2023

Связь FIFO с протоколом valid/ready
Применение valid/ready в шинах типа AXI
Конвейеры с контролем потока данных

Дмитрий Смехов

ЛяЛямбда 2023

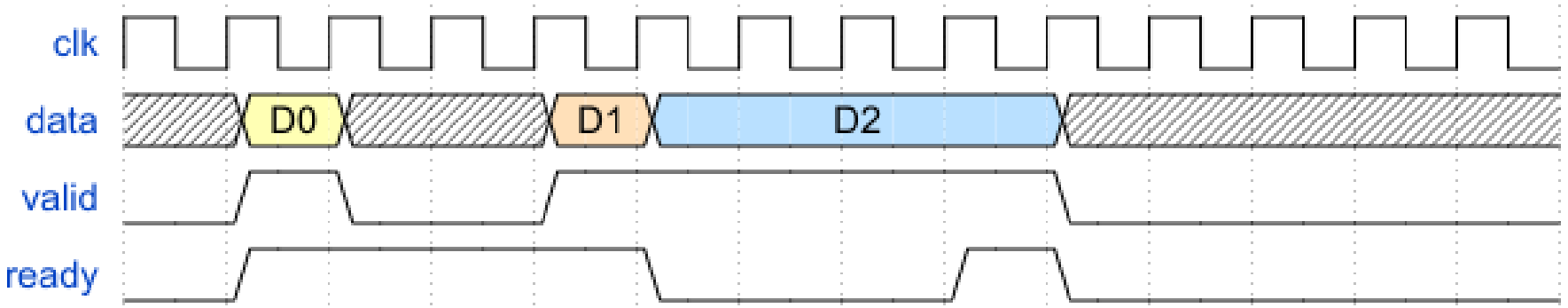
Дополнительные материалы

Дмитрий Смехов - Доклад на ChipExpo 2021:

«Интерфейс valid/ready. Двойные буфера. Счётчики кредитов. Организация конвейера».

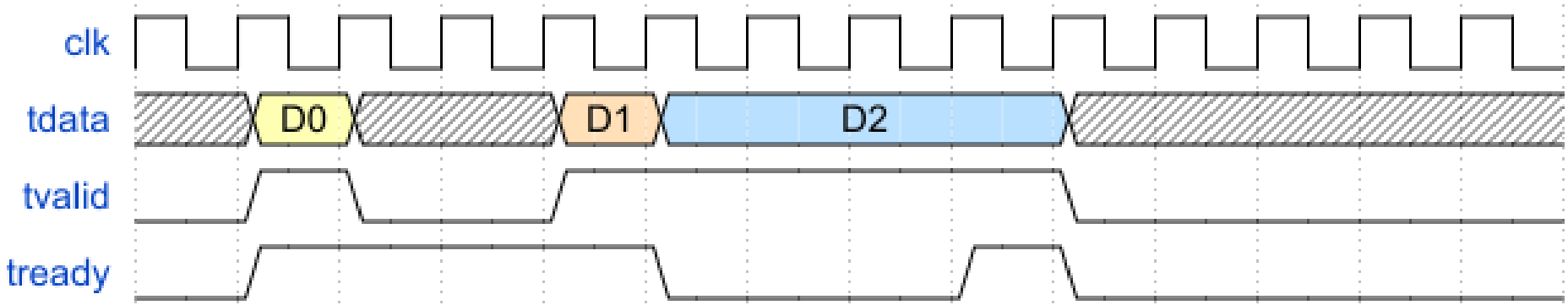
https://github.com/DigitalDesignSchool/ce2020labs/blob/master/next_step/dsmv/presentation/pr_1_crd.pptx

Интерфейс valid/ready



Слово данных считается переданным когда на фронте тактового сигнала установлены оба сигнала: **valid** и **ready**

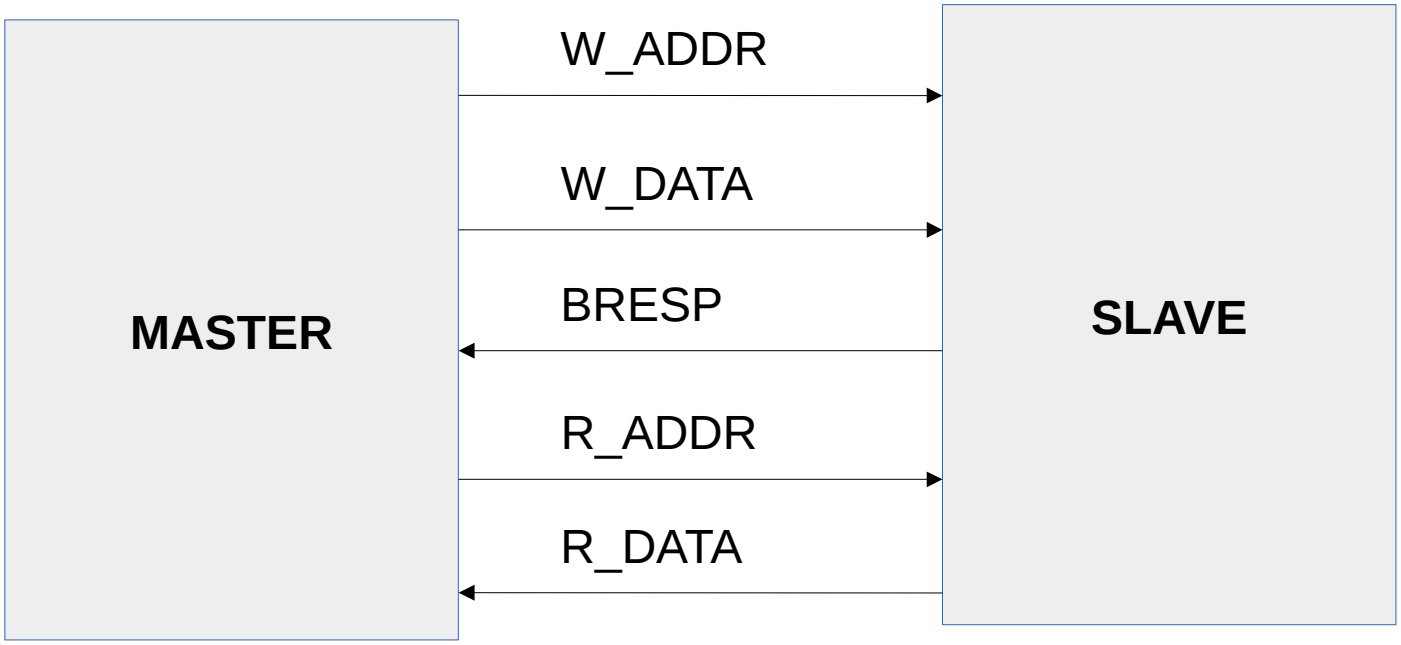
AXI STREAM



Дополнительные сигналы (**tlast**, **tuser**) передаются по тем же правилам что и **tdata**:

