



Введение в системы сбора данных

О.Соловьянов



Предисловие



- Сбор данных не является **точной наукой**. Это «алхимия» электроники, компьютеров, сетей, физики и ... опыта (включая «хакерство»)
 - ..., а так же денег, времени и собственно людей, ...
- Речь пойдет в основном о системах сбора данных экспериментов физики высоких энергий
- Цель данной лекции состоит в изложении основ, по возможности избегая технических деталей, которые будут объяснены позднее

- Многие материалы и идеи данных лекций позаимствованы из ISOTDAQ и CERN Summer Student Lectures



Содержание



→ *Введение*

→ *Что такое ССД(DAQ)?*

- *Общая схема*

→ *Основные принципы ССД*

- *Оцифровка (digitization), задержка (latency)*
- *Мертвое время (dead time), разравнивание (de-randomization)*

→ *Масштабирование*

- Чтение данных (readout) и построение событий (event building)
- Обратное давление
- Шины и сети (buses & networks)

Википедия:

Система сбора данных (ССД, DAQ) – комплекс средств, предназначенный для автоматизированного сбора информации о значениях физических параметров и перевода их в цифровую форму, накопление и передачу данных

Система сбора данных не является однородным объектом

Не очень четко очерчены границы

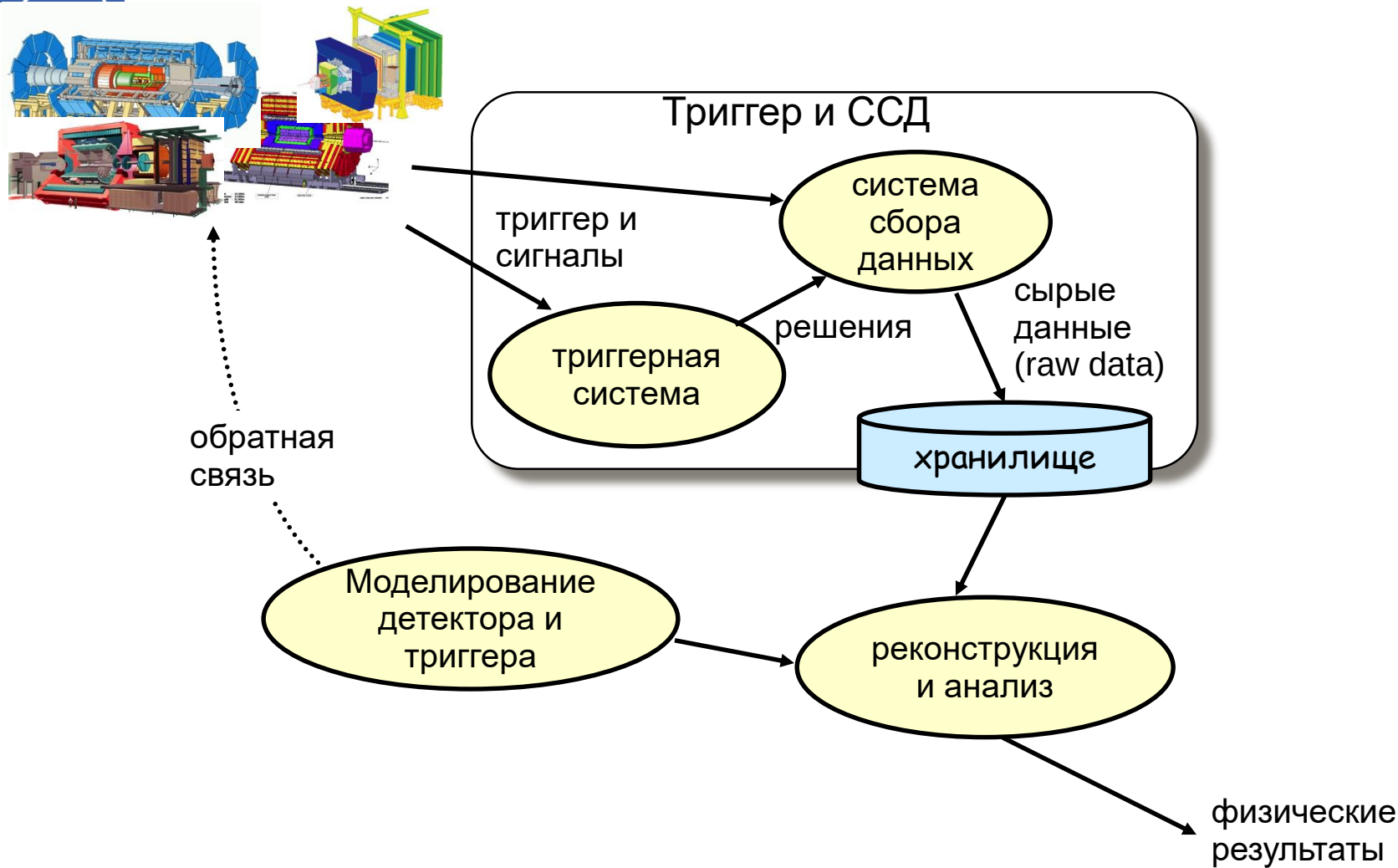
Смесь из физики, электроники, сетей и компьютеров

Требуется опыт и «хакерство»

Ресурсы (люди, деньги, время, место)



Общая схема



→ Основная роль триггера и ССД заключается в приеме и обработке сигналов с детекторов, с записью наиболее интересной информации на постоянный носитель для дальнейшего анализа (offline reconstruction & analysis)

Триггер и ССД

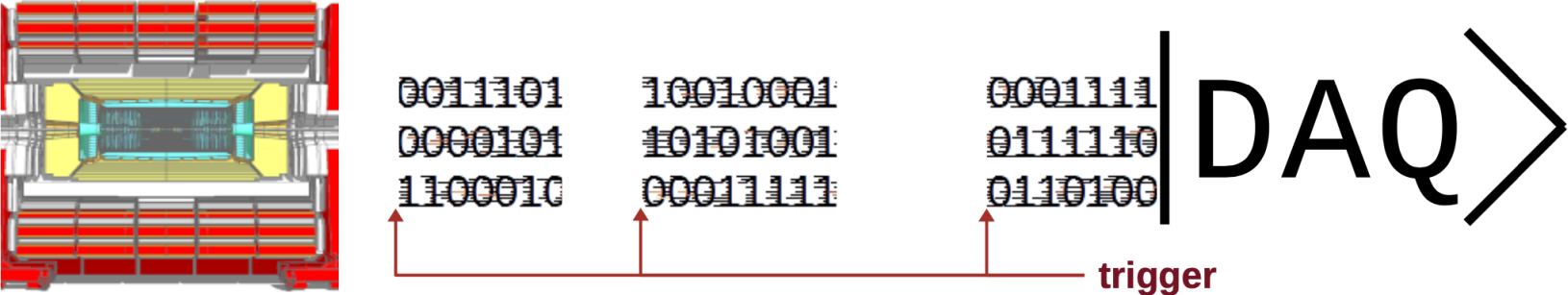
→ Триггер:

- Отбирает интересные события (events), и выкидывает ненужные, в режиме реального времени (real time) (то есть с минимальными задержками - latency)

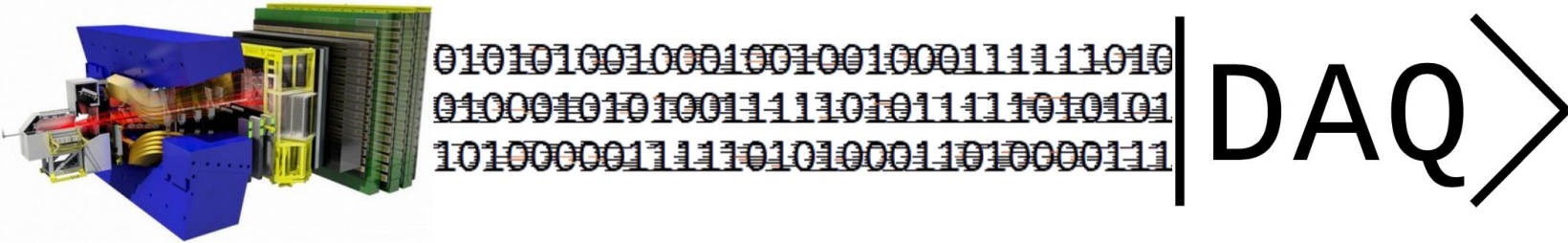
→ ССД:

- Собирает данные от детекторов - чтение (readout)
- Возможно инициирует триггер более высокого уровня (HLT)
- Собирает полные события – построение событий (event building)
- Записывает данные (data recording)
- Предоставляет возможность контроля, конфигурации и наблюдения процесса сбора данных (run control, monitoring)

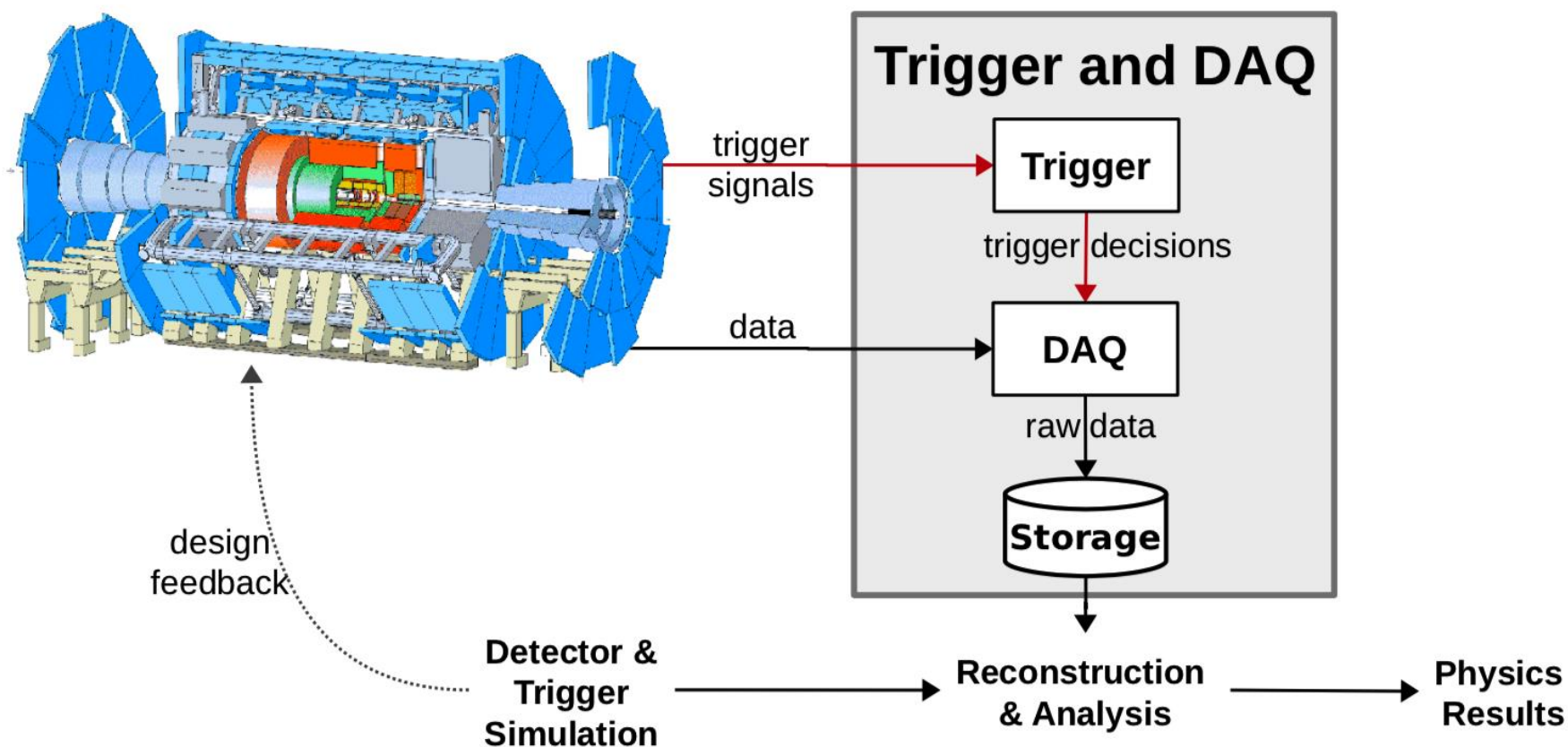
- Триггерные ССД
 - Данные читаются из детектора только при получении триггера



