```
module serial_adder(
    input [7:0] A, B,
    input reset,
    input clock,
    output wire [7:0] sum
);
    reg [3:0] count;
    reg carry_in;
    reg s;
    reg carry_in_next;
    wire [7:0] qa, qb;
    wire run;
    shiftreg A1(A, reset, 1'b1, 1'b0, clock, qa);
    shiftreg B1(B, reset, 1'b1, 1'b0, clock, qb);
    shiftreg SUM1(8'b0, reset, run, s, clock, sum);
    always @(posedge clock or posedge reset) begin
        if (reset) begin
            carry_in <= 0;
            count <= 8;
        end else if (run) begin
            carry_in <= carry_in_next;
        end
    end
    always @(qa, qb, carry_in) begin
        s = qa[0] ^ qb[0] ^ carry_in;
        carry_in_next = (qa[0] & qb[0]) | (qa[0] & carry_in) | (qb[0] & carry_in);
    end
    always @(posedge clock or posedge reset) begin
        if (reset)
            count <= 8;
        else if (run)
```

```
        count <= count - 1;
    end
    assign run = (count > 0);
endmodule
```