Задача

Главной задачей данной работы является создание потокового кроссбара NxM. Данное устройство имеет N входных портов, и M выходных, его целью является обеспечение связи между любыми входными и выходными портами.

При разработке кроссбара было выделено три основных подмодуля

- Коммутационная сеть - отвечает непосредственно за соединение входных портов с выходными
- Планировщик - данный модуль создает управляющие сигналы для коммутационной сети и вычисляет выходные управляющие сигналы для кроссбара
- Round robin - реализует round robin арбитр для каждой выходной очереди

Конфигурационные параметры

- T_DATA_WIDTH - разрядность входных, выходных данных
- S_DATA_COUNT - количество master устройств
- M_DATA_COUNT - количество slave устройств
- T_ID_WIDTH log2(S_DATA_COUNT) - для обозначения номера master
- T_DEST_WIDTH log2(M_DATA_COUNT) - для обозначения номера slave

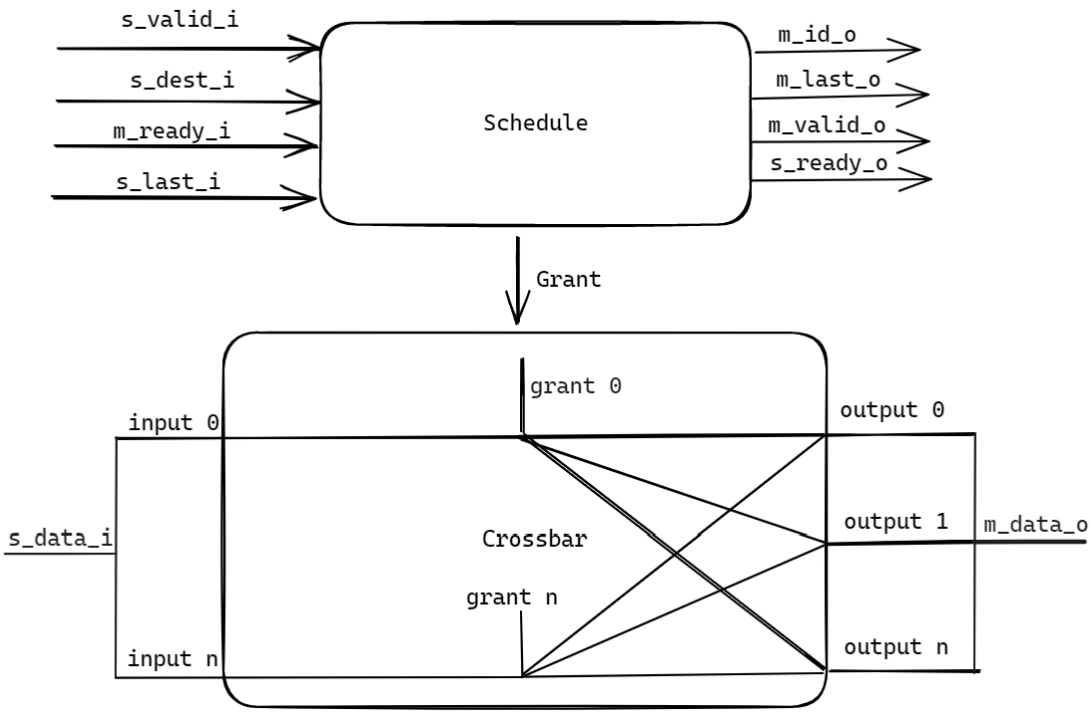
Порты ввода

- S_data_i [T_DATA_WIDTH] [S_DATA_COUNT] - матрица входных данных
- S_dest_i [T_DEST_WIDTH] [S_DATA_COUNT] - каждый мастер выбирает на какого slave передавать данные
- S_last_i [S_DATA_COUNT] - массив master, если стоит 1, то данный байт последний
- S_valid_i [S_DATA_COUNT] - массив master, если стоит 1, то запрос на передачу пакета
- M_ready_i [M_DATA_COUNT] - массив разрешения на передачи от slave, аналогично s_valid_i