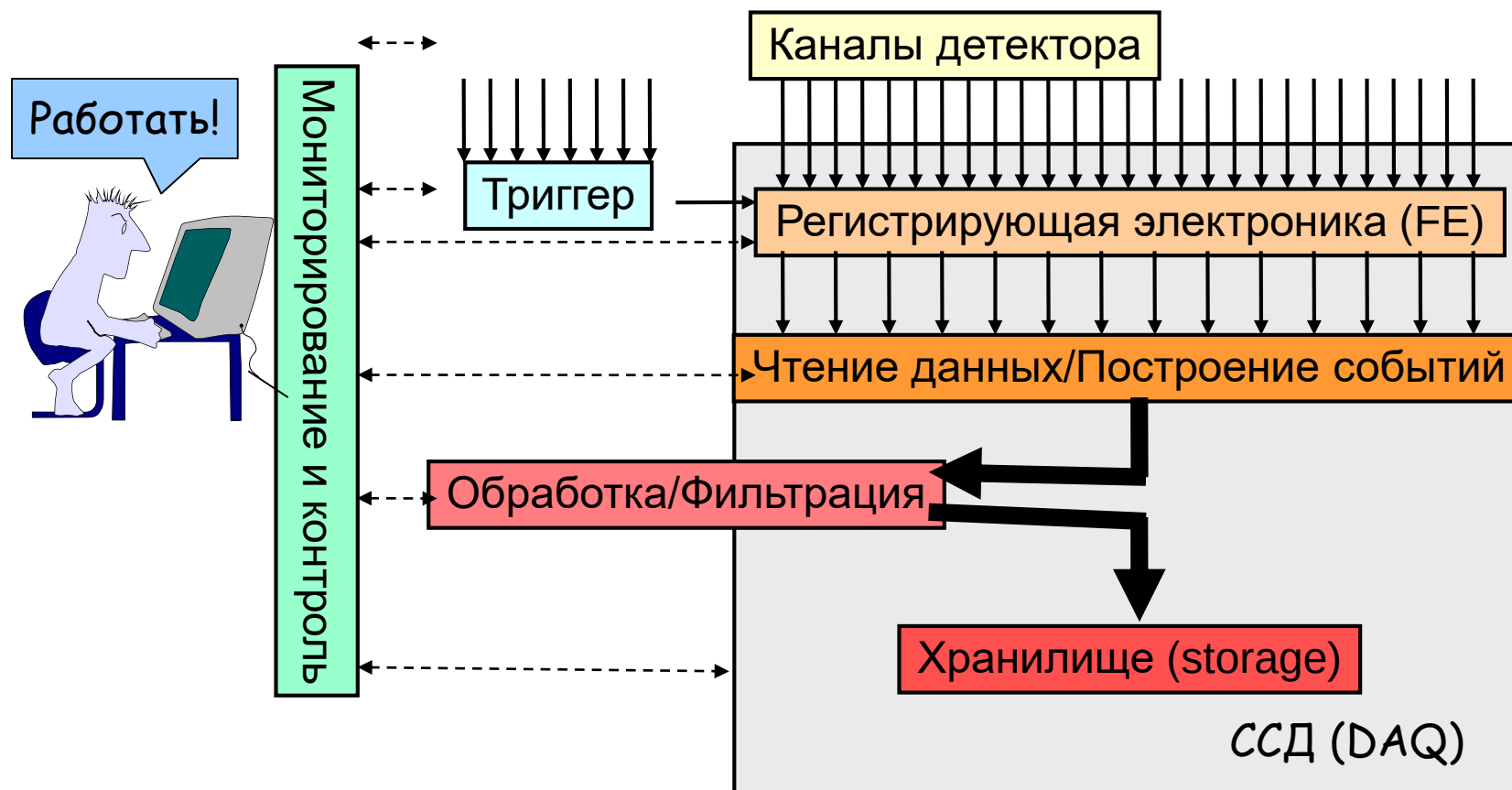
- Без-триггерные ССД
 - Данные читаются из детектора по мере поступления

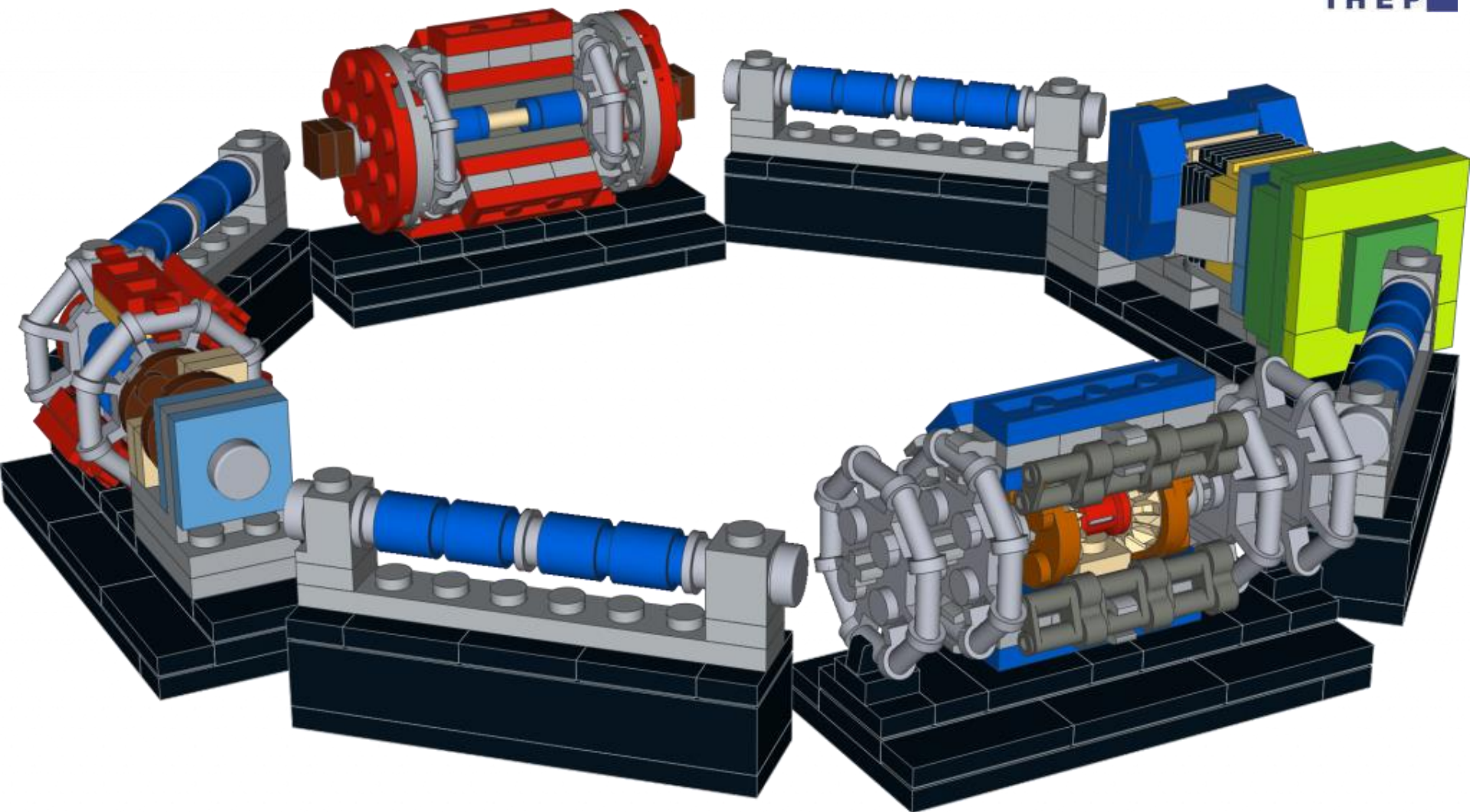


Триггер и ССД

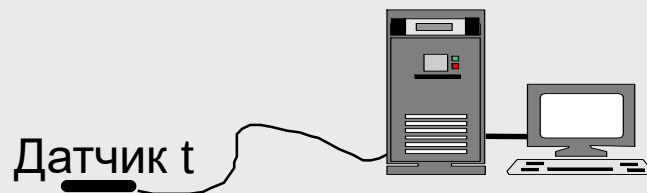


Триггер, ССД и контроль



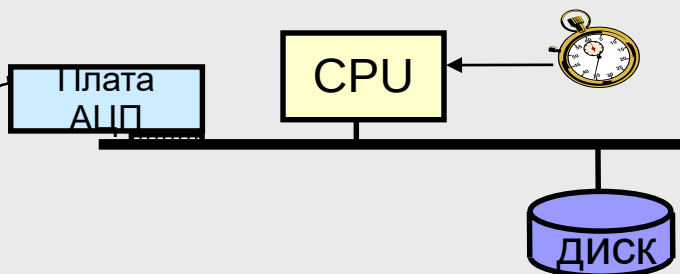


Внешний вид

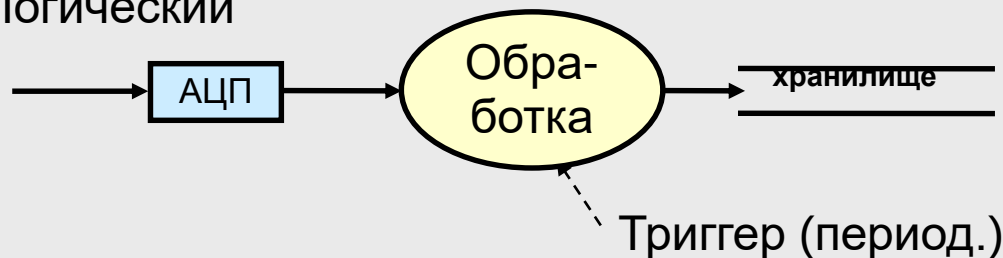


Физический

Датчик t



Логический



→ Измерение температуры с фиксированной частотой

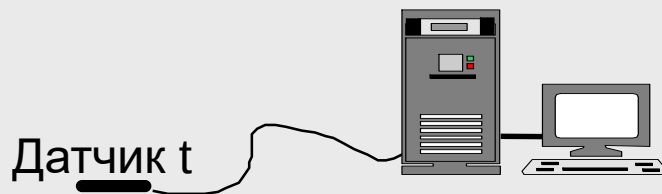
→ АЦП (ADC) переводит аналоговый сигнал в цифровой - оцифровка (**digitization**)

- Регистрирующая электроника
- (front-end electronics)

