```
module synchronous_ram #(
    parameter ADDR_WIDTH = 4,  // Number of address bits
    parameter DATA_WIDTH = 8  // Width of data
) (
    input clk,                // Clock signal
    input we,                 // Write enable (1 = write, 0 = read)
    input [ADDR_WIDTH-1:0] addr, // Address for read/write
    input [DATA_WIDTH-1:0] din,  // Data input for write operation
    output reg [DATA_WIDTH-1:0] dout // Data output for read operation
);
    // RAM storage
    reg [DATA_WIDTH-1:0] ram [2**ADDR_WIDTH-1:0];

    // Synchronous process for read and write
```

```
    always @(posedge clk) begin
        if (we) begin
            ram[addr] <= din;    // Write data to
RAM
        end else begin
            dout <= ram[addr];   // Read data
from RAM
        end
    end
endmodule
```