Apartado 2

MUX21

```
module mux21 (input a,b, s,output z);
  assign z = ~s&a | s&b;
endmodule
```

Apartado 3

MUX41

```
module mux21 (input a,b, s,output z);
  assign z = ~s&a | s&b;
endmodule

module mux41 (input [1:0] s, input [3:0] d, output z);
  wire [1:0] net;

  mux21 mux1 (.a(d[0]),.b(d[1]),.s(s[0]),.z(net[0]));
  mux21 mux2 (.a(d[2]),.b(d[3]),.s(s[0]),.z(net[1]));
  mux21 mux3 (.a(net[0]),.b(net[1]),.s(s[1]),.z(z));

endmodule
```

Apartado 4

CONTADOR

```
module contador(
input clk,
input ld,
input en,
  input [3:0] x,
  output [3:0] z);

reg [3:0] q;

  always @(posedge clk)
    if (ld == 1)
       q <= x;
  else if (en)
       q <= q + 1;

  assign z = q;

endmodule
```

Apartado 5

TEST DEL CONTADOR

```
module contador_tb;
  wire [3:0] z;
  reg en;
  reg clk;
  reg ld;
  reg [3:0] x;

  contador uut (.clk(clk),.ld(ld),.z(z),.x(x),.en(en));

initial begin
  $dumpvars(1,contador_tb);
    clk=0; ld=1; en=0; x=4'hc;
  @(negedge clk)
  x=4'h3;
  @(negedge clk)
  x=4'hd;
  en=1;
  @(negedge clk)
  ld=0;
  x=4'h9;
  repeat (5) @(negedge clk);
  en=0;
  @(negedge clk)
    $finish;
  end
  always begin
    #20
    clk = ~clk;
    $display ("clk=%b, en=%b,  ld=%b, x=%b, z=%b", clk,en,ld,x,z);
  end
endmodule
```