→ Процессор выполняет чтение и обработку

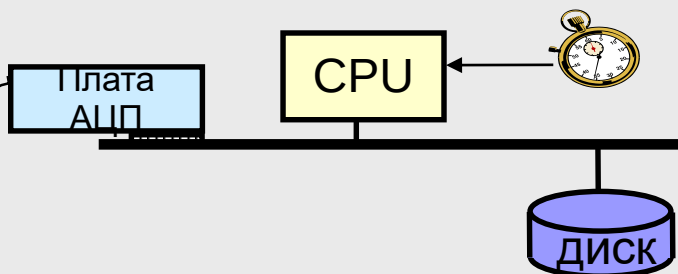
Простейшая ССД: периодический триггер

Внешний вид

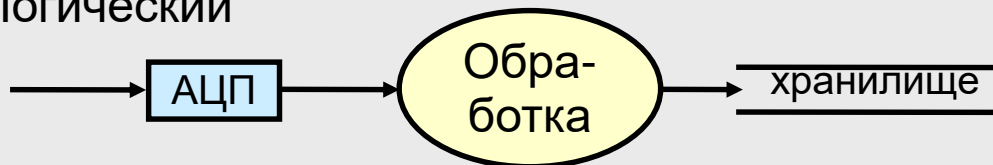


Физический

Датчик t



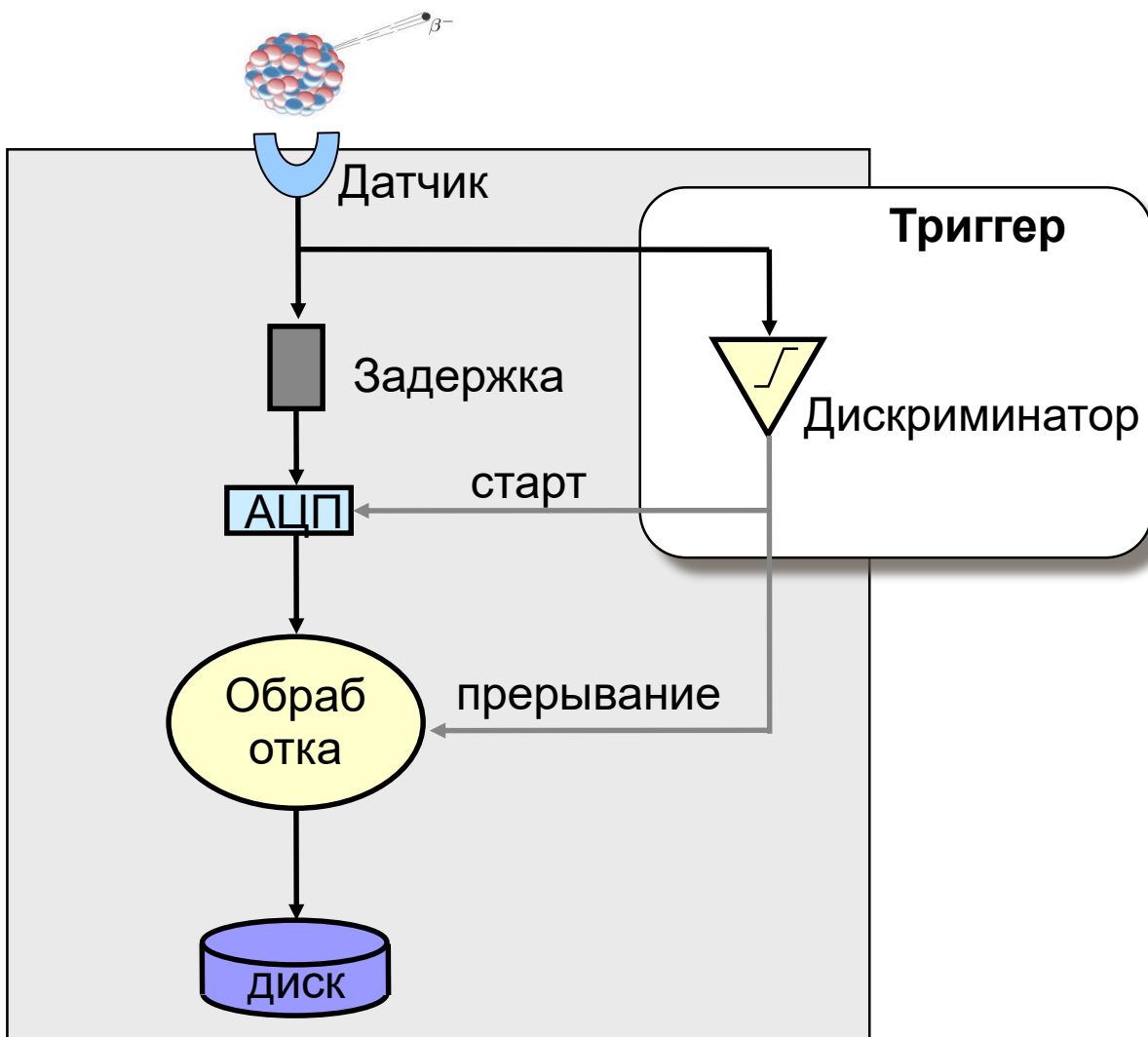
Логический



Триггер (период.)

- Измерение температуры с фиксированной частотой
- Система явно ограничена временем обработки события
- Например $\tau = 1$ мс для
 - АЦП преобразование + обработка + запись
- Выдерживает $\sim 1/1$ мс = 1 кГц частоту **периодического триггера**
- $R = 1/\tau$

Простейшая ССД: настоящий триггер



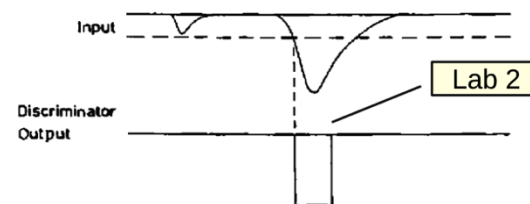
→ Измерение свойств бета-распада

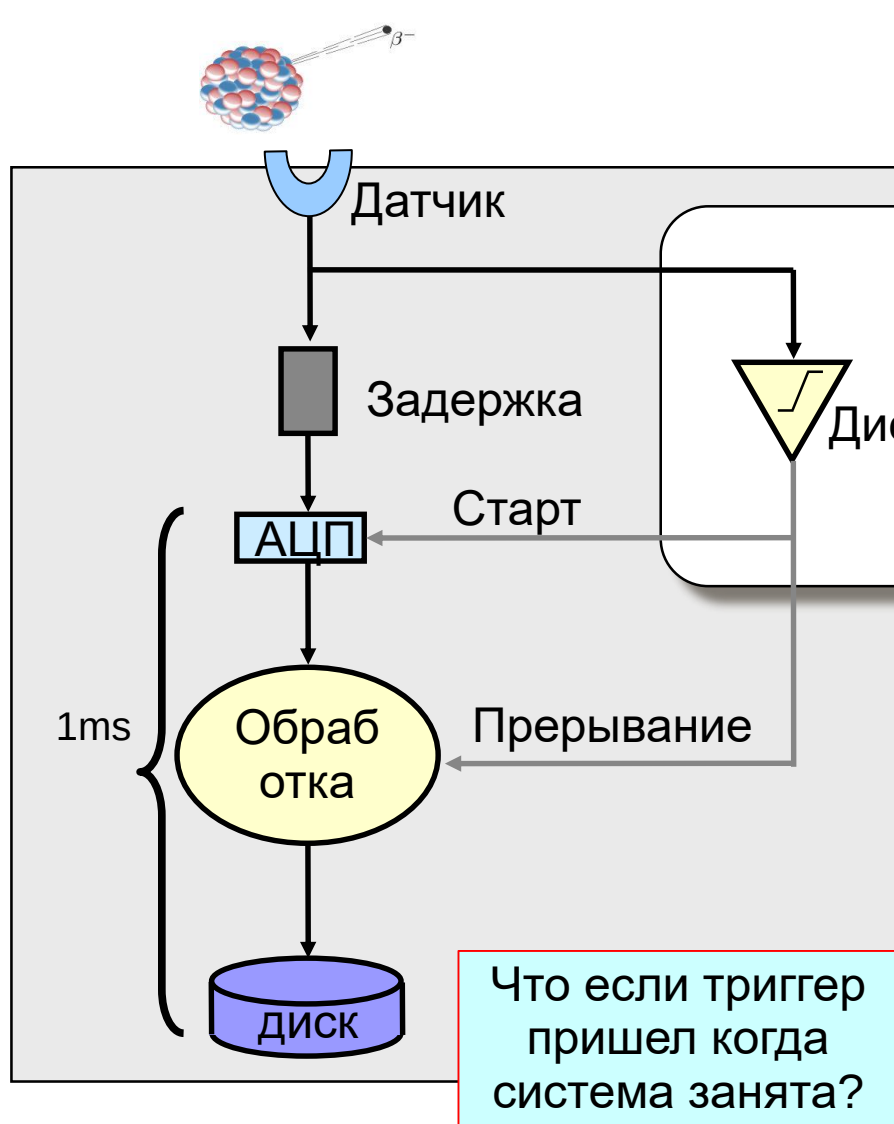
→ События приходят асинхронно и непредсказуемо

- Нужен **физический** триггер

→ Задержка компенсирует время выработки решения триггера

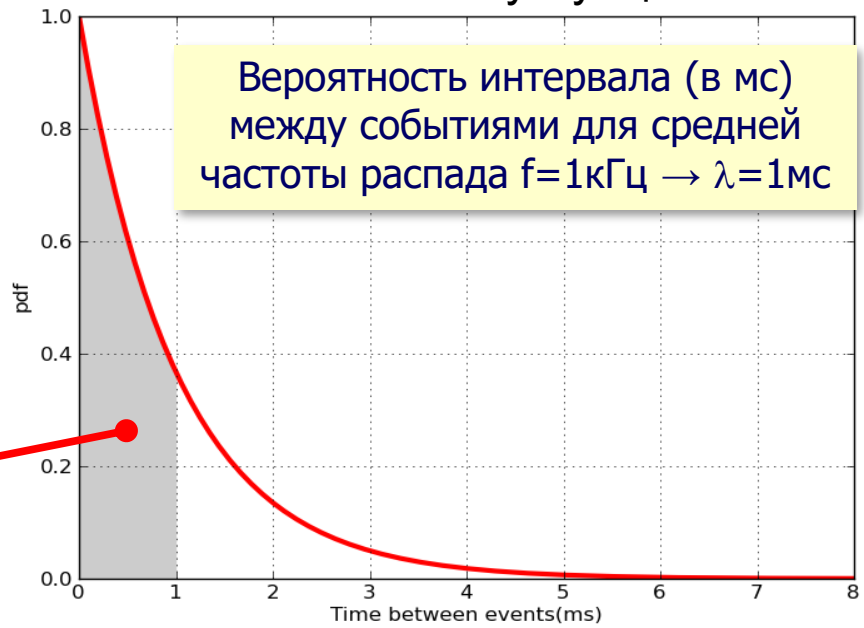
→ Дискриминатор – выдает триггер если входной сигнал больше порога





→ Измерение свойств бета-распада

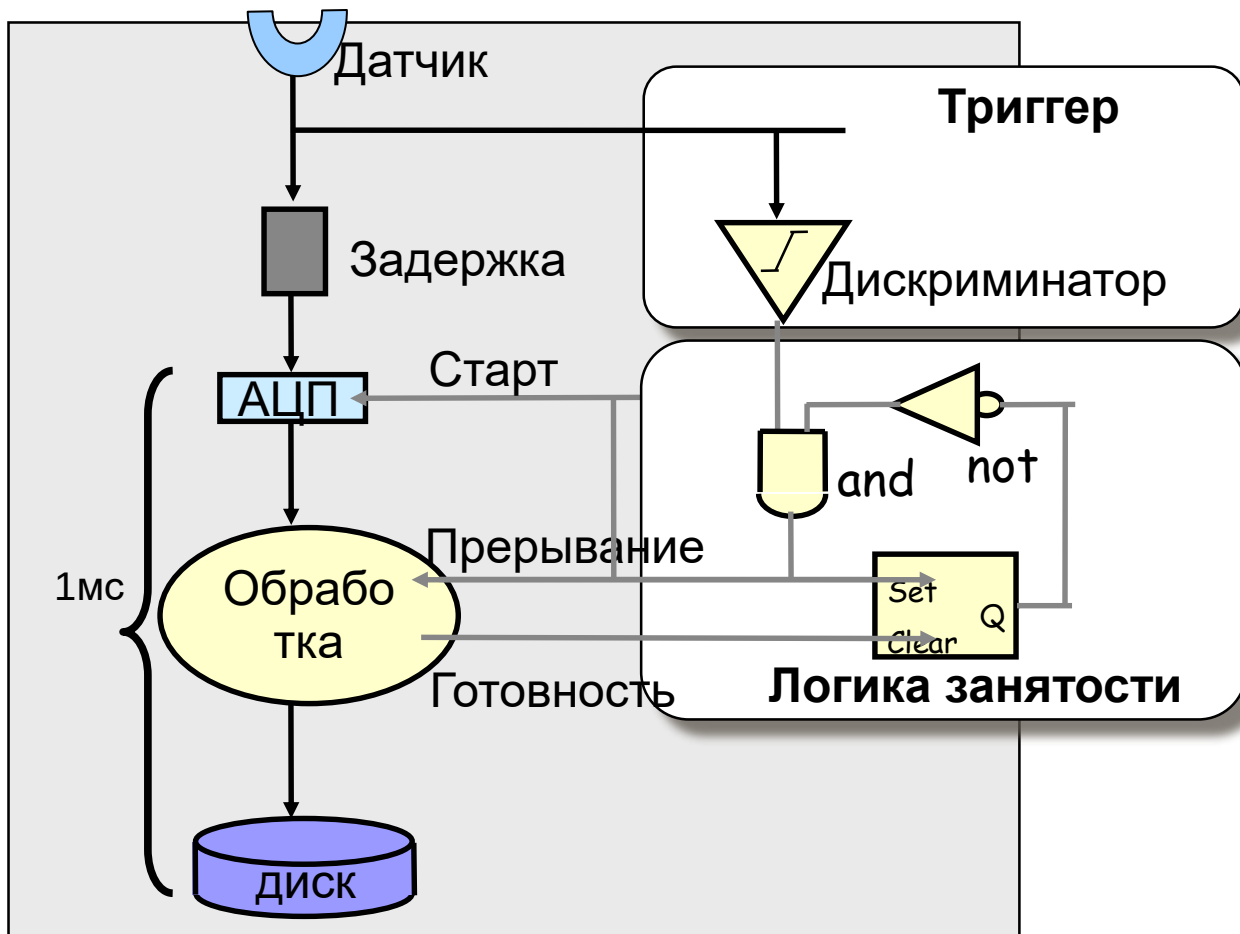
- Нужен **физический** триггер
- Стохастический процесс
- Флуктуации



