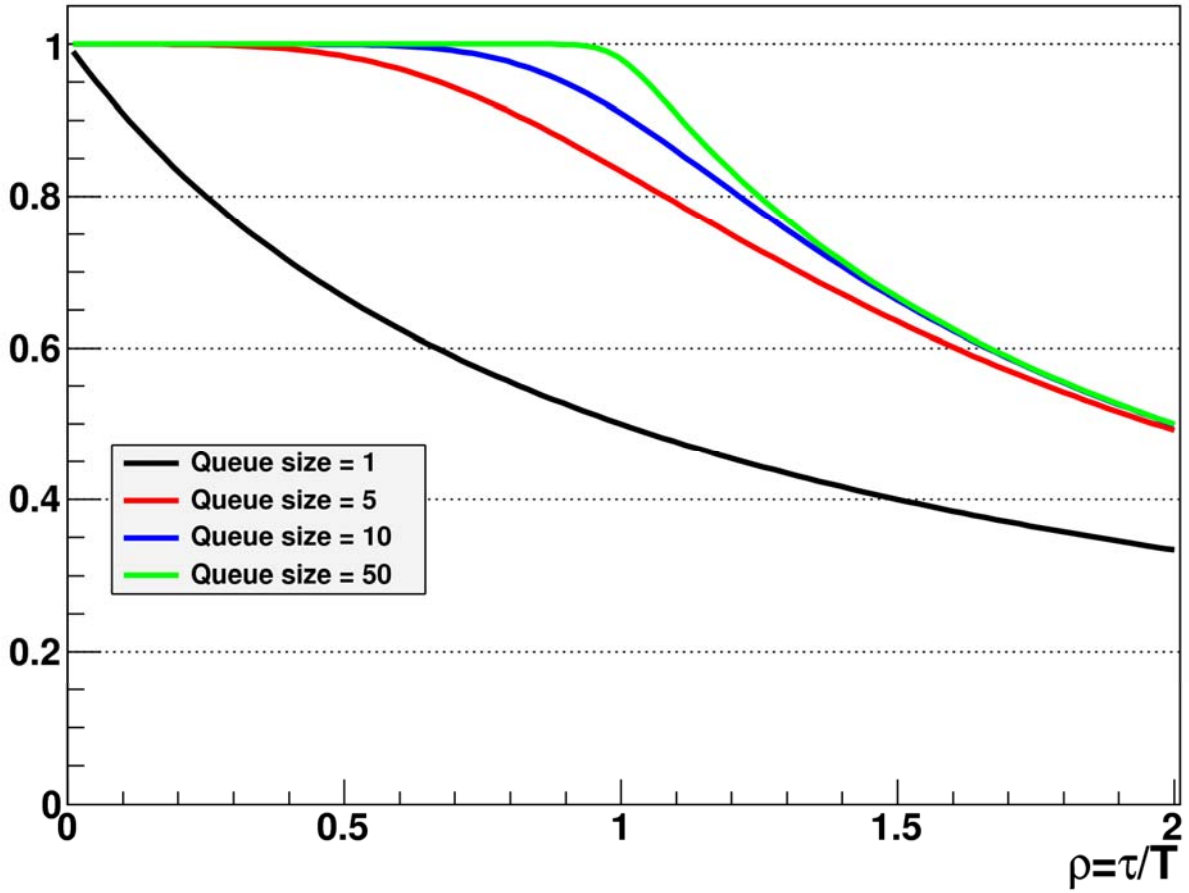
Доля времени, в течении которого заполнены n слотов:

$$P_n = \frac{(1-\rho)\rho^n}{1-\rho^{N+1}}$$

Эффективность работы системы:

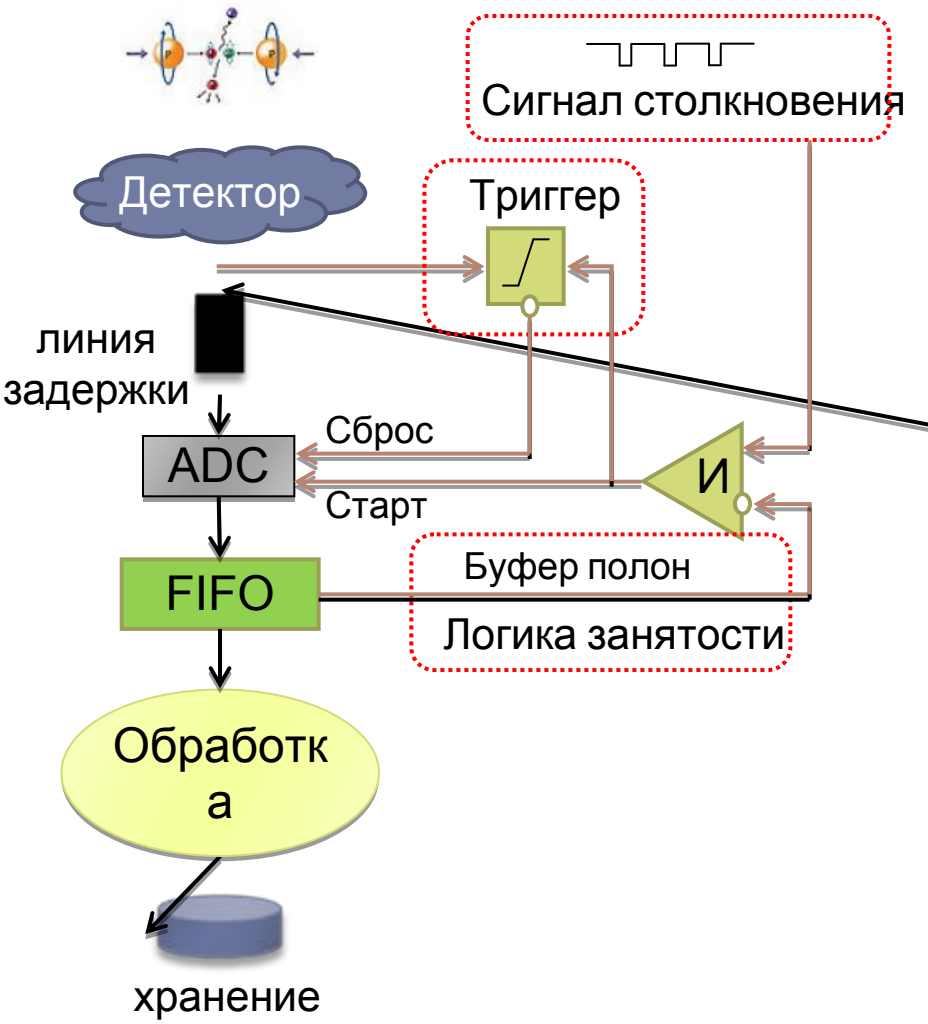
$$\varepsilon = 1 - P_N = \frac{1-\rho^N}{1-\rho^{N+1}}$$