Порты вывода

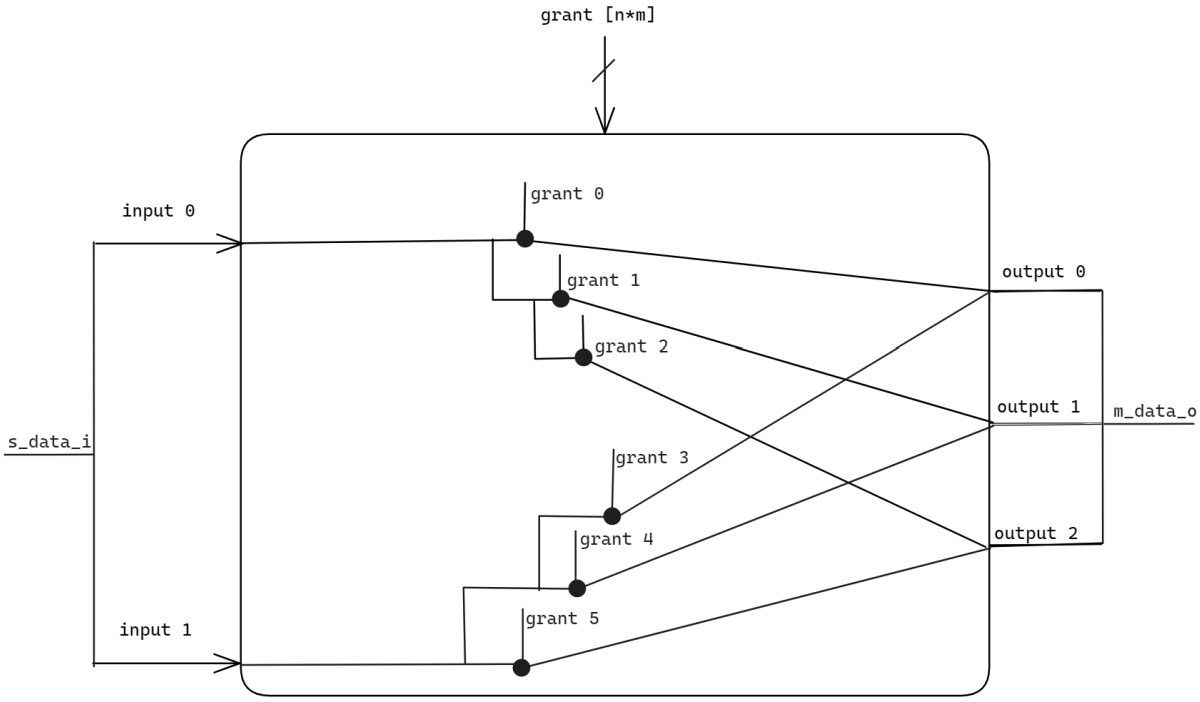
- M_data_o [T_DATA_WIDTH] [M_DATA_COUNT] - матрица выходных данных
- M_id_o [T_ID_WIDTH] [M_DATA_COUNT] - Сигнал для slave какой мастер передает информацию
- M_last_o [M_DATA_COUNT] - массив флагов последнего фрагмента от master
- M_valid_o [M_DATA_COUNT] - массив флагов, которые сообщают передаются ли сейчас данные по линии
- S_ready_o [S_DATA_COUNT] - s_valid_i