Что если триггер пришел когда система занята?

Простейшая ССД: триггер и логика занятости

β^-
 $f=1\text{кГц}$
 $1/f=\lambda=1\text{мс}$



→ Логика занятости (busy logic) запрещает поступление триггеров во время обработки события

→ Какую (в среднем) частоту мы теперь сможем получить?

- Напоминание: $\tau=1\text{мс}$ было достаточно для работы на частоте 1кГц с периодическим триггером

Мертвое время и эффективность (1)

Define ν as average DAQ frequency

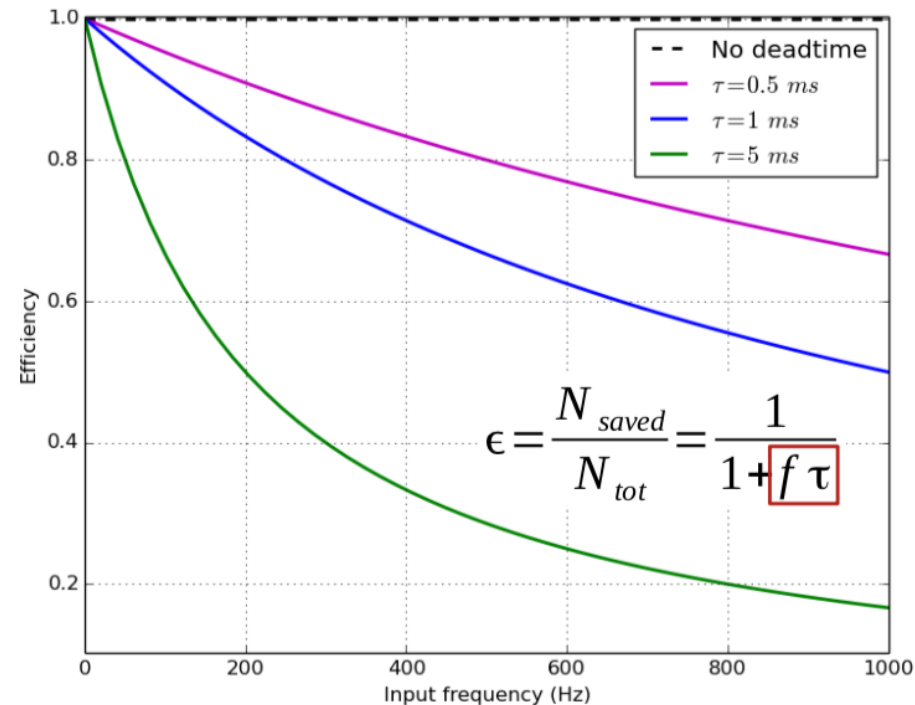
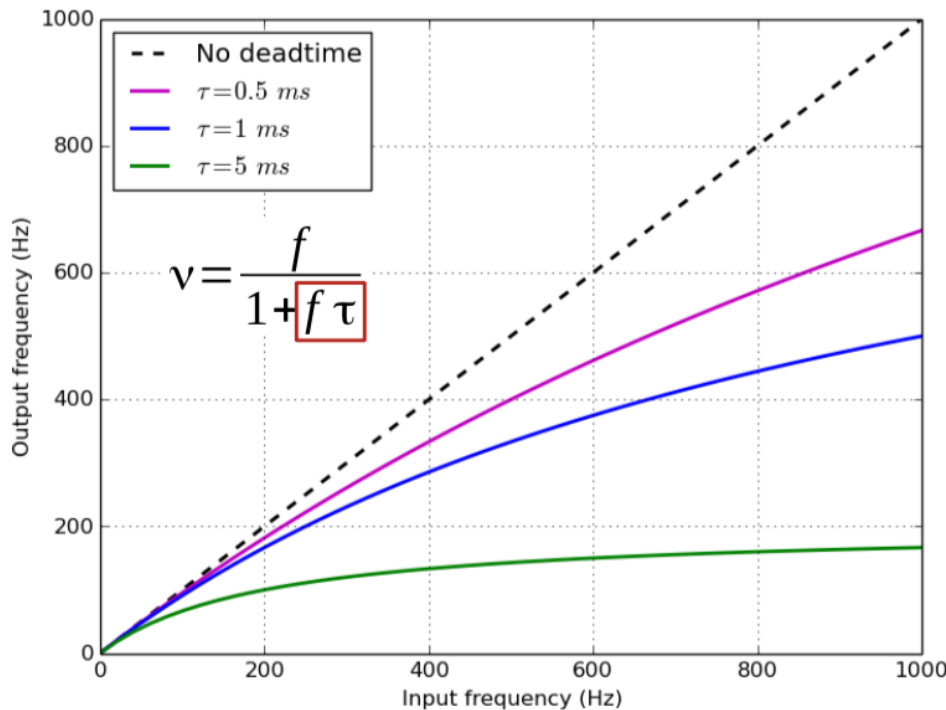
$\nu\tau \rightarrow$ DAQ system is busy - $(1 - \nu\tau) \rightarrow$ DAQ system is free

$$f(1 - \nu\tau) = \nu \rightarrow \nu = \frac{f}{1 + f\tau} < f$$

$$\epsilon = \frac{N_{saved}}{N_{tot}} = \frac{1}{1 + f\tau} < 100\%$$

- Определим мертвое время (d) как отношение между временем когда система была занята и полным временем. В нашем примере $d=0.1\%/Гц$
- Из-за флуктуаций внесенных стохастическим процессом эффективность всегда будем меньше 100%
 - В нашем примере, $d=0.1\%/Гц$, $f=1kHz \rightarrow \nu=500Hz$, $\epsilon=50\%$

Мертвое время и эффективность (2)

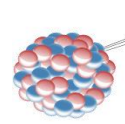


→ Если мы хотим получить $\nu \sim f$ ($\epsilon \sim 100\%$) $\rightarrow f\tau \ll 1 \rightarrow \tau \ll \lambda$

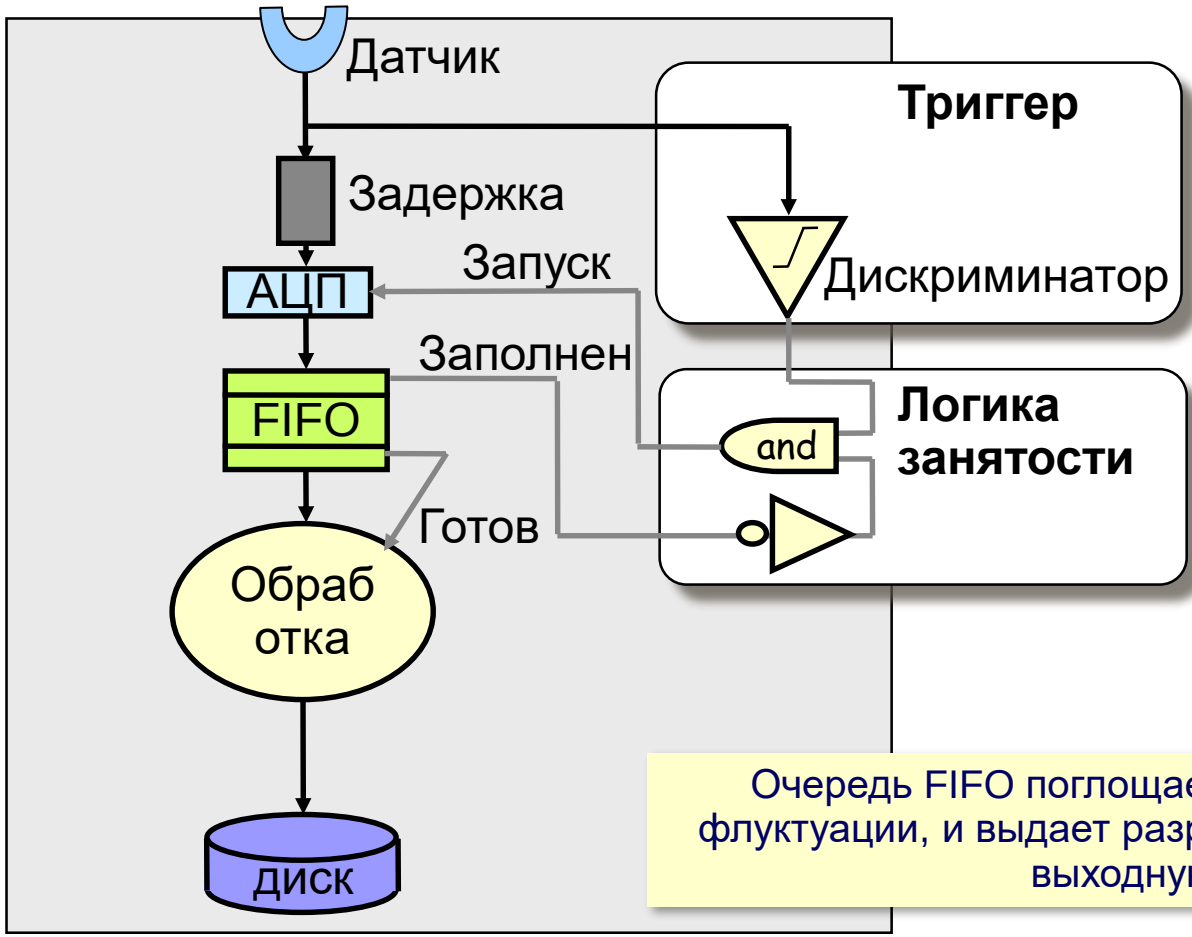
• $f = 1 \text{ kHz}$, $\epsilon = 99\% \rightarrow \tau < 0.01 \text{ ms} \rightarrow 1/\tau > 100 \text{ kHz}$

→ Чтобы справиться с флуктуациями частоты поступления сигнала, мы должны спроектировать ССД с запасом в 100 раз!!!

→ Можно ли обойти это требование?

