```
module slow_division (
  input wire clk,
  input wire reset,
  input wire [7:0] dividend, // 8-bit dividend
  input wire [3:0] divisor,  // 4-bit divisor
  output reg [7:0] quotient, // 8-bit quotient
  output reg [3:0] remainder // 4-bit remainder
);

  reg [7:0] temp_dividend;
  reg [3:0] temp_divisor;
  reg [3:0] count;
  reg subtract_flag;

  always @(posedge clk or posedge reset) begin
    if (reset) begin
      temp_dividend <= 0;
      temp_divisor <= 0;
      count <= 0;
      quotient <= 0;
      remainder <= 0;
      subtract_flag <= 0;
    end
    else begin
      temp_dividend <= dividend;
      temp_divisor <= divisor;

      if (count < 8) begin
        if (subtract_flag) begin
          if (temp_dividend >= temp_divisor) begin
            temp_dividend <= temp_dividend - temp_divisor;
            quotient[count] <= 1;
          end
          subtract_flag <= 0;
        end
        else begin
          temp_dividend <= temp_dividend << 1;
          subtract_flag <= 1;
          count <= count + 1;
        end
      end
      else begin
        remainder <= temp_dividend;
      end
    end
  end

endmodule
```