Передача происходит только при **tvalid**=1 и **tready**=1

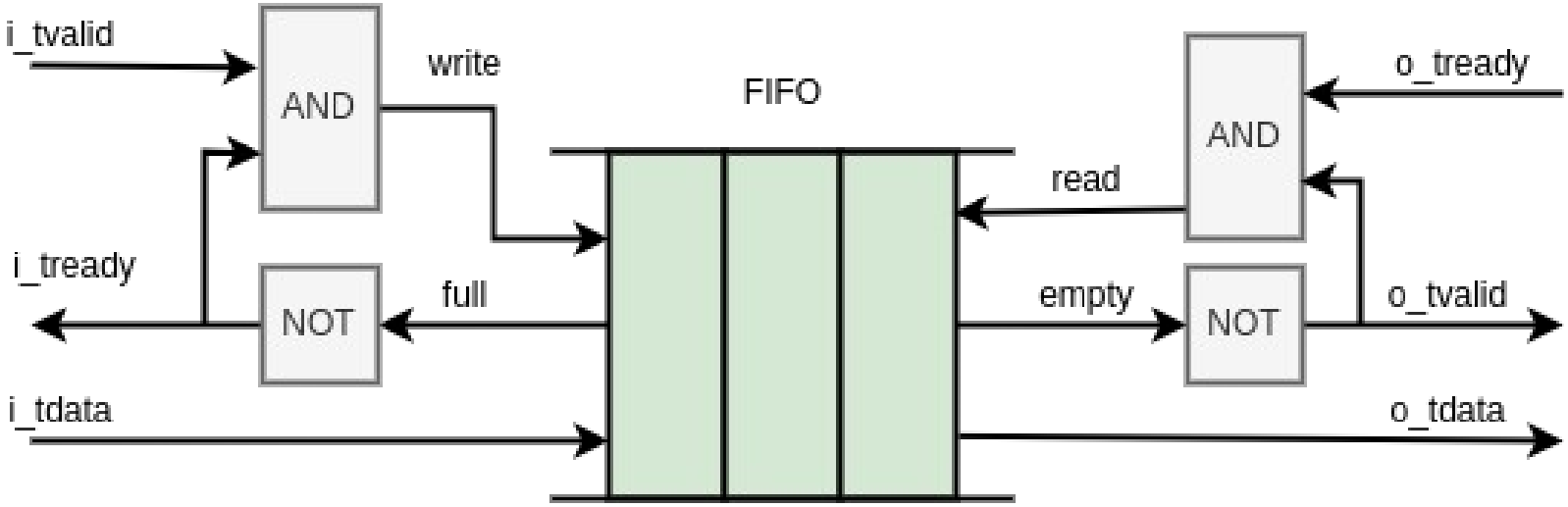
AXI MEMORY MAP



Каждая шина содержит сигналы **valid** и **ready**

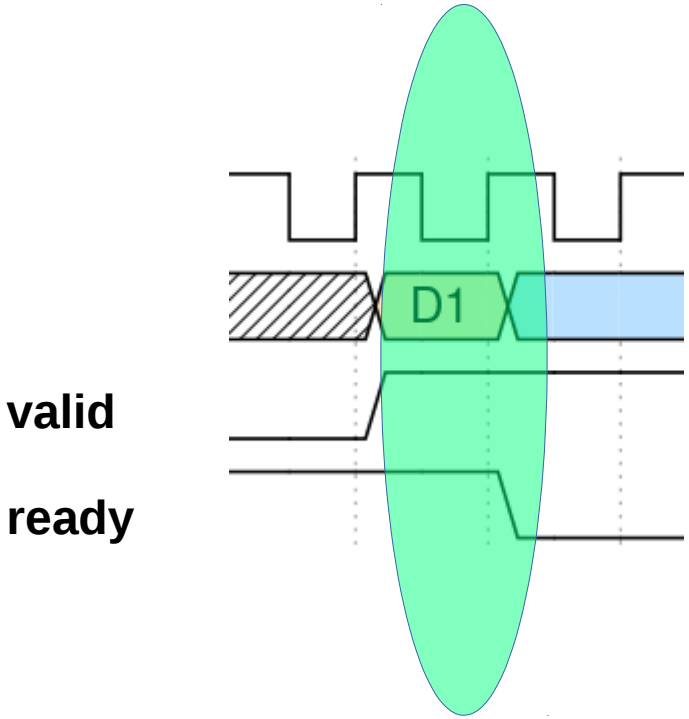
Передача происходит только при **valid**=1 и **ready**=1

Связь FIFO и AXI STREAM



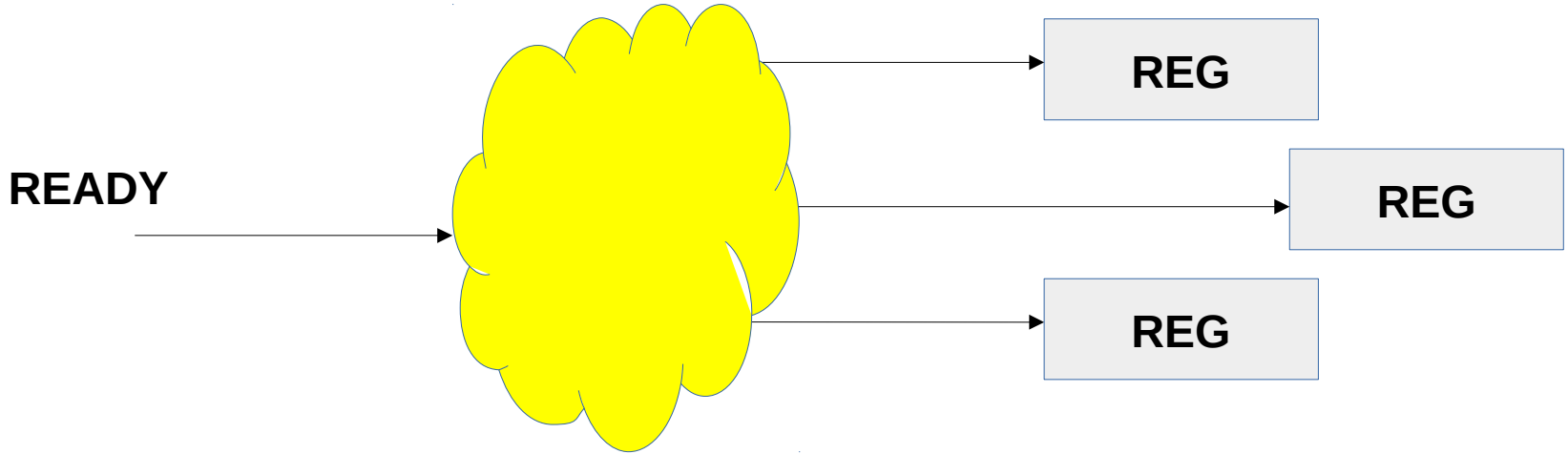
Добавление небольшой комбинационной схемы согласует порты FIFO и сигналы шины AXI STREAM