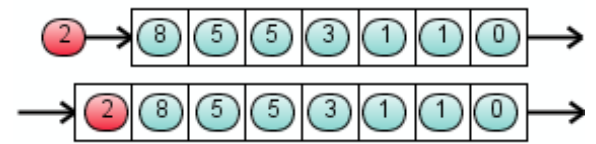


$f=1\text{кГц}$
 $1/f=\lambda=1\text{мс}$



→ Очередь (First-In First-Out – FIFO)

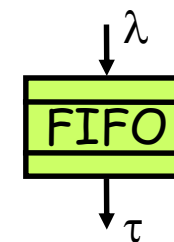
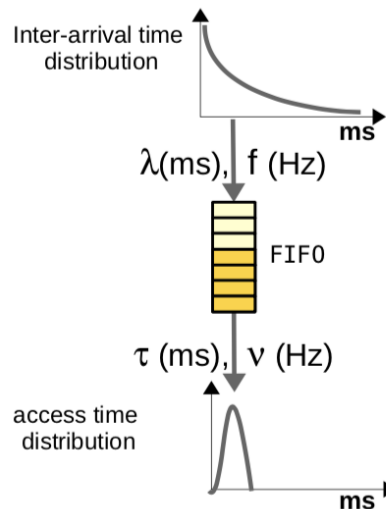
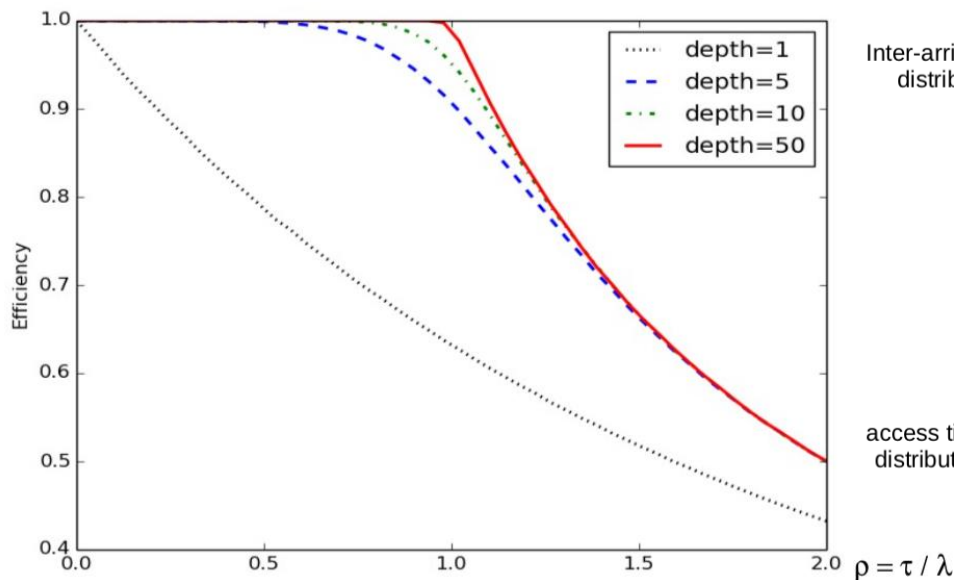
- Буферная зона организована в виде очереди
- Глубина: число ячеек
- Выполнена в «железе» (HW) или в ПО (SW)



→ Очередь FIFO добавляет дополнительную задержку

Очередь FIFO поглощает и сглаживает входные флуктуации, и выдает разравненную (De-randomized) выходную частоту

Разравнивание: теория очередей



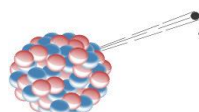
→ Теперь мы можем достигнуть эффективности FIFO $\sim 100\%$ с $\tau \sim \lambda$

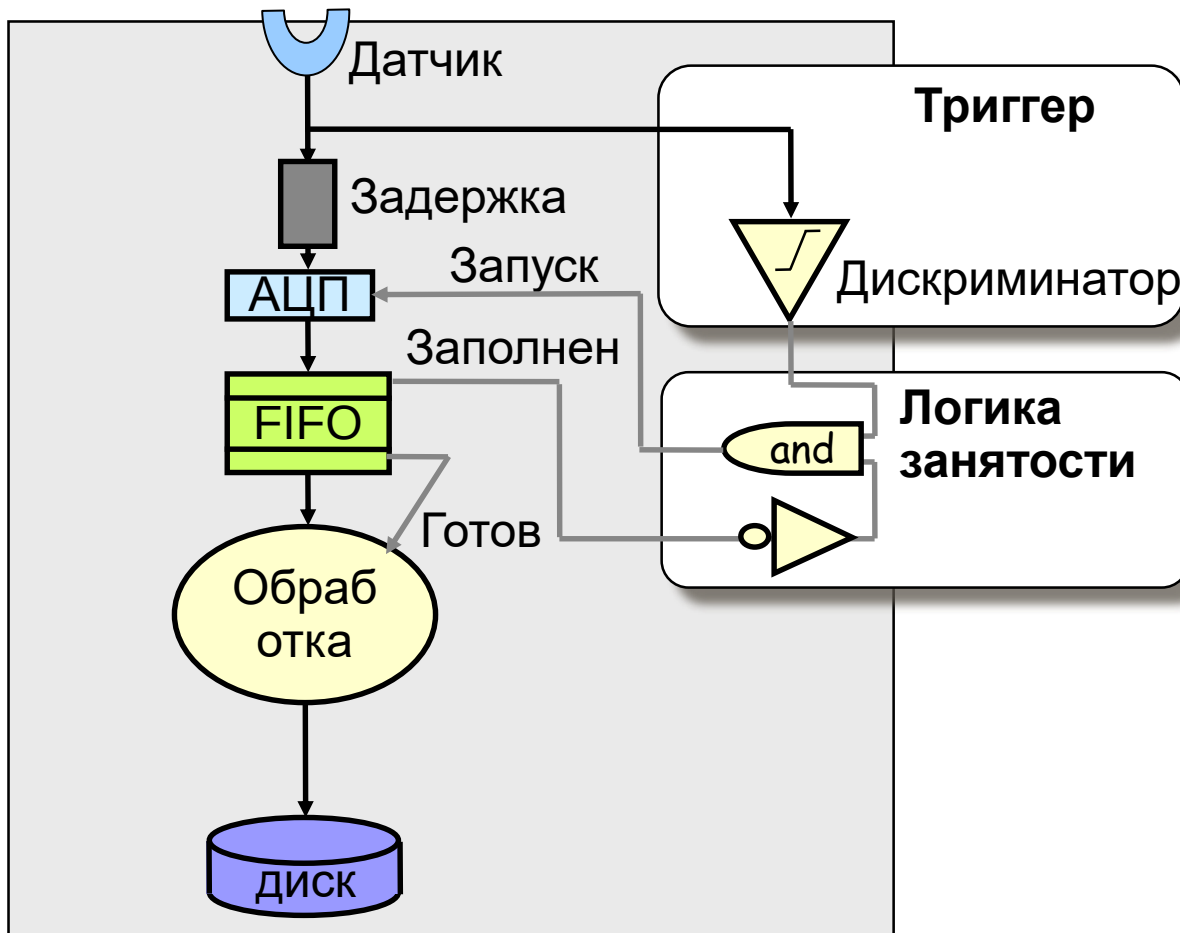
Умеренный
размер буфера

$\rho > 1$ – система перегружена
 $\rho \ll 1$ – система слишком большая
 $\rho \sim 1$ – система оптимальна

Аналитические вычисления возможны только для очень простых систем.
 В остальных случаях требуется тщательное моделирование.

Разравнивание: заключение


 $f=1\text{kHz}$
 $1/f=\lambda=1\text{ms}$



→ Можно достичь почти 100% эффективности и минимального мертвого времени при условии

- АЦП может работать на частотах $\gg f$
- Частота обработки данных $\sim f$

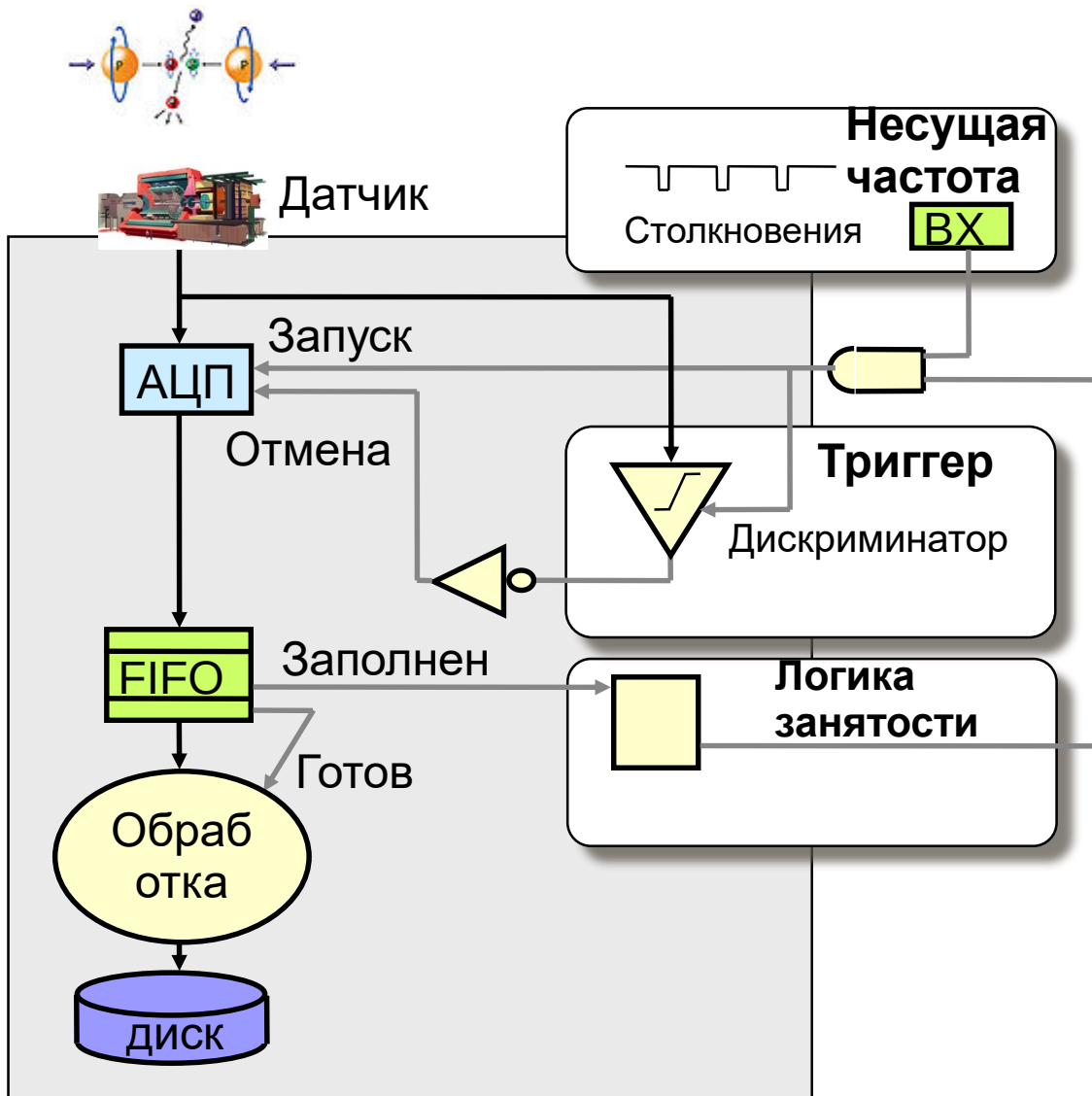
→ FIFO отделяет быстродействующую регистрирующую электронику от обработки данных