Эффективность работы буферизованной системы



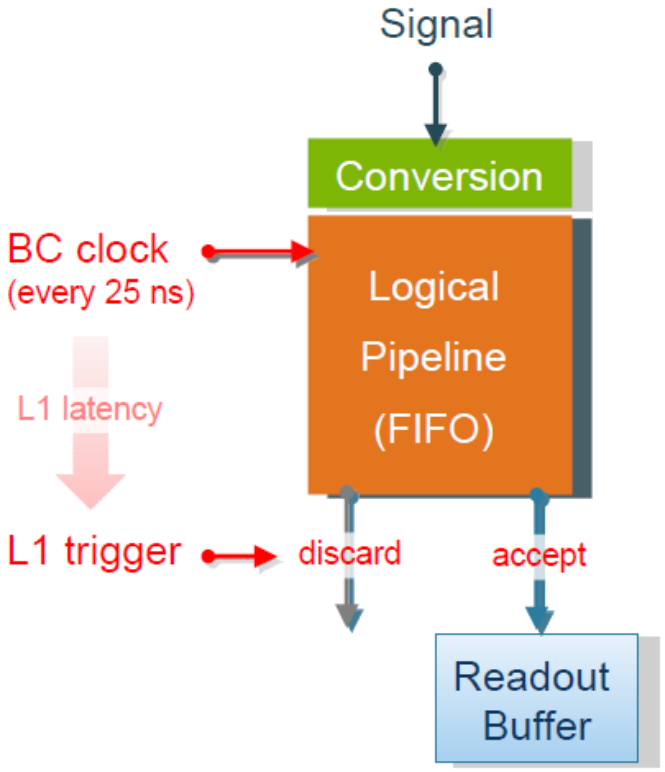
Буферизация позволяет достичь приемлемой эффективности без повышения пропускной способности системы сбора данных сверх необходимой

Использование сигнала о столкновении пучков



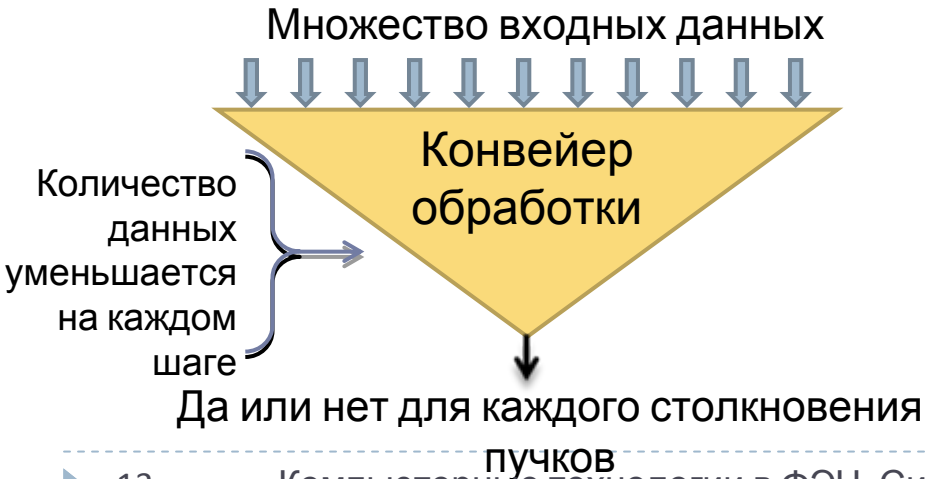
- В экспериментах на ускорителях столкновения пучков происходят в известные моменты времени
- Сигнал столкновения пучков может быть использован для стробирования сигнала триггера, либо использован в качестве пре-триггера, тогда триггер отбрасывает неинтересные события
- Хотя столкновения пучков происходят в определенные моменты времени, интересные события при столкновении все равно происходят случайно. Поэтому буферизация все равно необходима.

Конвейерное чтение



- Часто в системах сбора данных применяется конвейер.
- Данные в конвейере могут сохраняться в аналоговой форме (конденсаторы) или в цифровой форме.
- Конвейер может применяться как линия задержки, или как часть триггера, тогда по мере продвижения по конвейеру данные обрабатываются.
- Длина конвейера определяется временем работы первичного триггера. В качестве временного шага конвейера часто используется время между столкновениями пучков.
- Если сигнал первичного триггера не появился, по достижении конца конвейера данные уничтожаются.