Главная проблема valid/ready



- На передатчике сигнал **ready** должен быть обработан в том же самом такте.
- Нет возможности подать сигнал на триггер.
- Это достаточное основание для отказа от **valid/ready**

Проблемы для автомата передачи

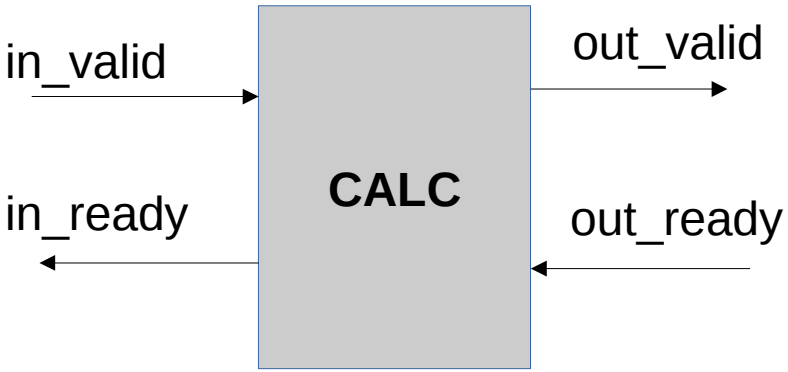


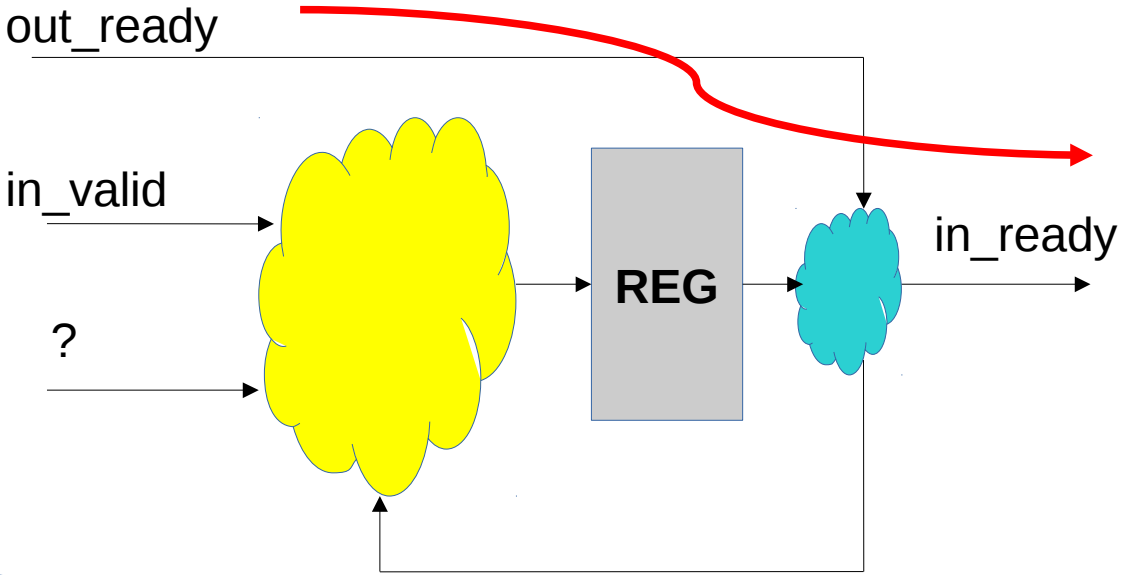
Как правило существует конечный автомат который передаёт данные на шину.

Сигнал **ready** должен через некую комбинационную логику попасть на все триггеры автомата.