- Минимизирует число быстрых компонентов

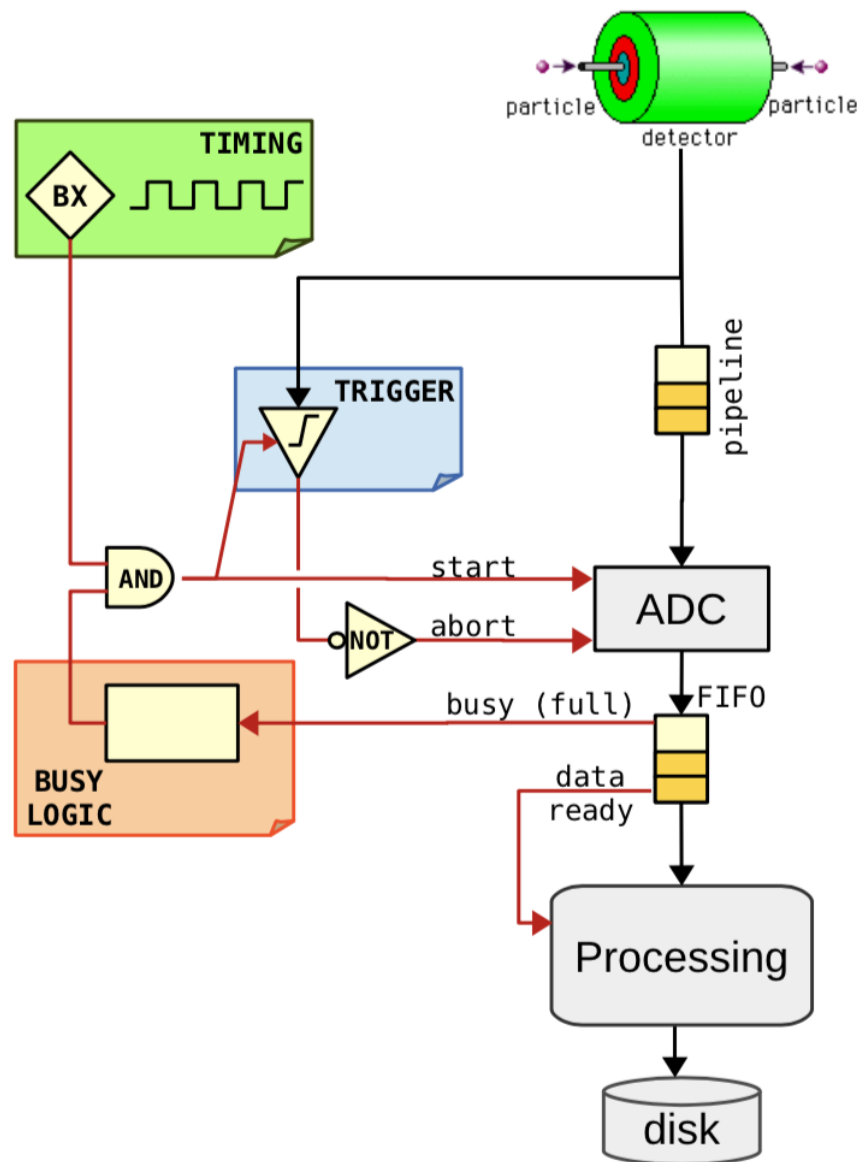
→ Можно ли заменить задержку на “FIFO”?

- Аналоговые задержки → широко используются на экспериментах LHC

Простейшая ССД: режим коллайдера



- Столкновения частиц синхронны (с задающей частотой)
- Триггер отбрасывает неинтересные события
- Даже если столкновения синхронны, то триггера (т.е. хорошие события) **непредсказуемы**
- Необходимо **разравнивание** (de-randomization)
- Более сложная логика занятости, например – принять N событий за M пересечений



