Problem 1
Problem 2
Problem 3

module ff_rs_en_oe(input clk,d,res,en,oe,
                   output out);

    reg q;

    always @(posedge clk, posedge res) begin
        if (reset) q <= 1'b0;
        else if (en) q <= d;
    end

    assign out = oe ? q : 1'bz;
endmodule

Problem 4

a.) Yes. In this circuit, one drop will come out of output $B$ for every 16 drops entering input $A$. Only four ‘flip-flops’ are required; the entire tree need not be filled (which would require 15 units).

b.) No, this is not possible. Each output of the ‘flip-flop’ simply outputs a drop for every other input drop.

Without combinational logic, it is not possible to output drops at a frequency of $\frac{1}{3}$ the input frequency.

Problem 5

a.) See SARTestbench.v

b.) Block diagram:
c.) The state of the entire SAR is a 4-bit counter that counts down from 9 to zero, with ‘done’ being asserted in state 9 (the first cycle of converting the next sample), during which the preceding conversion is read out. The ‘sample’ signal is asserted in state zero to prepare the value for the next conversion.
The behavior of each bit of the guess data to the ADC can then be described as an individual sub-FSM that causes the bit to turn on or off based on the value of the counter:

Now, the only remaining complication to resolve is latching the decided output on the transition between count 0 and count 9; this is because only ten cycles are allowed per sample, not 11. Therefore, the resulting output after the decision based on ‘tb’ to either turn off or leave on bit 0 must be stored in a separate register, as the ADC guess must immediately be reset to ‘10’b1000000000’ for the next sample.
d.) See SAR.v

e.) See waveform output – displayed across 2 PDF documents, waveform_1.pdf and waveform_2.pdf