Это создаёт большие сложности при трассировке цепей

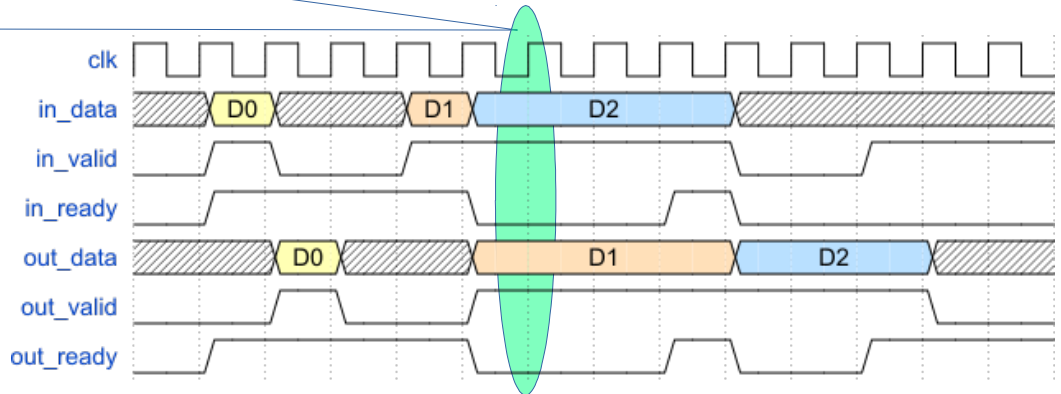
Комбинационное формирование ready



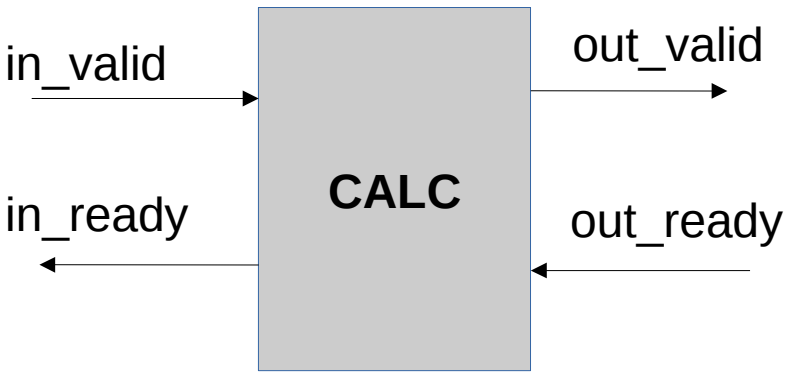


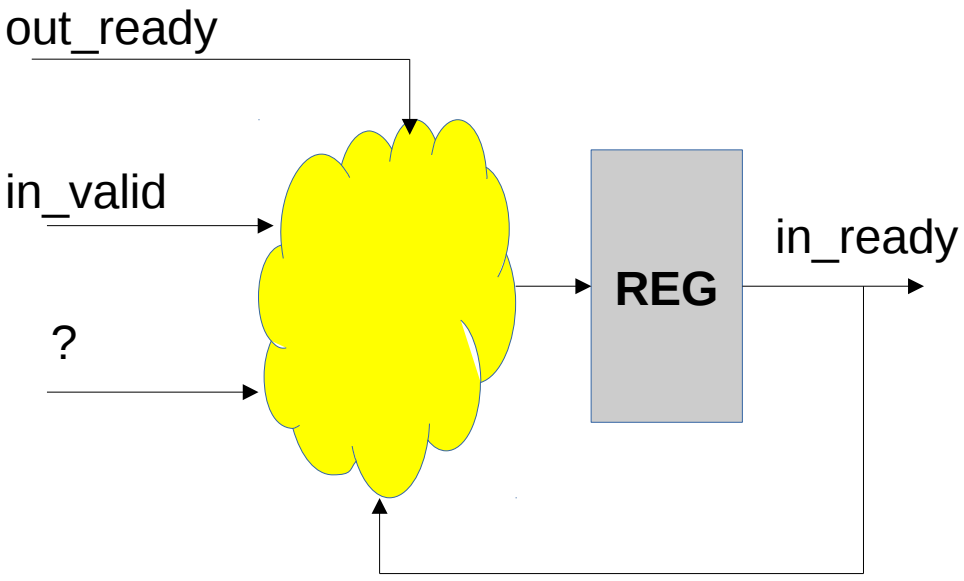
По каким то причинам приёмник не готов

Необходимо задержать передатчик

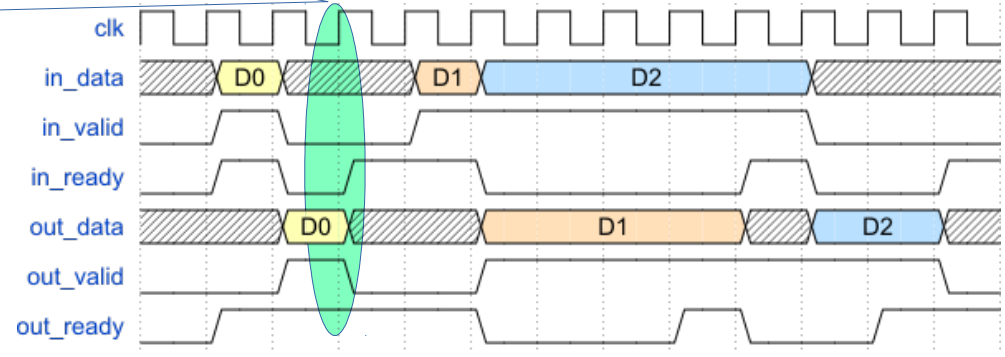


Формирование ready с паузой

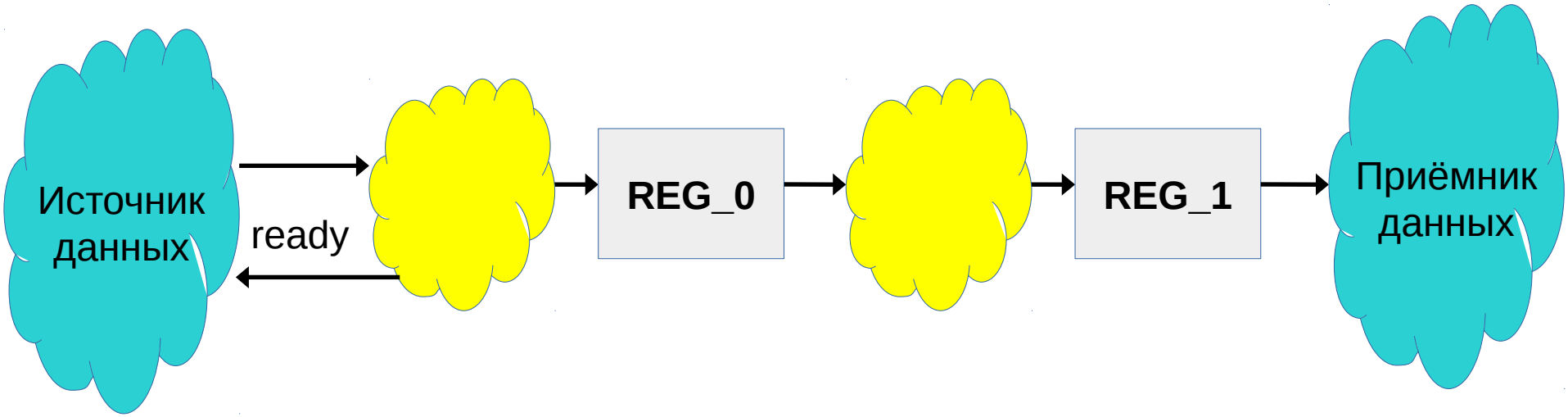




Обязательная пауза до момента захвата данных приёмником

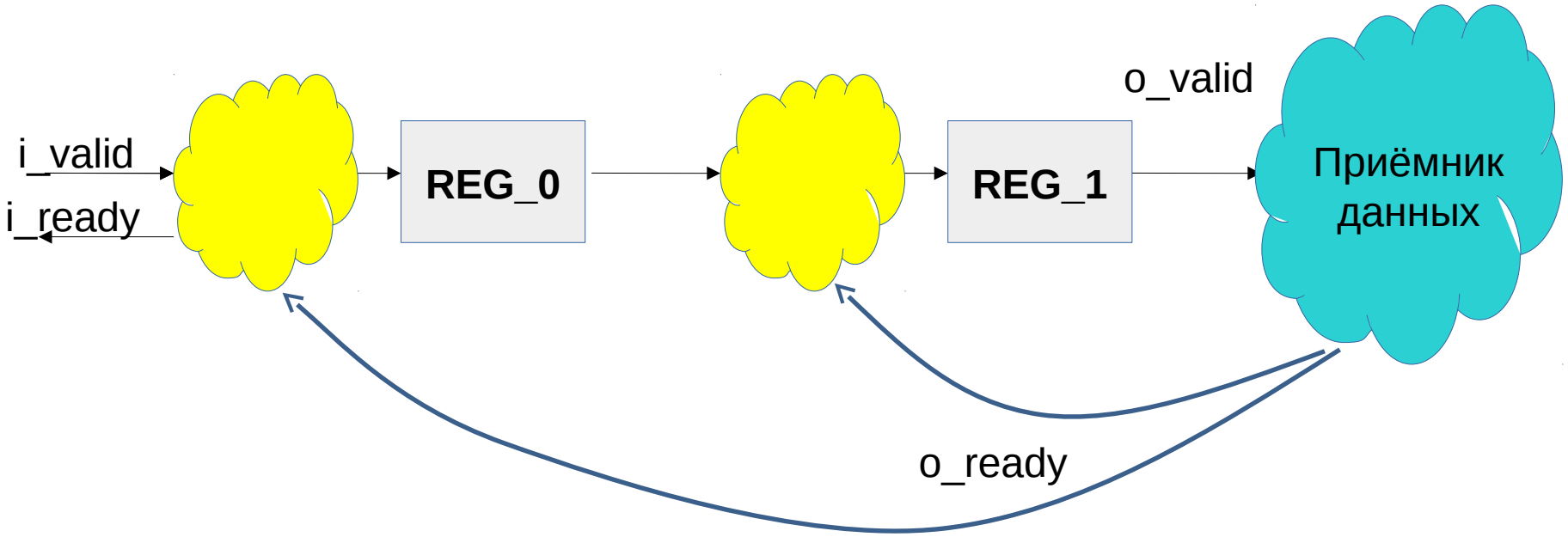


Типичный конвейер



Проблема — как сообщить источнику данных что приёмник готов или не готов к приёму данных