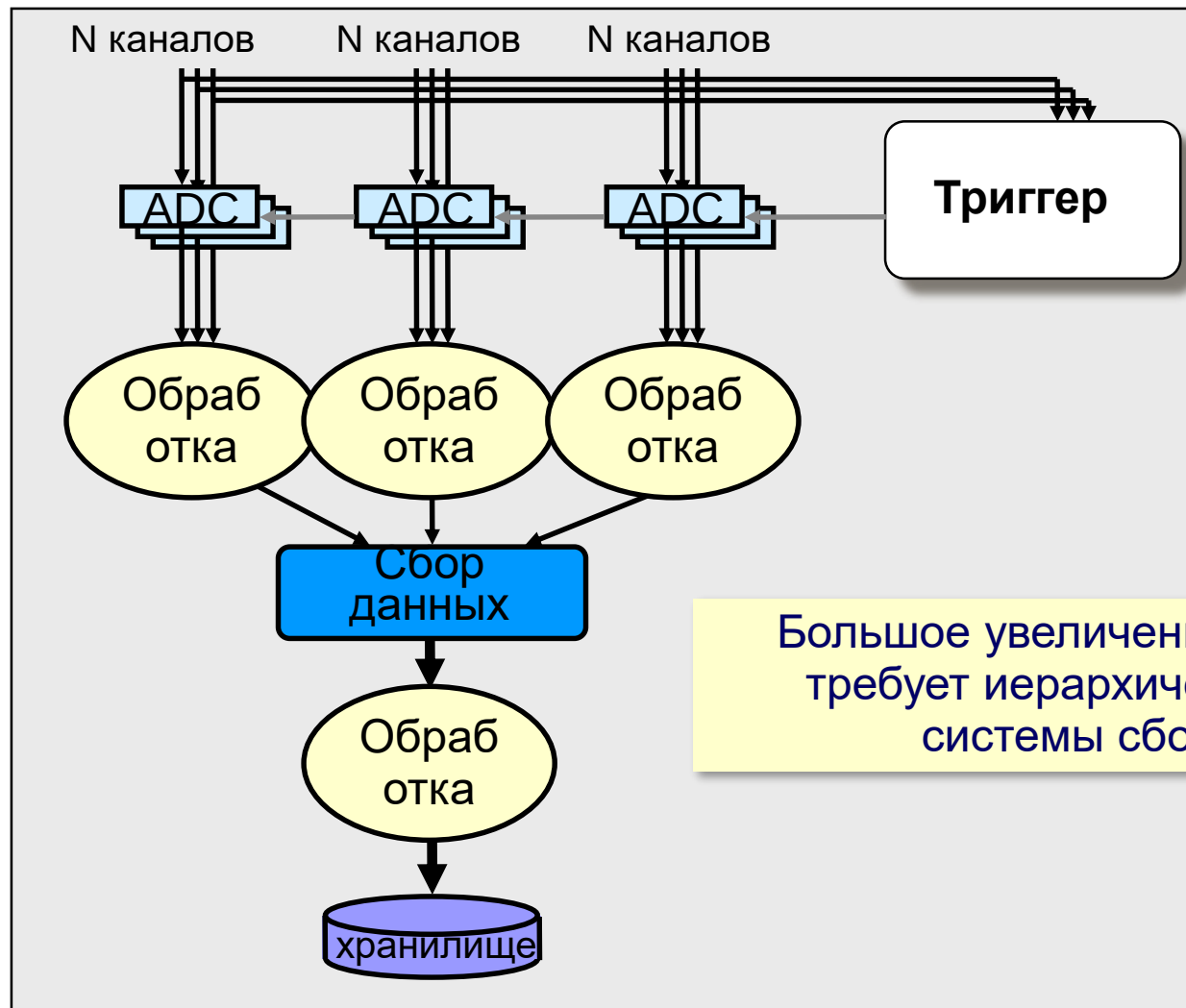


Масштабируемость

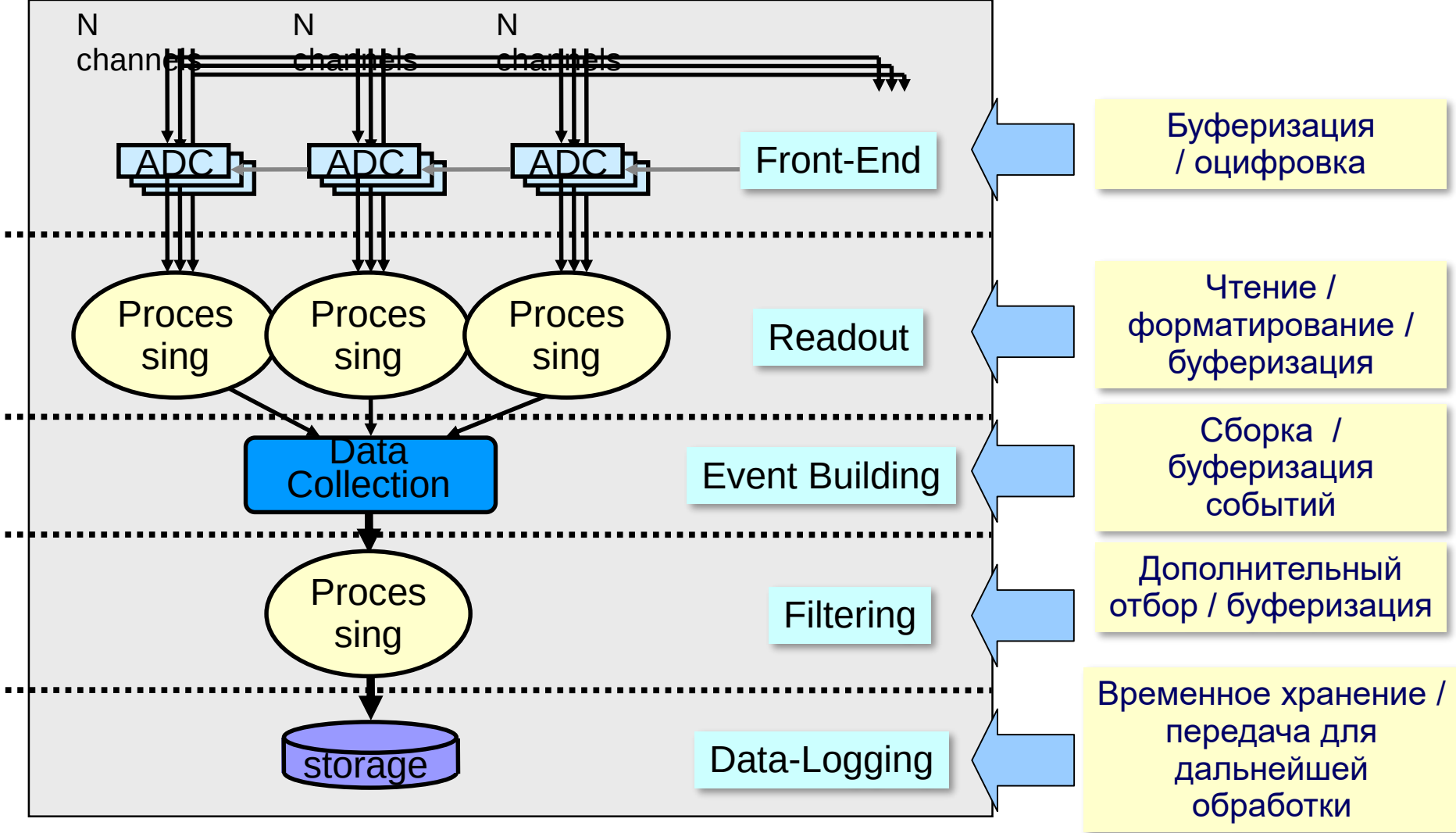


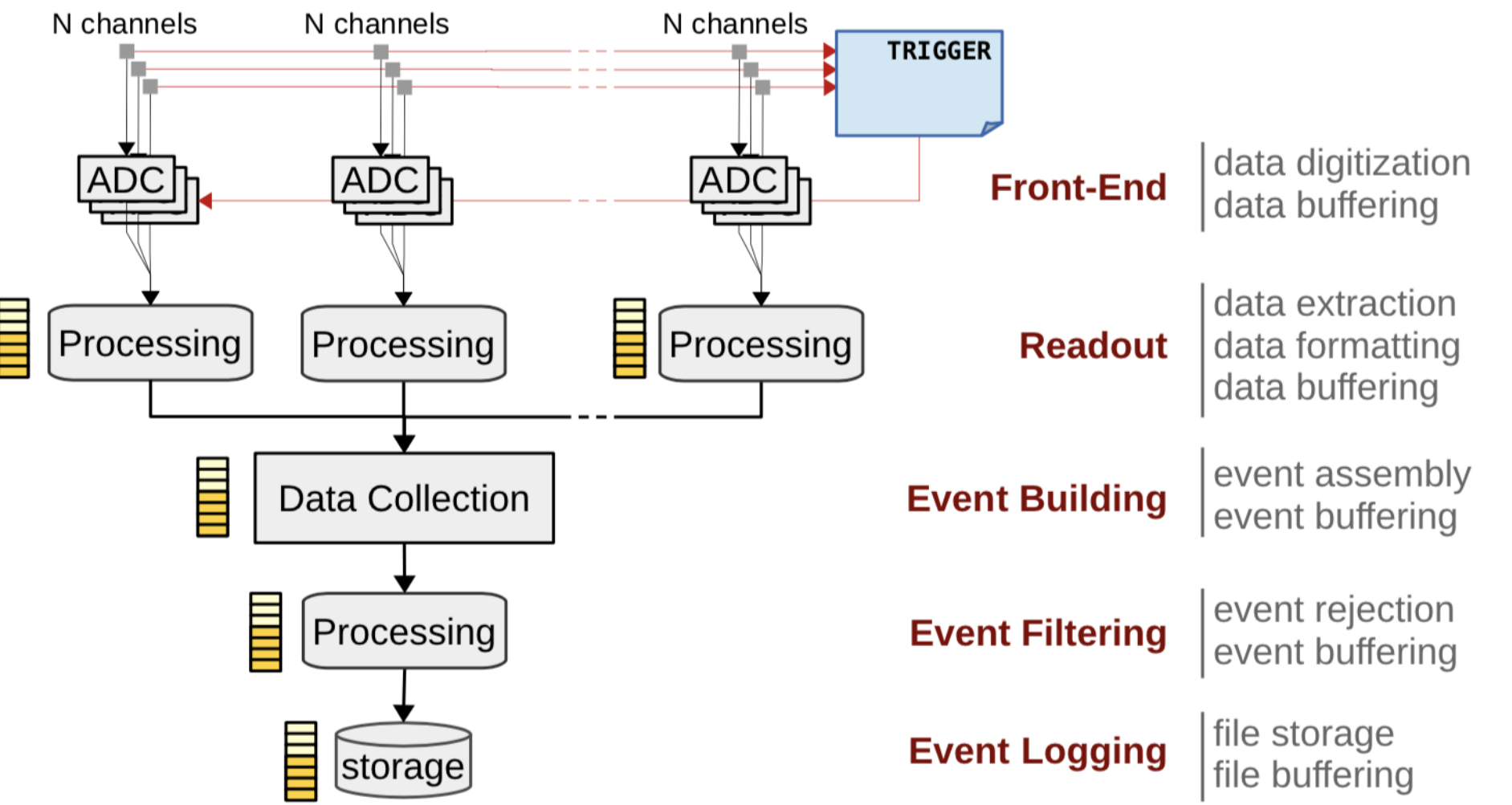
?

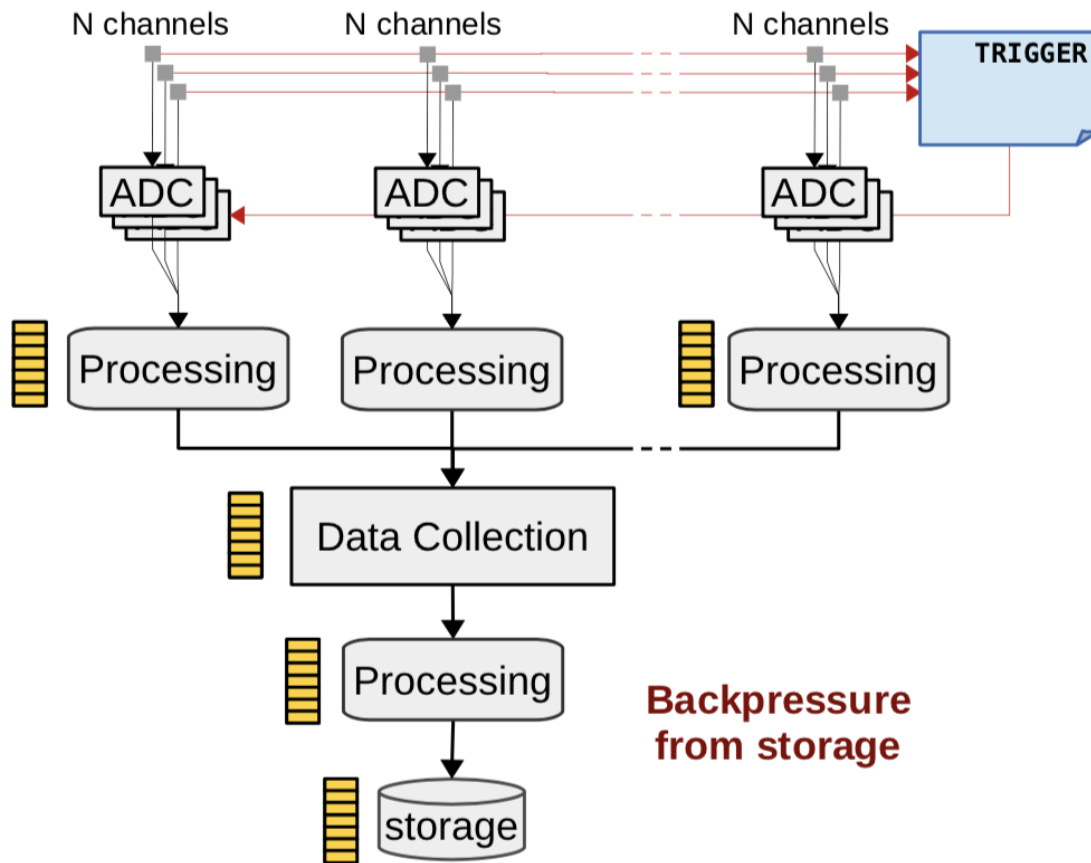




Большое увеличение числа каналов требует иерархической структуры системы сбора данных





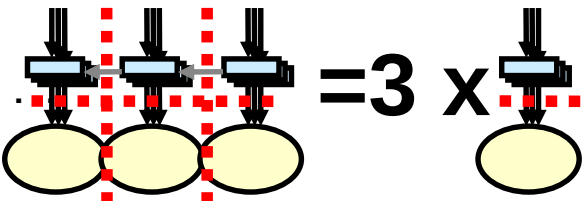


- ✓ Если система/буфер переполнились – давление передается обратно
- ✓ Недостаточная пропускная способность
- ✓ Недостаточные вычислительные ресурсы

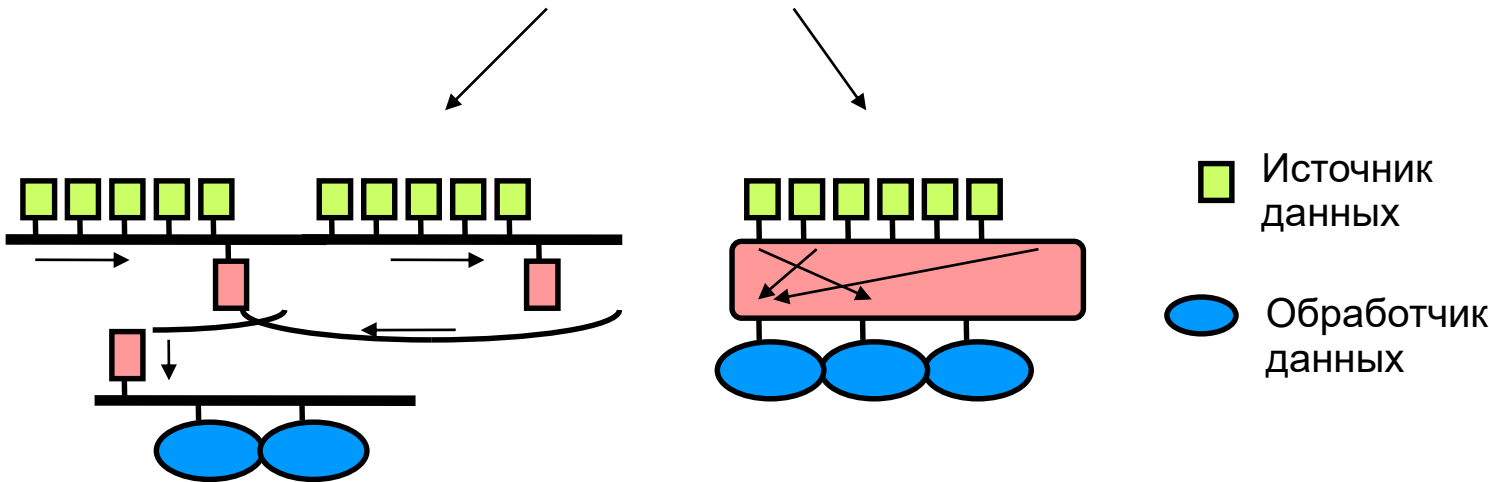
Топология систем сбора данных

- Чтение или построение событий из множества каналов требует множество компонентов
- В дизайн нашей иерархической системы сбора данных стоит добавить «строительные блоки»

- например: каркасы, вычислительные узлы, .



- Как организовать связи внутри и между блоками?
- Два основных класса: **шина (bus)** и **сеть (network)**



Шины

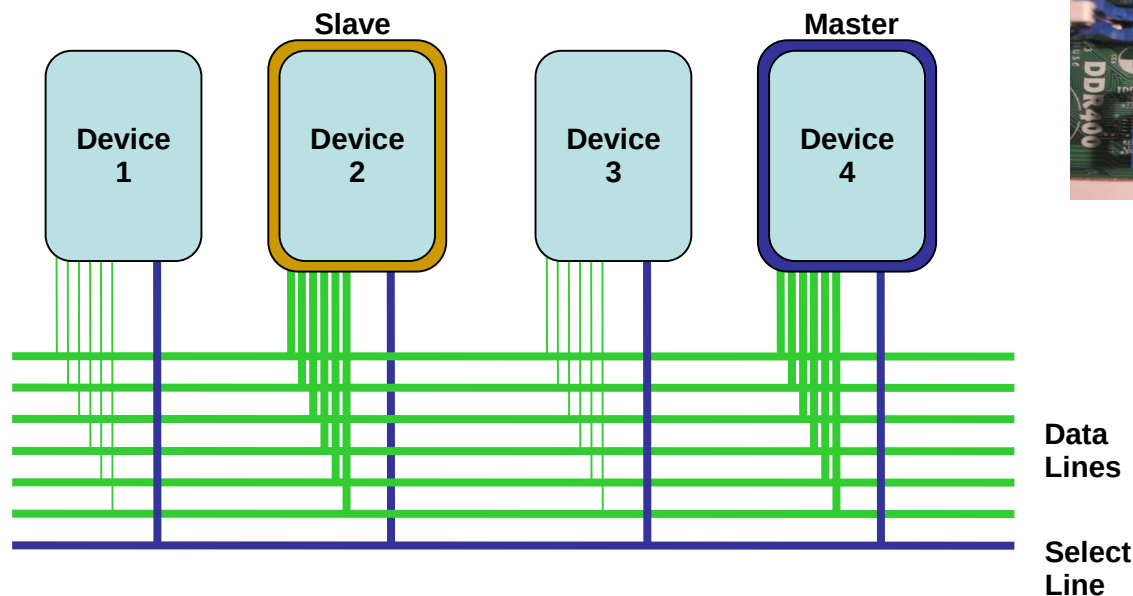
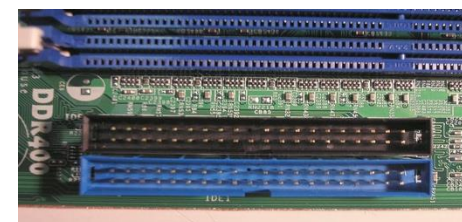
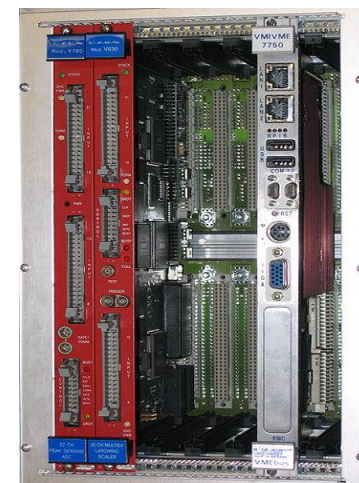
→ Примеры: CAMAC, VME, PCI, SCSI, Parallel ATA, ...

