```
module NoC (
  input clock,
  input [3:0] src_id,
  input [3:0] dest_id,
  input [7:0] data_in,
  output [7:0] data_out,
  output busy
);
  // Memory array
  reg [7:0] mem [0:15][0:15];
  // Output valid flag
  reg valid;
  // FIFO pointers
  reg [3:0] rd_ptr, wr_ptr;
  // FIFO depth
  parameter FIFO_DEPTH = 16;
  // FIFO storage
  reg [7:0] fifo [0:FIFO_DEPTH-1];
  // Write FIFO
  always @(posedge clock) begin
    if (wr_ptr != rd_ptr) begin
      fifo[wr_ptr] <= data_in;
      wr_ptr <= wr_ptr + 1;
      if (wr_ptr == FIFO_DEPTH) wr_ptr <= 0;
    end
  end
  // Read FIFO
  always @(posedge clock) begin
    if (rd_ptr != wr_ptr) begin
      mem[dest_id][src_id] <= fifo[rd_ptr];
      rd_ptr <= rd_ptr + 1;
      if (rd_ptr == FIFO_DEPTH) rd_ptr <= 0;
    end
  end
  // Read memory
  always @(posedge clock) begin
    if (valid) data_out <= mem[src_id][dest_id];
  end
  // Set valid flag
  always @(posedge clock) begin
    valid <= (rd_ptr != wr_ptr);
  end
  // Set busy output
  assign busy = (rd_ptr == wr_ptr);
```

```
endmodule
```