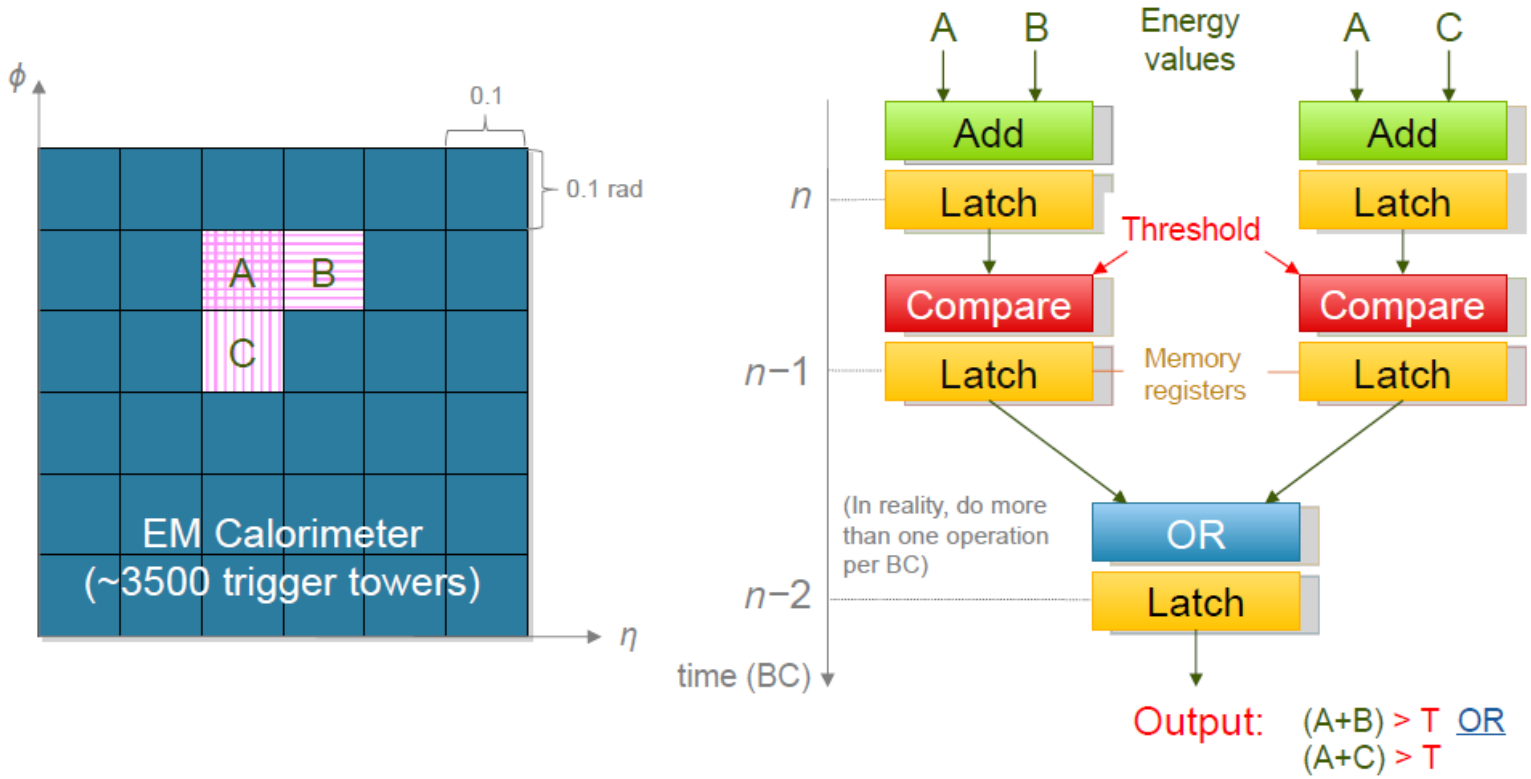
Многоуровневый триггер



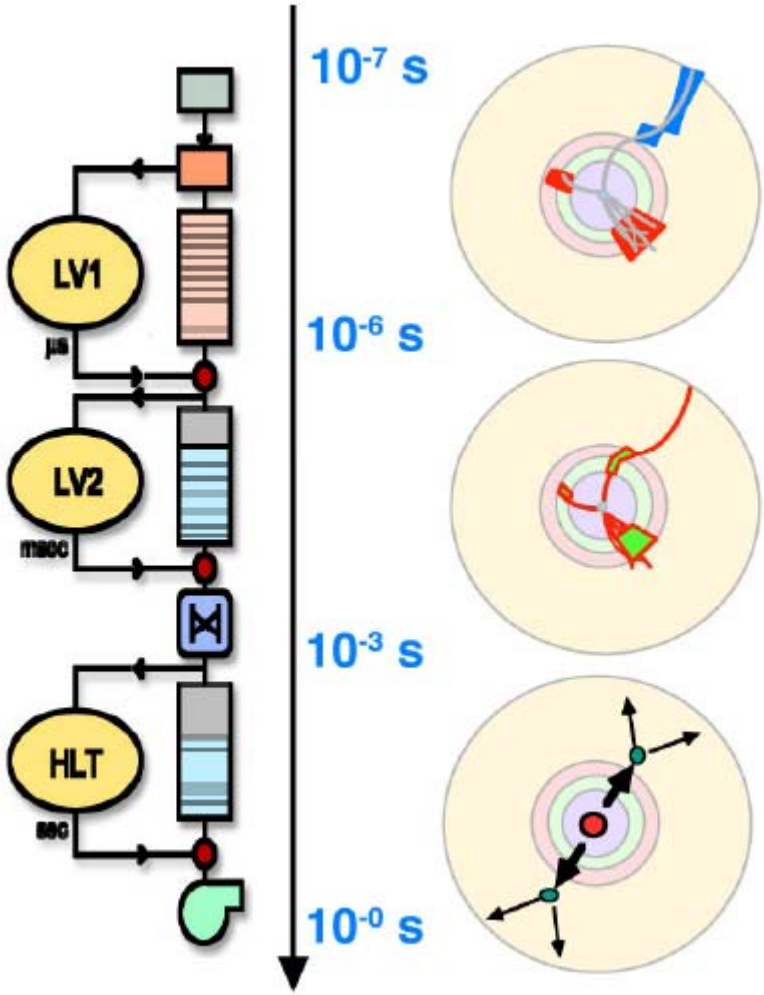
- Для достижения оптимального соотношения производительности, цены и сложности, реализуют многоуровневый триггер.
- Ключевой параметр: произведение числа событий на входе на время работы триггера на каждом уровне
- Чем выше уровень триггера, тем с большим массивом данных он оперирует
- Триггер нижнего уровня часто реализован в электронике с использованием конвейера, триггер высокого уровня, как правило, чисто программный.