Вариант 1 — распространение o_ready

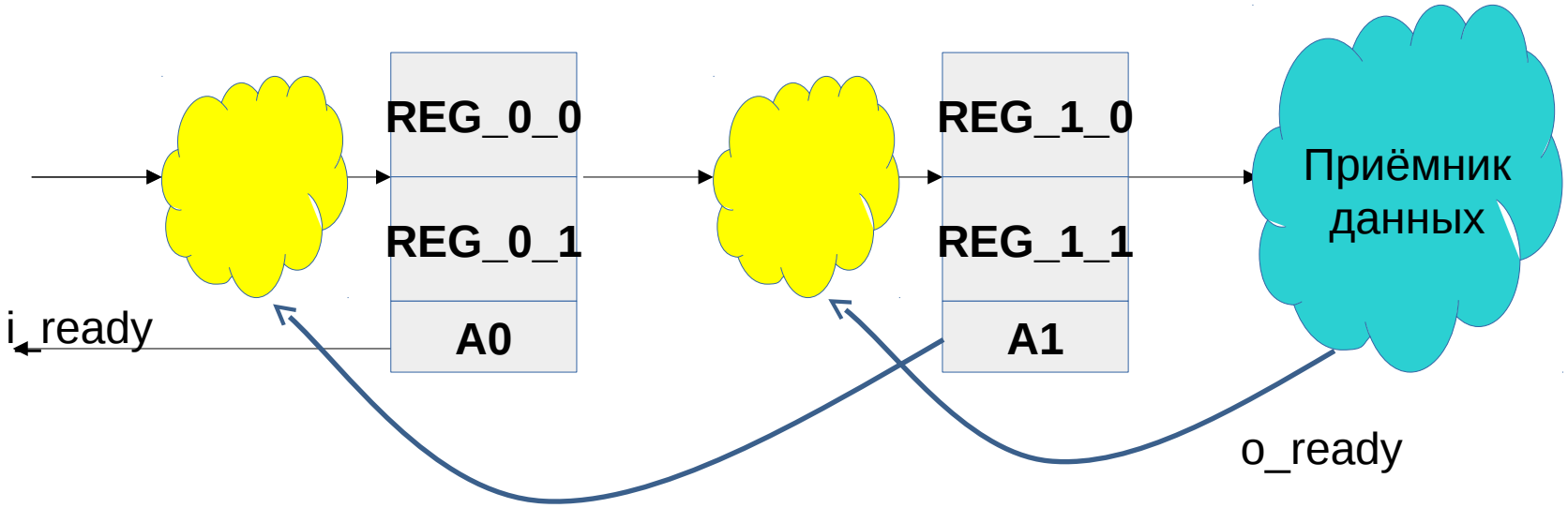


Как правило — приёмник формирует сигнал готовности к приёму данных

Сигнал **o_ready** поступает на все стадии конвейера.

Недостаток — проблемы с трассировкой

Вариант 2 — двойной буфер

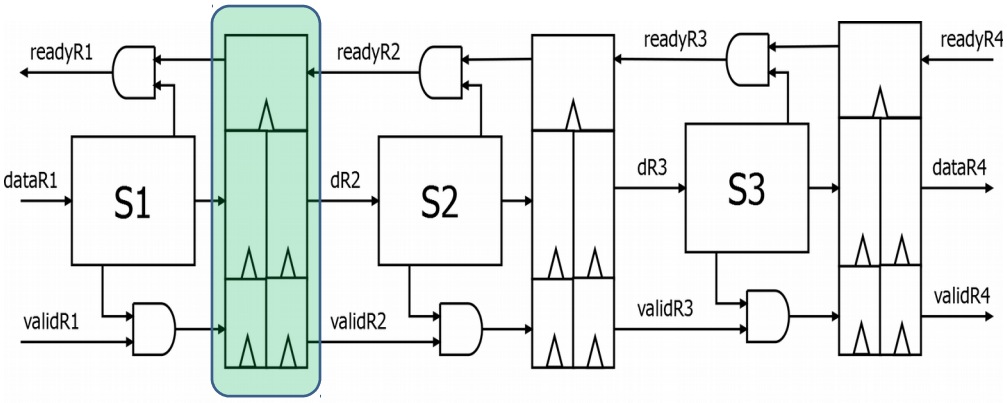


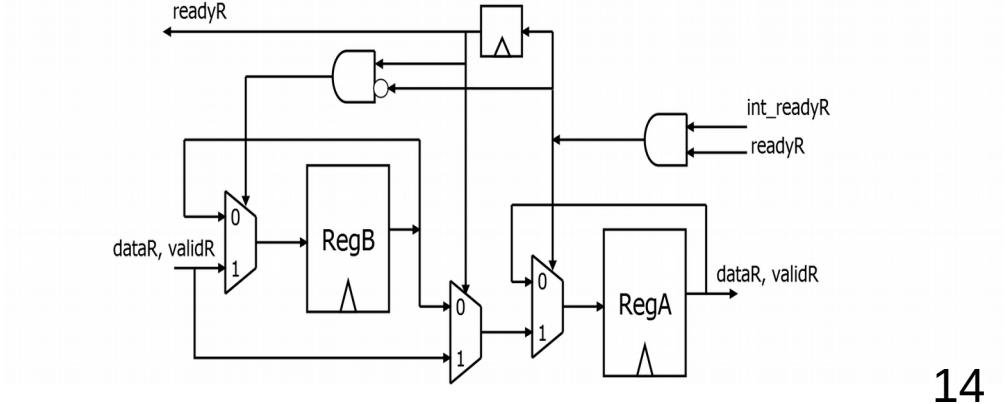
На каждой стадии конвейера используется двойной буфер.

Комбинационная схема анализирует только сигналы от соседних стадий конвейера.

