

Задание: разработать и верифицировать cross-bar, обеспечивающий коммутацию 2 master и 2 slave устройств.

Описание базового интерфейса:

- {master|slave}_{N}_req - однобитный сигнал, запрос на выполнение транзакции (master-> slave)
- {master|slave}_{N}_addr - 32-битный шина, содержит адрес запроса (master->slave)
- {master|slave}_{N}_cmd - однобитный сигнал - признак операции: 0 - read, 1 - write (master->slave)
- {master|slave}_{N}_wdata - 32-битный шина, содержит записываемые данные. передаются в том же такте, что и адрес (master-> slave)
- {master|slave}_{N}_ack - однобитный сигнал-подтверждение, что в данном такте slave принял запрос к исполнению (slave->master). при этом, slave должен зафиксировать _addr, _cmd, и _wdata (в случае транзакции записи). перевод _ack в активное состояние разрешает master устройству снять запрос в следующем такте
- {master|slave}_{N}_rdata - 32-битная шина, содержит считываемые данные. передаются на следующем такте после подтверждения транзакции(_ack = 1)(slave-> master)

Выбор slave устройства определяется старшим битом адреса. Арбитраж между несколькими master запросами в одно slave устройство осуществляется по дисциплине "round-robin".

Необходимо разработать и верифицировать базовый модуль, а также оптимизировать его по частоте. По желанию (опционально) можно выполнить одно или более расширений задания из списка внизу.

В результате выполнения ожидается код на Verilog или System Verilog, test bench и набор тестовых векторов для среды ModelSim, отчеты о синтезе для произвольного FPGA device из поддерживаемых Quartus lite (<http://dl.altera.com/?edition=lite>).

Результаты выполнения должны быть размещены в репозитории на github.com и передаваться в виде ссылки на репо. Вопросы по заданию можно задавать по email: interns.rtl@syntacore.com. Рекомендуем также написать перед началом выполнения задания, а также сообщить примерные сроки ожидания результатов.

Возможные расширения базового задания (опционально - доп. преимущество. Можно выполнить любое количество):

1. распространить случай на 4 master и 4 slave порта
2. {master|slave}_{N}_resp - однобитный сигнал (slave-> master), подтверждение считываемых данных _rdata. Соответственно _rdata возвращаются не в следующем такте, а в том такте, когда активен данный сигнал
3. slave может буферизировать до 4 запросов на чтение, до выставления rdata и resp (выставлять задержку на данные)
4. slave может возвращать данные не в той последовательности, в которой были приняты запросы, соответственно в cross-bar должна быть предусмотрена логика для буферизации и сиквенсирования считываемых данных, - таким образом, чтобы в master данные возвращались в той последовательности, в которой были подтверждены запросы