

Проект

2 июня 2025 г.

Проект содержит 2 модуля. Один из них контролирует доступ к одному slave-устройству, второй создаёт нужное количество модулей первого типа и контролирует выход для master-устройств.

round_robin.sv

Этот модуль управляет доступом к slave-устройству с политикой Round-Robin. Фиксируется по одному экземпляру для каждого slave-устройства, поэтому удобно здесь же управлять выходами для этого устройства. Интерфейс модуля:

```
module round_robin #(
    parameter  T_DATA_WIDTH = 8,
               S_DATA_COUNT = 2,
               M_DATA_COUNT = 3,
               T_ID___WIDTH = $clog2(S_DATA_COUNT),
               T_DEST_WIDTH = $clog2(M_DATA_COUNT)
) (
    input  logic                clk,
    input  logic                rst_n,

    input  logic [T_DEST_WIDTH - 1 : 0] number,

    input  logic [T_DATA_WIDTH - 1 : 0] s_data_i [S_DATA_COUNT - 1 : 0],
    input  logic [T_DEST_WIDTH - 1 : 0] s_dest_i [S_DATA_COUNT - 1 : 0],
    input  logic [S_DATA_COUNT - 1 : 0] s_last_i,
    input  logic [S_DATA_COUNT - 1 : 0] s_valid_i,

    input  logic                m_ready_i,

    output logic                m_valid_o,
    output logic                m_last_o,

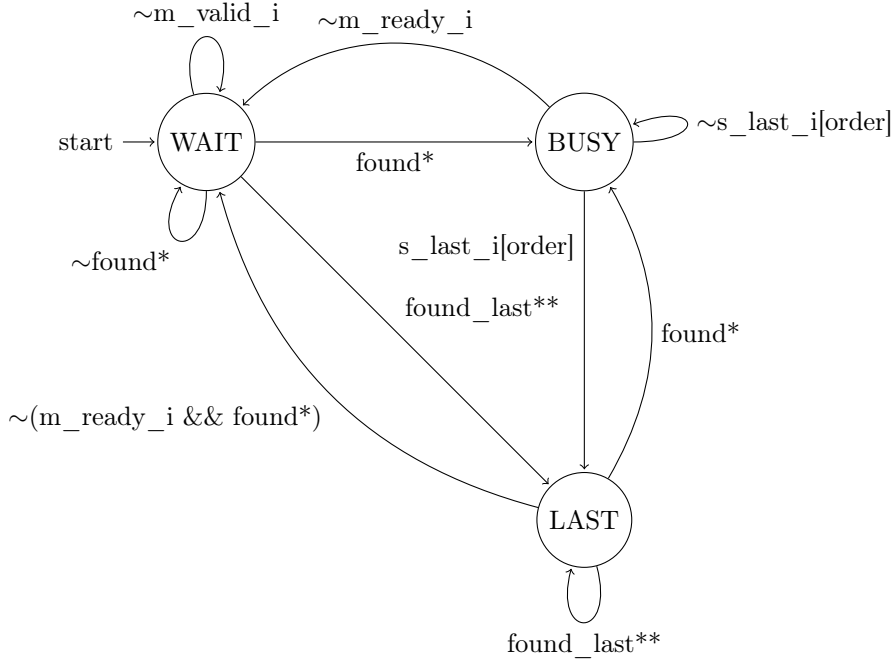
    output logic [T_DATA_WIDTH - 1 : 0] m_data_o,
    output logic [T_ID___WIDTH - 1 : 0] m_id_o,

    output logic [T_ID___WIDTH - 1 : 0] s_ready_id
);
```

У модуля есть дополнительные порты: number и s_ready_id. Значение входа number устанавливается при создании экземпляра модуля и не меняется, по нему модуль находит запросы, относящиеся к нему. Выход s_ready_id возвращает id master-устройства, с которого в данный момент принимается пакет.

Модуль работает как конечный автомат с 3 состояниями: WAIT, BUSY, LAST. WAIT отвечает за ситуацию, при которой соответствующее slave-устройство не готово принимать пакет или ни одно master-устройство не запрашивает передачи. В состоянии WAIT пакеты пересылаются, пока не придёт последний, а в состоянии LAST обрабатывается последний пакет.

Внутренний регистр order отвечает за определение очереди входов. Его значение - вход, который передал последний пакет.



$*found = \exists j(s_valid_i[j] \ \&\& \ s_dest_i[j] == number)$

$**sound_last = \exists j(s_valid_i[j] \ \&\& \ s_dest_i[j] == number \ \&\& \ s_last_i[j])$

В состояниях приёма пакетов (BUSY, LAST) устанавливаются значения выходов m_valid_o , m_last_o , m_data_o , m_id_o . Регистр $order$ устанавливается при переходе в состояния приёма равным значению текущего номера master-устройства, а при переходе из LAST в WAIT увеличивается на 1.

stream_xbar.sv

Основной модуль устройства. В нём создаётся M_DATA_COUNT модулей типа `round_robin`, им сразу назначаются номера.

Для каждого master-устройства задаётся комбинационная формула для $s_ready_o[i]$. Этот выход равен 1, когда master-устройство готово к передаче, целевое slave-устройство готово и принимает от соответствующего номера:

```

logic dest;
logic match;
assign dest = s_dest_i[j];
assign match = s_valid_i[j] && (s_ready_id[dest] == j) && m_ready_i[dest];
assign s_ready_o[j] = match;

```