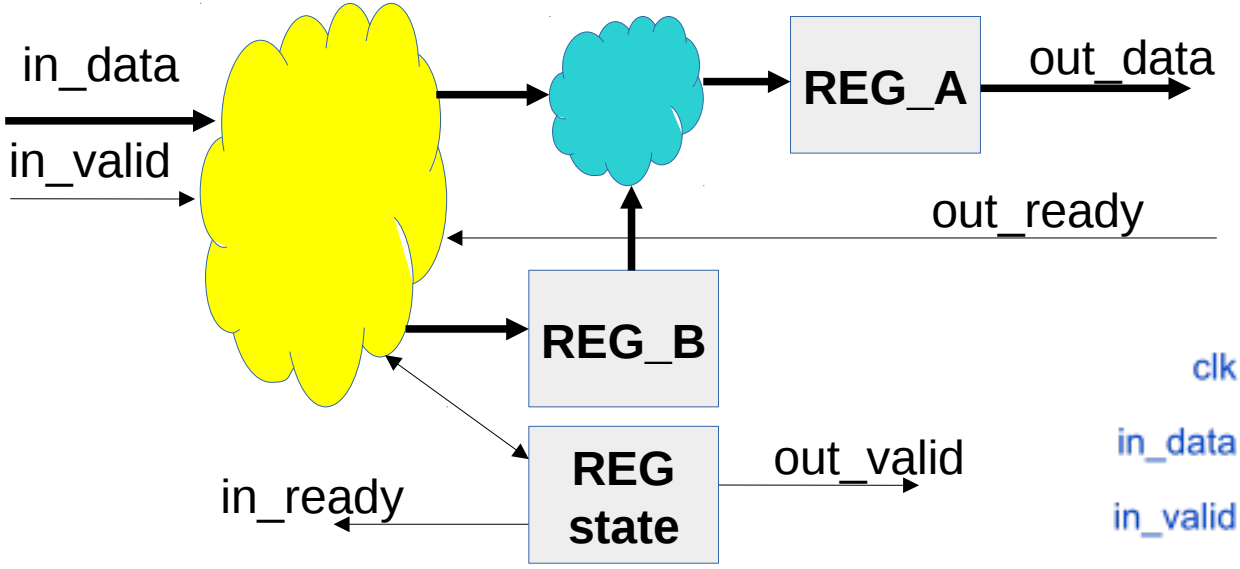
Недостаток — усложнение логики и дополнительные ресурсы

