(0)

Ch 17 HW

9

```
module ring_counter(
    input clk,
    input control,
    output [3:0] count);
always_ff @(posedge clk) begin
    if(control)
        count <= 4'b1000;
    else if(!control)
        count[3] <= !control;
        count[2] <= count[3];
        count[1] <= count[2];
        count[0] <= count[1];
        count[3] <= count[0];
    end
endmodule
```

b
```
module averaging(
  input clr, clk,
  input [23:0] sample,
  output [23:0] average);
  wire logic [4:0] q, q';
  sum[27:0], start, exsample[27:0];
  always_ff @(posedge clk) begin
  if(clr)
    q <= 4'b0000;
  else if(!clr)
    q <= q + 1;
  end
  start <= q == 4'b0000;
  exsample[27:24] <= sum[23];
  always_ff @(posedge) begin
  exsample[23:0] <= sum;
  sum <= sum + exsample;
  if(start)
     q' <= exsample;
  else if(!start)
     q' <= sum;
  average <= q' / 16;
  end
  end module
```