Пример: конвейер первичного триггер калориметра детектора ATLAS

Задача: определить, превысило ли суммарное энерговыделение в соседних ячейках определенный порог.



Пример: многоуровневый триггер в экспериментах на LHC



Частота столкновений 40 МГц, 10^9 Гц событий

Первичный триггер, 100 кГц

Локальная идентификация признаков (High p_t, e, струи,…)

- Локальная реконструкция
- Суммарное энерговыделение в области детектора

Вторичный триггер, 1 кГц

Идентификация частиц (Z, W,…)

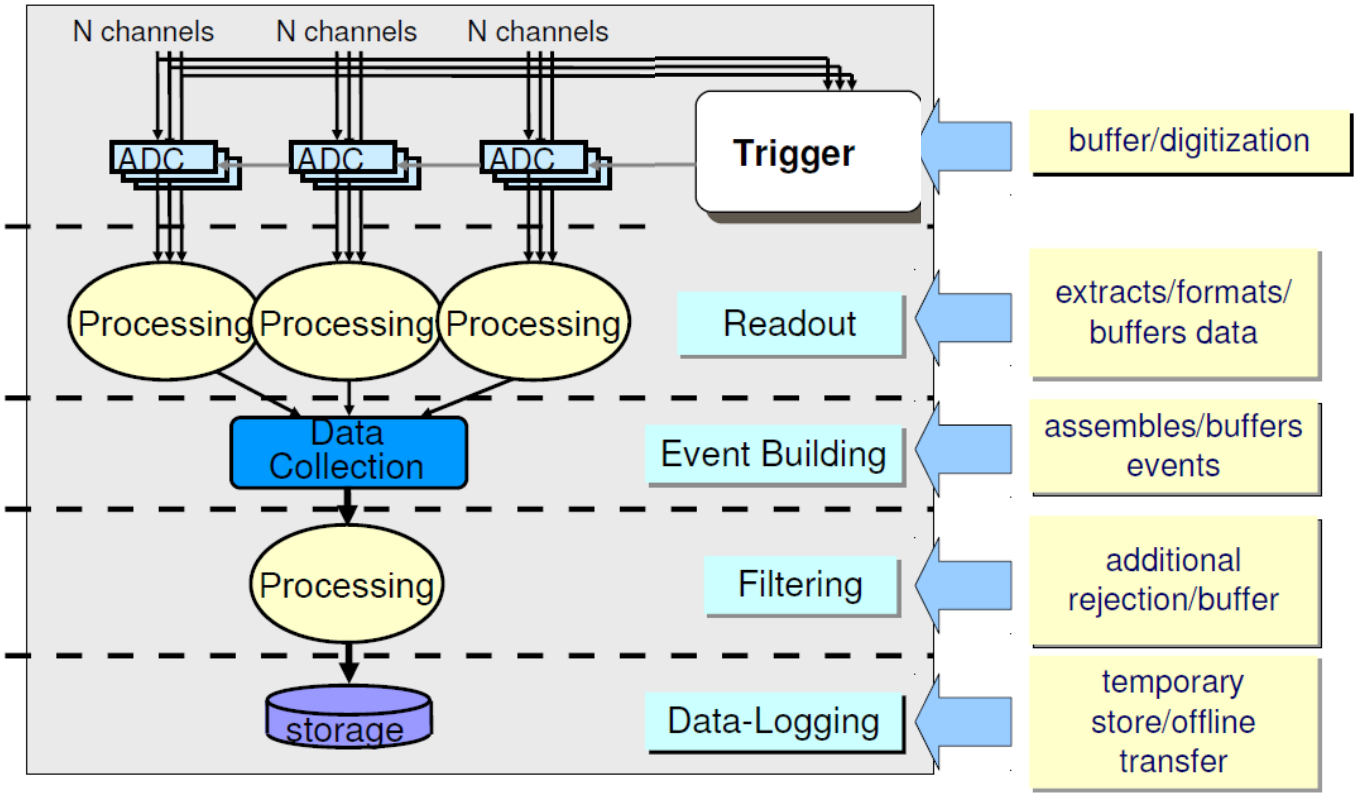
- Более точная реконструкция
- Реконструкция трека в нескольких системах
- Кинематика события