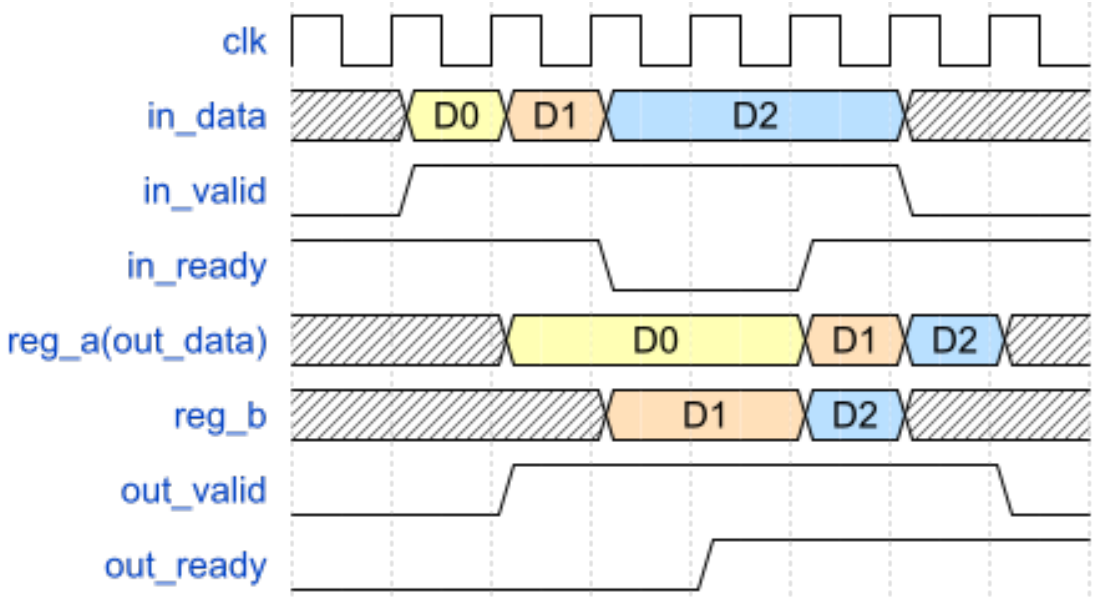
Классическая реализация





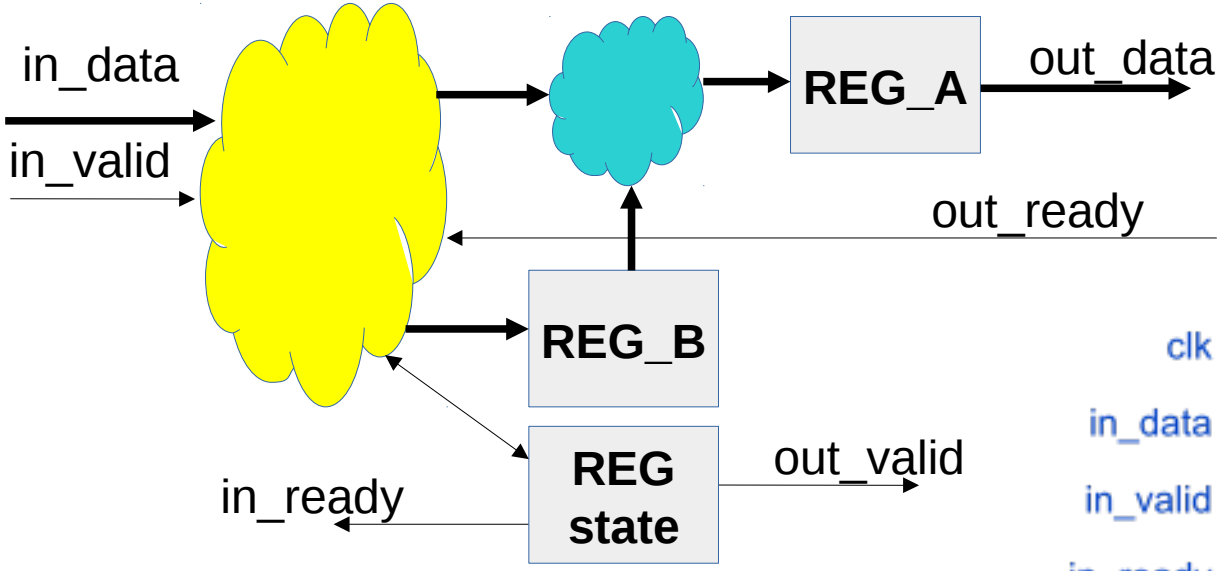
Цикл работы двойного буфера

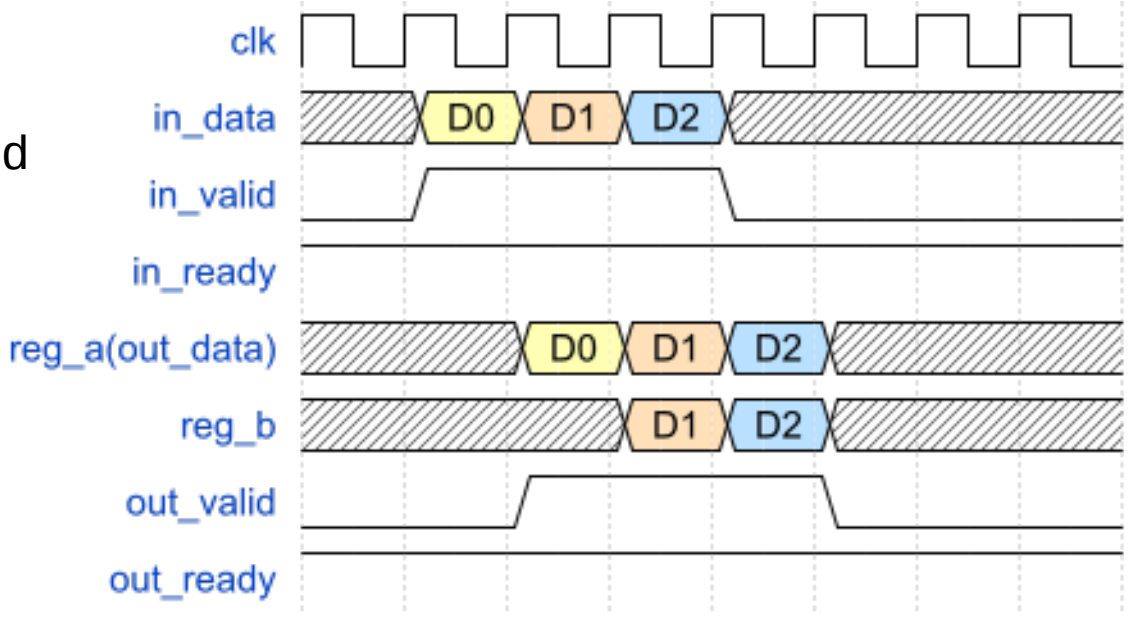




В момент приёма **D0** нет информации о том будет ли передача по шине **out_data**

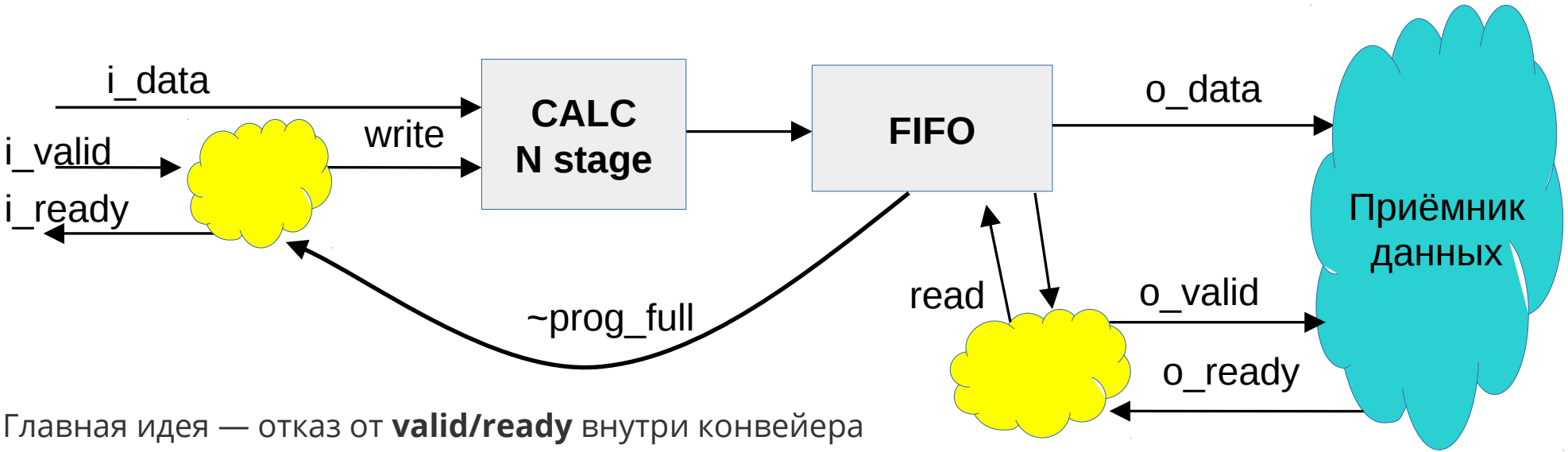
Цикл работы двойного буфера





Непрерывная передача при
постоянной готовности приёмника

Вариант 3 - FIFO



Главная идея — отказ от **valid/ready** внутри конвейера

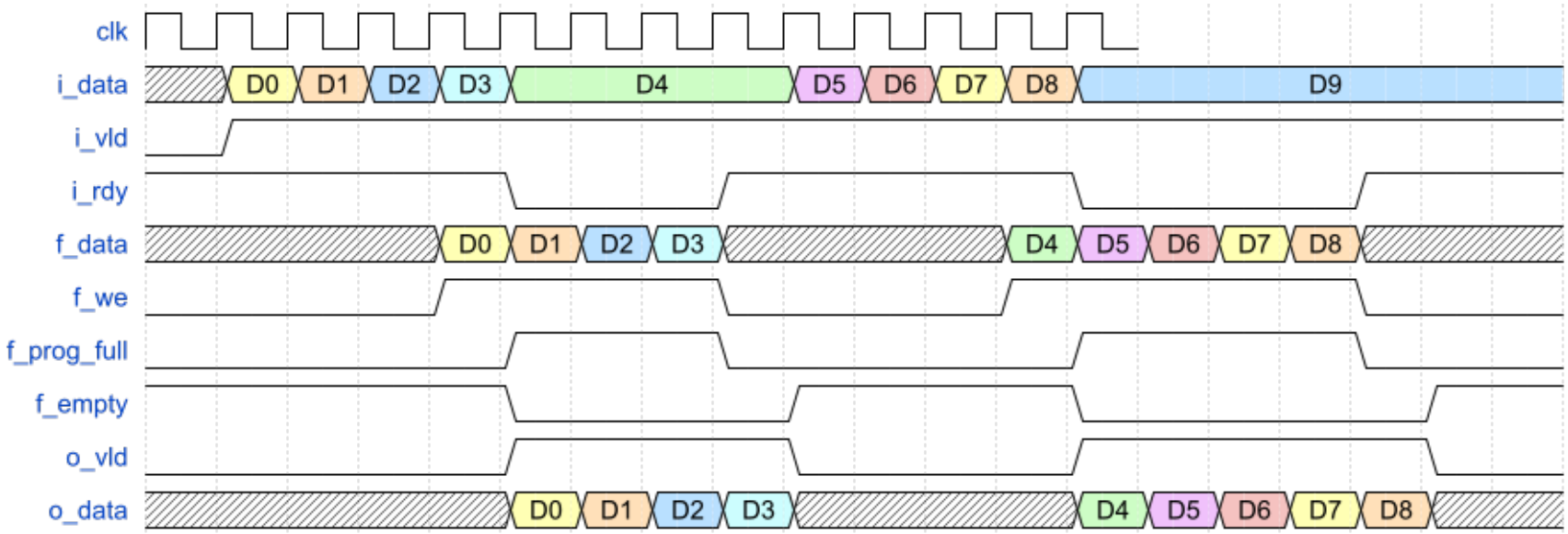
N — число стадий конвейера

Какой оптимальный размер FIFO ?