Общая схема кроссбара



Основные модули

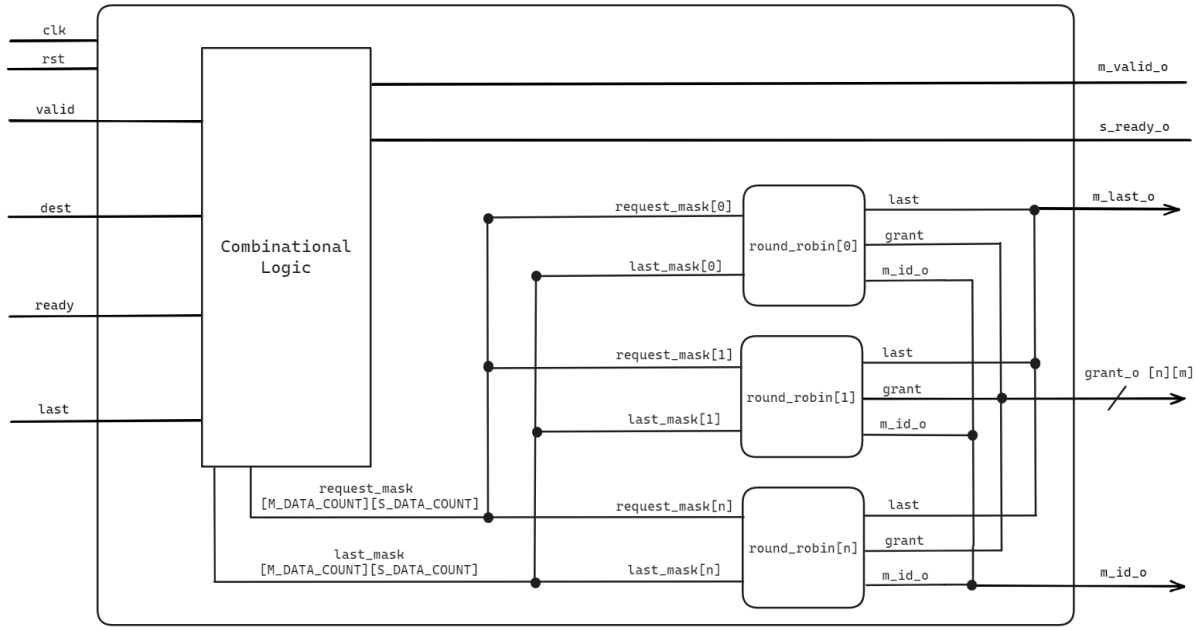
Коммутационная сеть



- Входные сигналы
  - S_data_i - Массив входных данных
  - Grant - массив управляющих сигналов
- Выходные сигналы
  - M_data_o - массив выходных данных

Коммутационная сеть реализует передачу данных с входных портов на выходные. Данная сеть связывает каждый входной порт с каждым выходным. Выбор на какой именно порт будут переданы данные определяется управляющими сигналами grant. Сигнал grant представляет собой шину, размером S_DATA_COUNT * M_DATA_COUNT, каждый бит шины подключен к своей точке соединения входного и выходного порта, если бит равен одному, то данные с входа передаются на выход, иначе передача блокируется. Данные сигналы гарантируют что на один выходной порт будут передаваться данные только с одного входного.