Третичный триггер, 10-100 Гц

Идентификация события

- Полная реконструкция всего события и последующий анализ

Система сбора данных большого эксперимента

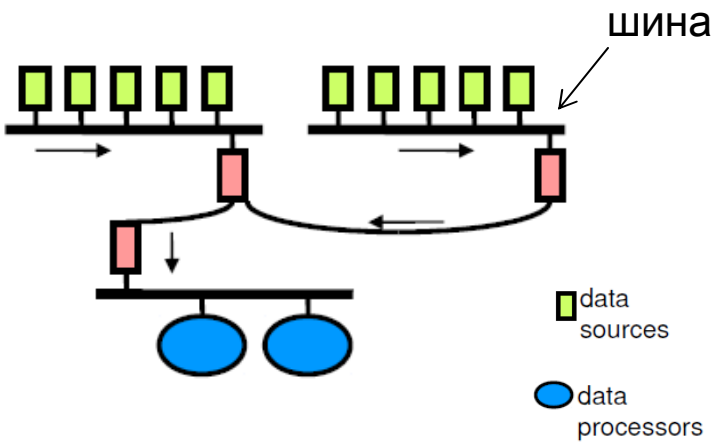


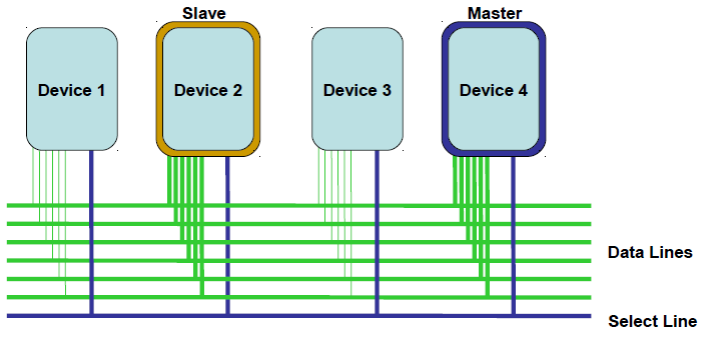
• В большой системе возникает иерархическая структура и добавляется стадия «сборки события»

• Множество оцифровщиков нужно вычитывать параллельно. Две основных топологии: шина и сеть.

Шина

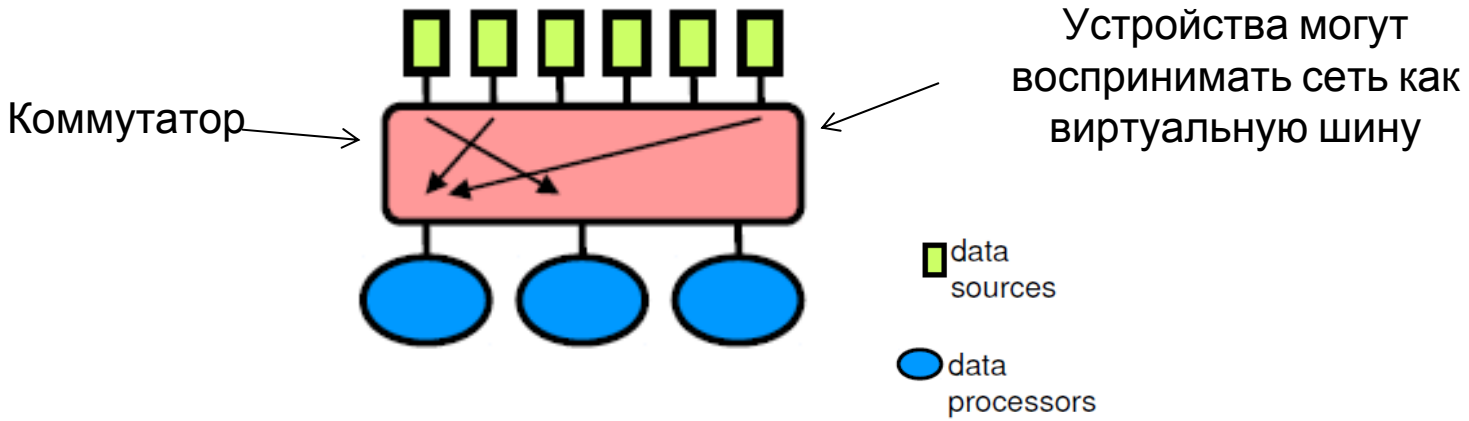
- Шина – это линии связи, разделяемые устройствами на шине.
- Поскольку шина общая, на ней должен присутствовать арбитр. Все устройства на шине должны разделять определенный стандарт (электрический стандарт, протокол связи и т.п.)
- Примеры: CAMAC, VME, PCI, SCSI, Parallel ATA,…
- Чтение данных через общую шину чаще всего встречается в самых первых уровнях системы сбора данных
- Недостатки: ограничение на количество устройств, на общую пропускную способность