prog_full — флаг почти полного FIFO, как минимум равен N

github - пример **conv_with_fifo**

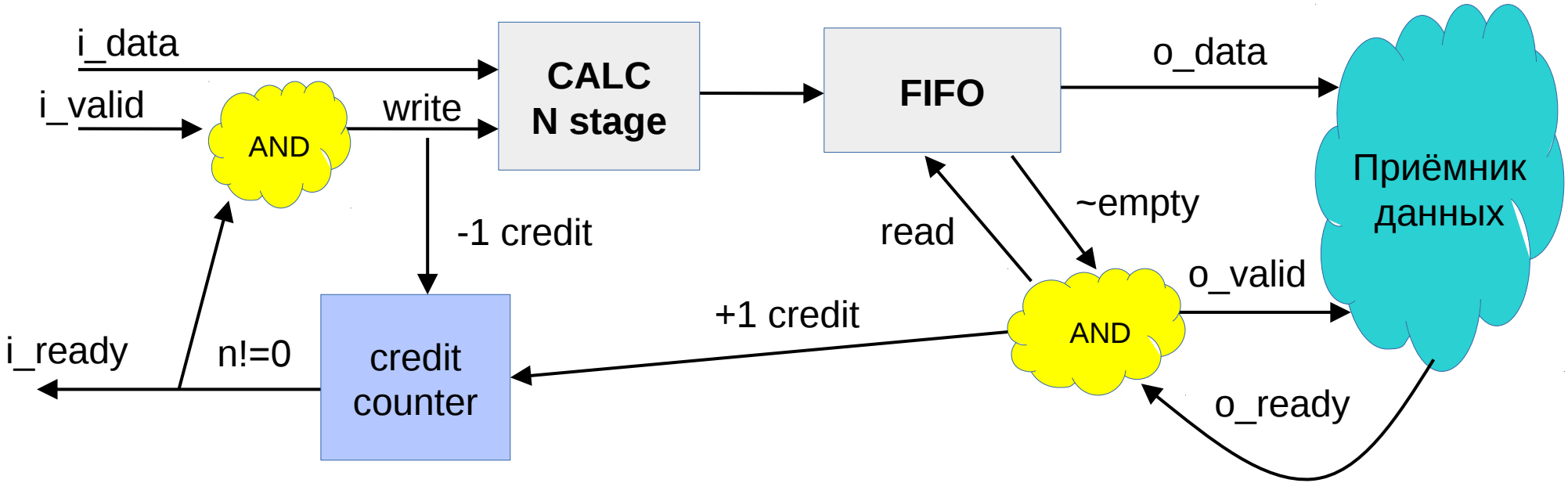
Размер FIFO 4 слова, конвейер 3 такта



При постоянной записи и при постоянном разрешении чтения возникают паузы при работе конвейера.

Оптимальный размер FIFO равен 2N

Вариант 4 — счётчик кредитов



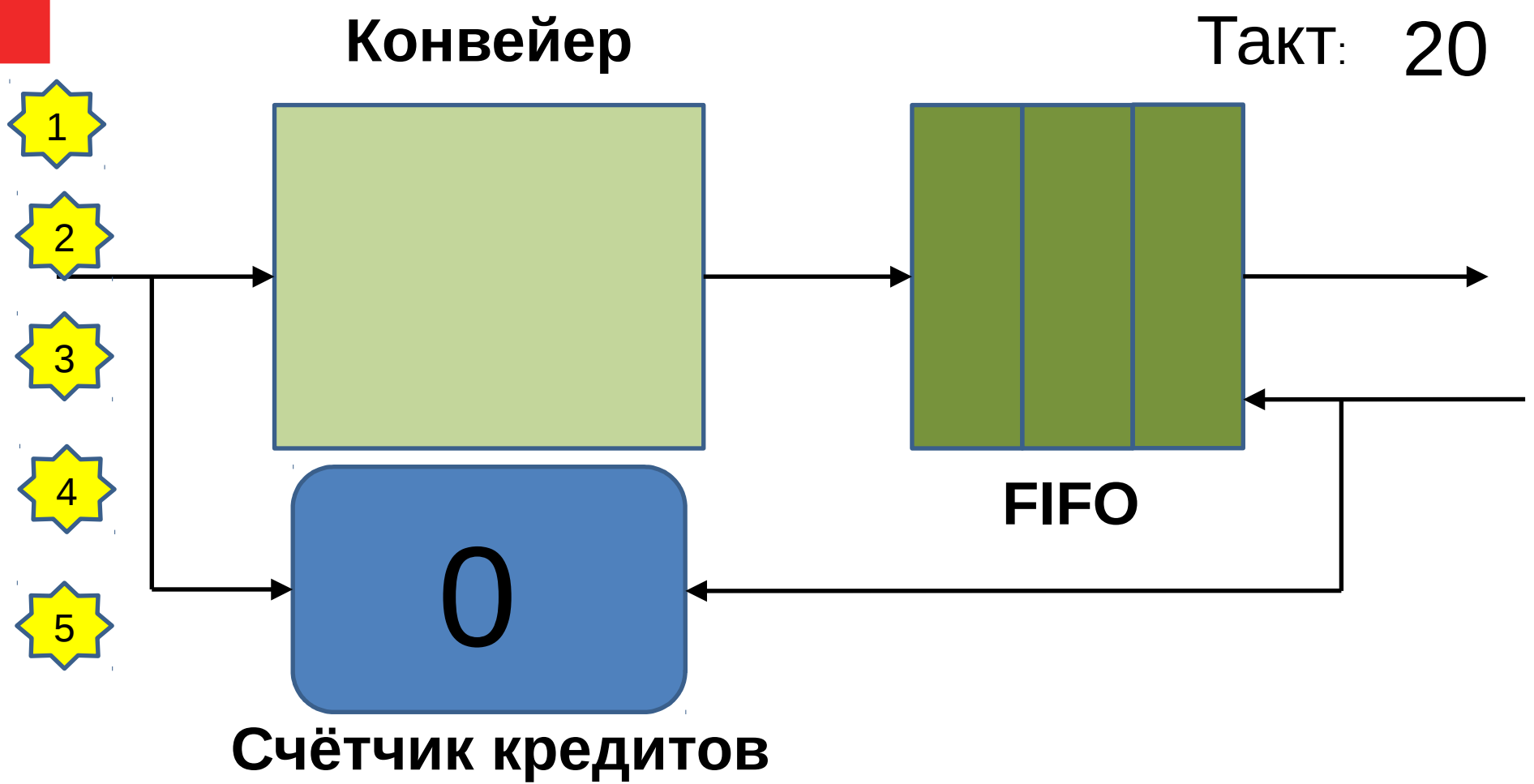
Начальное значение счётчика кредитов равно размеру FIFO

Оптимальный размер FIFO равен N+2

На вход узла CALC подаётся столько данных, сколько есть места в FIFO

github — пример **credit**

Работа счётчика кредитов




2023