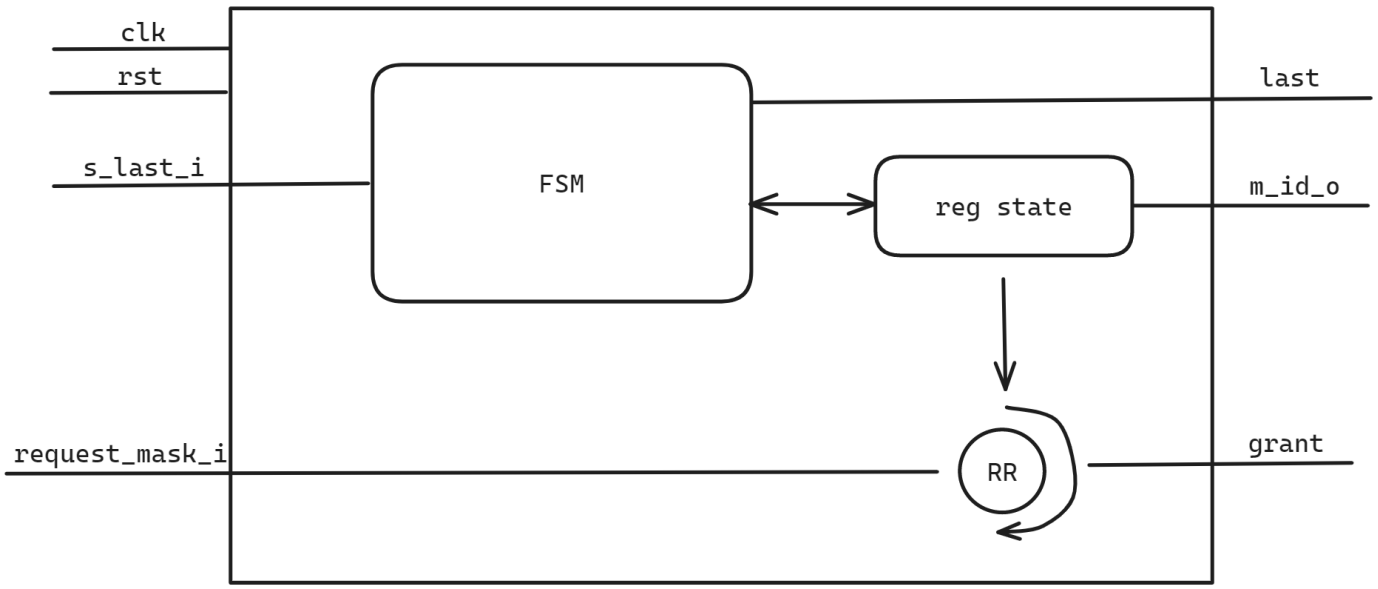
Планировщик



- Входные сигналы
    - S_valid_i - массив master, если стоит 1, то данные master посылает запрос на передачу пакета
    - S_dest_i - каждый мастер выбирает на какого slave передавать данные, выставляя его номер в двоичном виде
    - M_ready_i - массив разрешения на передачи от slave, если установлена единица, то данный slave готов к передаче данных.
    - S_last_i - массив master, если установлена единица, то это флаг завершения транзакции от i-го master
- Промежуточные значения
    - Request_mask - данный провод по сути является битовой маской для каждого выходного порта данных. Если i-й вход готов к передаче и порт, к которому он обращается также готов передавать данные, то устанавливается единица. Иначе устанавливается ноль.
    - Last_mask - данный провод так же является битовой маской для каждого выходного порта данных. Если на i-й позиции входного сигнала s_last_i установлена единица, то устанавливается единица. Иначе устанавливается ноль.
- Выходные сигналы
    - M_valid_o - массив slave, если на i-й позиции установлена единица, то это значит, что данный выходной порт готов к приему входных данных
    - S_ready_o - массив master, если на i-й позиции установлена единица, то это значит, что данный входной порт запрашивает передачу входных данных у выходного порта
    - M_last_o - массив slave, если на i-й позиции установлена единица, то это является символом конца передачи данных
    - Grant_o - матрица управляющих сигналов, которые подаются на коммутационную сеть
    - M_id_o - массив slave, на каждой позиции передаются номера входного порта, который передает данные на i-й выходной

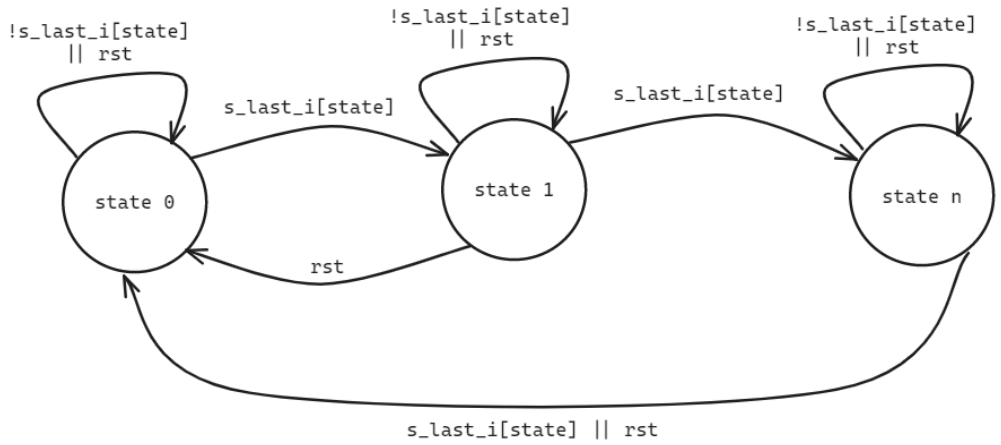
Данный модуль реализует основную логику работы кроссбара. Блок комбинационной логики на каждом такте формирует новые значения m_valid_o, s_ready_o, request_mask, last_mask. Сигналы request_mask, last_mask далее передаются на модули round robin, количество которых, равно количеству выходных портов. По окончании вычислений данные модули формируют выходную матрицу управляющих сигналов grant, а так же сигналы m_last_o и m_id_o.

Round-robin



- Входные сигналы
    - S_last_i - входная шина, которая показывает, какие входные порты могут передавать данные на данную выходной порт
    - Request_mask_i - входная шина, которая показывает, какие входные порты выставили флаг завершения передачи для данного выходного.
- Промежуточные значения
    - Reg state [T_ID___WIDTH] - регитр, который хранит состояние конечного автомата, значение регистра - номер входного порта, с которым установлена связь
- Выходные значения
    - M_last_o - пришел ли сигнал о завершении передачи данных с порта передачи
    - M_id_o - с какого порта передаются данные (его номер)
    - Grant - массив управляющих сигналов, если i-й входной порт передает данные, то на i-й позиции массива устанавливается единица, которая разрешает передачу данных на выходной порт

Данный модуль реализует логику арбитра round-robin. На каждом такте он будет вычислять текущее состояние автомата и вычислив его формировать выходные сигналы. При сбросе кроссбара (сигнал rst) регистр, который хранит текущее состояние конечного автомата, будет сброшен.



Переход из оного состояние в следующее осуществляется при условии, что закончена текущая транзакция (выставлен сигнал last), тогда автомат переходит в следующее состояние. Также текущее значение может приходить из вне, в случае, когда установлена связь с новым входным портом.

От действующего состояния будут перебираться входные порты так, что если первый встречный порт готов к передаче, то он захватит линию передачи и в сигнал state установится его номер, как только передача будет закончена, то к текущему состоянию прибавиться единица и перебор готовых портов пойдет с него. Тем самым обеспечивается приоритетный доступ (по индексу) без голодания.

Трюки и решения

- При разработке данного модуля пришлось чуть изменить интерфейс, а именно в verilog не поддерживается в качестве интерфейса массив сигналов, поэтому вместо двумерных массивов используются одномерные массивы того же размера.
- Модуль реализован по классической схеме - data path, control unit
- Сигнал rst инициализирует все регистры нулями