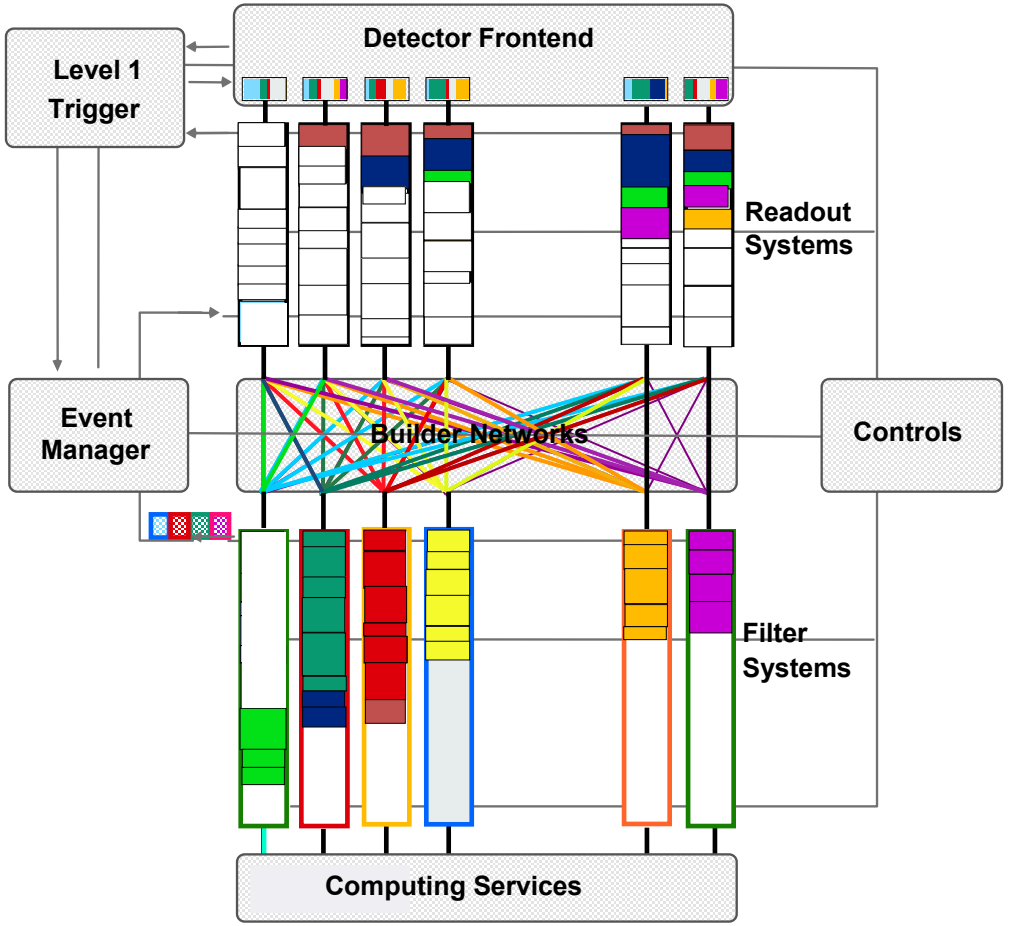
Сеть

- В сети устройства «напрямую» общаются друг с другом, пересылая сообщения.
- Примеры: Ethernet, телефонная сеть,…
- Все устройства в сети равноправны.
- Специальные устройства – коммутаторы – организуют связь отдельных устройств между собой и разрешают коллизии.
- Сеть – гораздо более расширяемое решение.
- Чтение данных через сеть обычно используется на верхних уровнях системы сбора данных



Сборка события

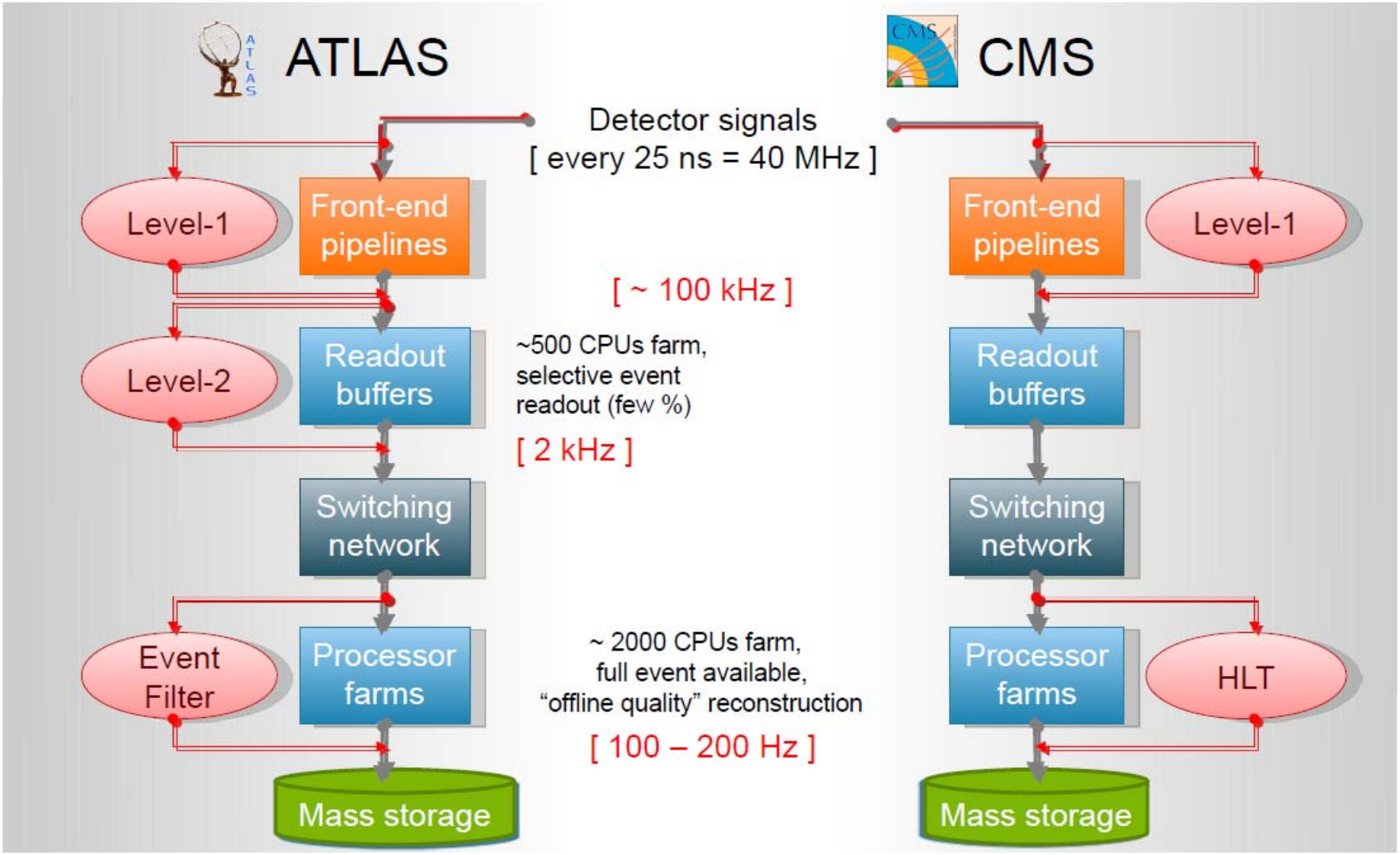
- В больших системах чтение данных в отдельных подсистемах производится независимо (иначе возникают проблемы с синхронизацией и производительностью)
- Каждый фрагмент данных помечается номером триггера или столкновения
- Фрагменты, принадлежащие одному событию, объединяются вместе – «сборка события»
- В реализации сборки событий широко используются сетевые технологии и кольцевой буфер.