→ Устройства соединены **общей** шиной

- Шина → набор проводников
- Общая шина подразумевает арбитраж

→ Устройства могут быть **мастером** или **подчиненным**

→ Устройство может быть адресовано (однозначно идентифицировано) на шине



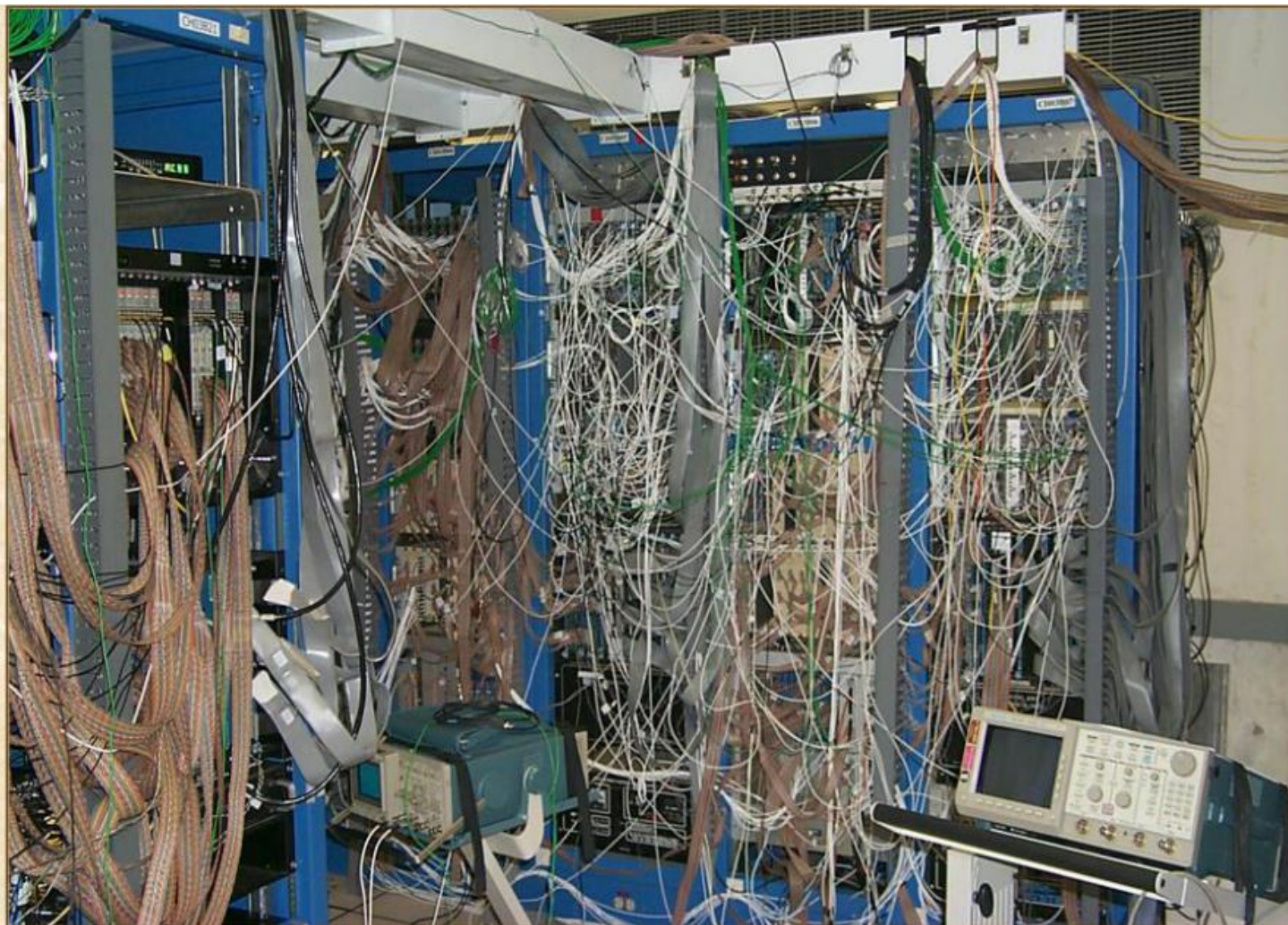
→ Простота 😊

- Фиксированное число линий (ширина)
- Устройства должны соответствовать жестко определенным интерфейсам
 - механическим, электрическим, ...

→ Проблемы масштабирования ☹️

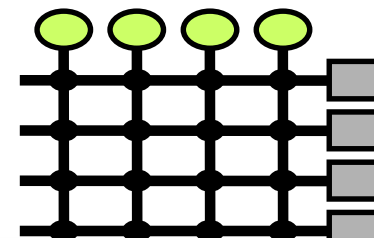
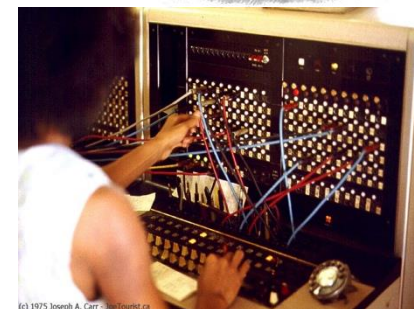
- Пропускная способность делится между устройствами
- Максимальная ширина довольно ограничена
- Максимальная частота шины обратно пропорциональна ее длине
- Максимальное число устройств на шине зависит от ее длины
- Другие «эффекты» могут создать проблемы в дальнейшем

Свойства шин



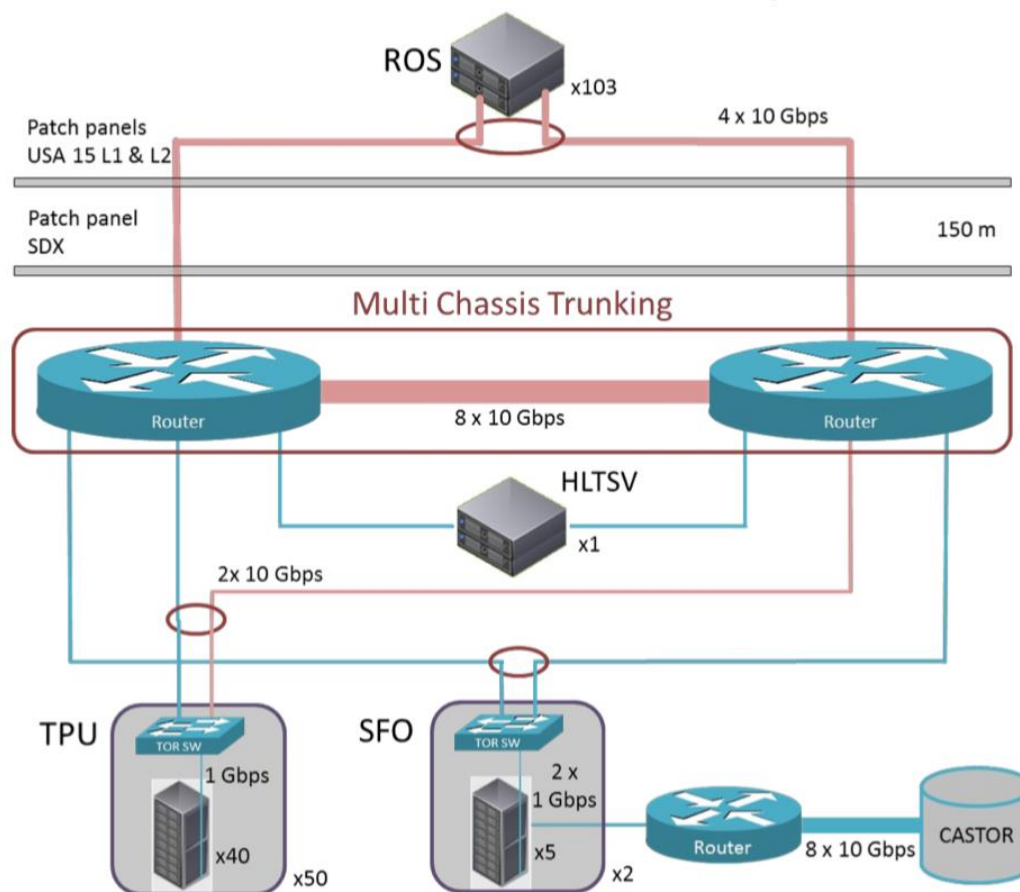
В дальней перспективе, **другие “эффекты”** могут ограничить масштабируемость

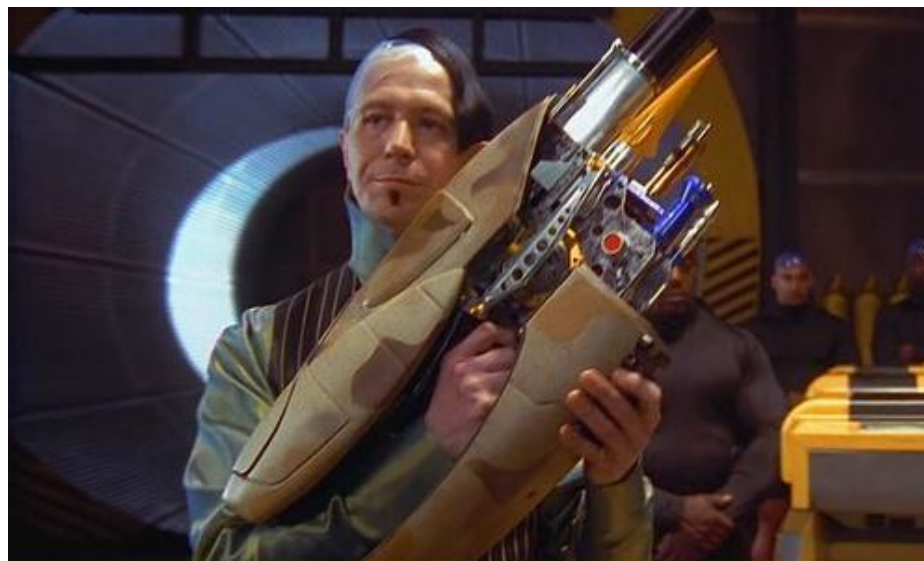
- Примеры: телефон, Ethernet...
- Все устройства **равны**
- Устройства **общаются напрямую** друг с другом
 - Нет арбитражи, одновременное взаимодействие
- Устройства обмениваются сообщениями
- В коммутируемой сети, **коммутаторы** передают сообщения между отправителем и получателем
 - Находят правильный путь
 - Справляются с “затором” (два сообщения одному получателю одновременно)
 - Удивительно что ключом к успеху и тут является буферизация...



Благодаря этим характеристикам, **сети хорошо масштабируются**. Они являются основой (кровеносной системой) систем сбора данных на LHC.

- ✓ Хорошо масштабируются
- ✓ Являются основой систем сбора данных на LHC





«Все надо делать самому»





Напутствия разработчику ССД



→ Изучение характеристик триггера.

- Периодический или стохастический - постоянный или «сгустками» - отсеивание или выборка

→ Необходимая эффективность

- Хорошо иметь запас, но слишком большой размер следует избегать

→ Следует определить источники флуктуаций и установить адекватные размеры буферизации.

- Внимание: (детерминированные) сложные системы вносят свои флуктуации: многопоточные процессы, сетевые соединения, ...

→ Подходящий буфер это небольшой буфер.

- Большой буфер делает систему менее стабильной и менее отзывчивой, подверженной к колебаниям. В общем случае ухудшает надежность.

→ Чем проще и меньше свободных параметров тем лучше.

→ Решение проблем требует настойчивости

- Очень редкий сбой в системе может быть симптомом гораздо больших проблем

→ В любом случае, ...

DON'T PANIC