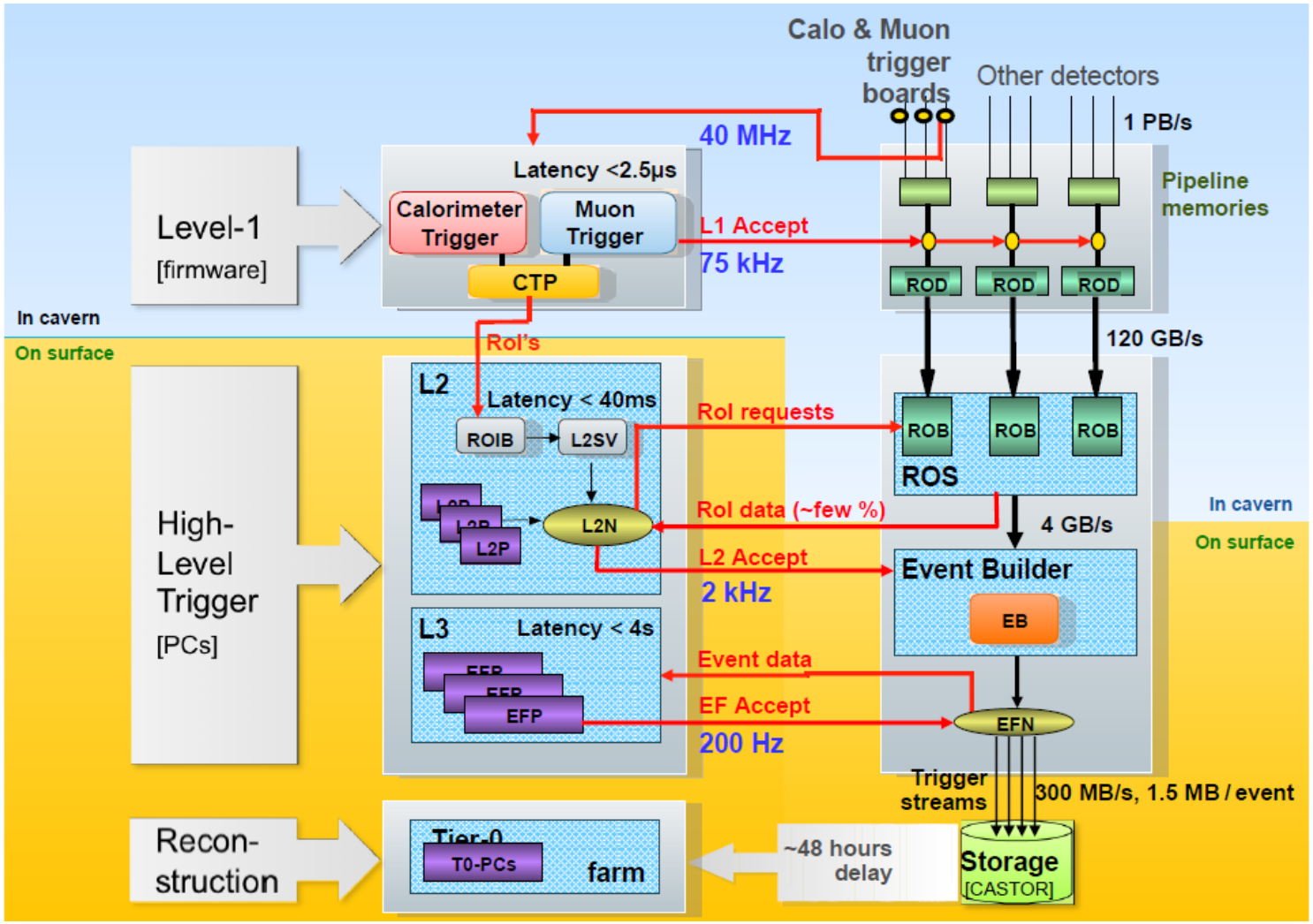
Параметры систем сбора данных в экспериментах на LHC

	No. Levels Trigger	Level-0,1,2 Rate (Hz)	Event Size (Byte)	Readout Bandw.(GB/s)	HLT Out MB/s (Event/s)
ATLAS	3	LV-1 10^5 LV-2 $3x10^3$	$1.5x10^6$	4.5	300 ($2x10^2$)
CMS	2	LV-1 10^5	10^6	100	O(1000) (10^2)
LHCb	2	LV-0 10^6	$3x10^4$	30	40 ($2x10^2$)
ALICE	4	Pb-Pb 500 p-p 10^3	$5x10^7$ $2x10^6$	25	1250 (10^2) 200 (10^2)

Сравнение систем сбора данных детекторов ATLAS и CMS



Триггер и потоки данных в детекторе ATLAS



Дополнительные функции ССД

Помимо основной функции (чтение и сохранение данных), система сбора данных выполняет ряд дополнительных функций:

1)управление системами детектора (например, подготовка электроники к работе)

2)управление процессом чтения и записи данных (Run Control)

3)мониторинг систем детектора и периферийных систем (слежение за температурами, напряжениями,…)

4)мониторинг качества оцифрованных данных

5)реализация различных компьютерных сервисов:

- ✓ хранилище данных
- ✓ база данных
- ✓ журнал работы системы
- ✓ …

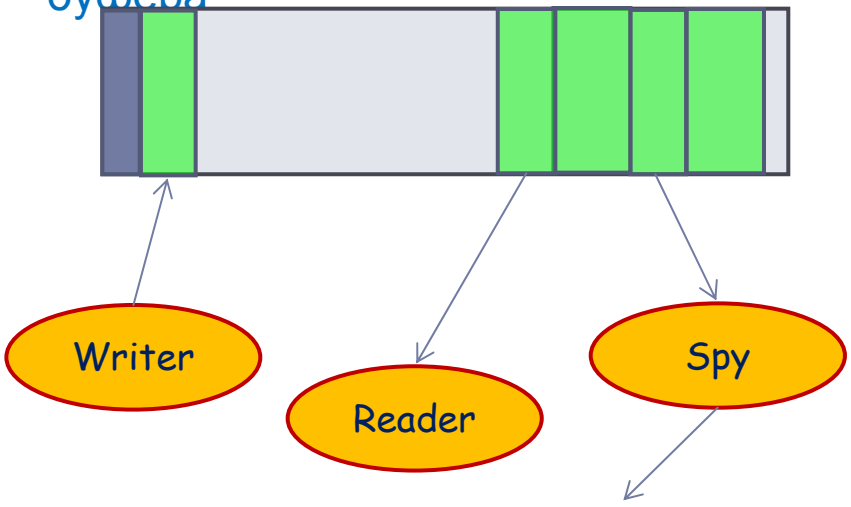
Часть системы сбора данных, которая реализует эти функции, часто выделяется в отдельную систему управления и контроля (slow control system)

Мониторирование потока данных

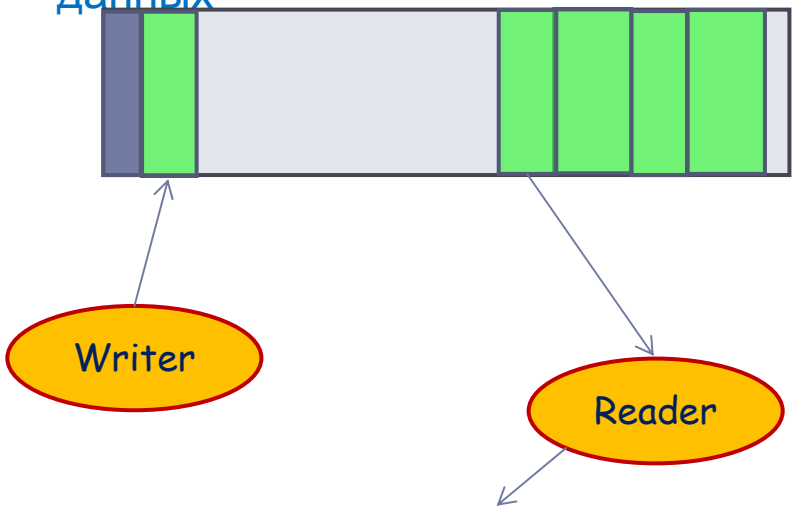
В программном обеспечении системы сбора данных часто выделяется сервис, который позволяет “подсмотреть” в поток данных и скопировать оттуда отдельные события или фрагменты. Эти данные анализируются и используются для мониторирования потока данных. “Подсматривание” не должно мешать основной функции системы сбора данных.

Возможные схемы реализации:

«Подсматривание» в содержимое буфера



Встраивание в процесс чтения данных



Обмен информацией между процессами ССД

В программном обеспечении ССД часто выделяется сервис, который позволяет организовать единое информационное пространство для процессов системы. Это может быть реестр, база данных,...

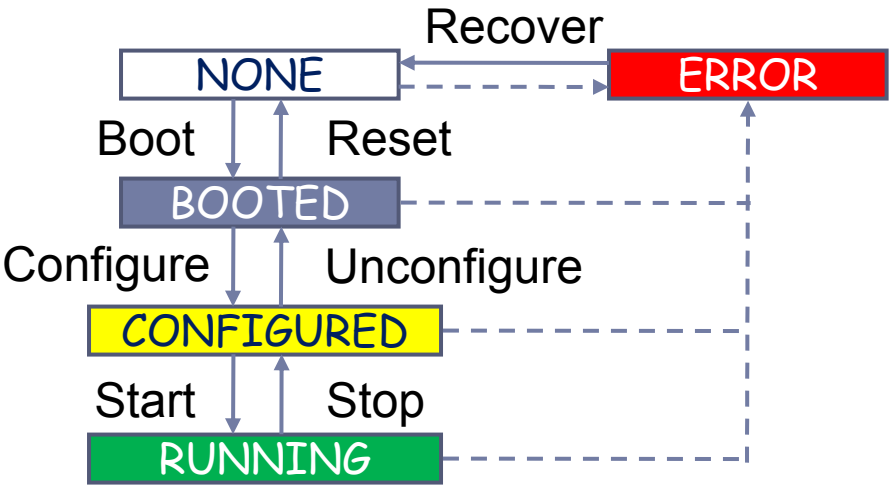
Управление состоянием ССД

Управление процессом работы системы сбора данных часто реализуется с помощью модели конечного автомата.
Конечный автомат характеризуется 4 свойствами:

- Набором состояний, определяющими поведение процесса
- Переходами, при которых процесс переходит из одного состояния в другое
- Правилами и условиями, при которых возможет переход
- Внешними или внутренними событиями, которые генерируют запрос о смене состояния

Каждый процесс ССД является конечным автоматом. Выделенный центральный процесс предоставляет пользовательский интерфейс и позволяет оператору изменять состояние системы (начать запись данных, закончить запись данных и т.п.)

Модель состояний типичного процесса ССД



Изменение состояния системы

Состояние процесса изменяется только тогда, когда все процессы, подчиненные по отношению к нему, завершили свой переход из одного состояния в другое.

Запросы на изменение состояния часто рассылаются в виде специальных команд или сообщений.

С программной точки зрения, переходы из одного состояния в другое обычно реализуются в виде функций обратного вызова (callback).