Pauta tarea 4

Contador

```
module maquina1 (input clk,
                 input reset,
                 input E, C,
                 output  [6:0] SEG );

reg [2:0] state, nextstate;

parameter S1 = 3'b001;
parameter S2 = 3'b010;
parameter S3 = 3'b011;
parameter S4 = 3'b100;
parameter S5 = 3'b101;
parameter S6 = 3'b110;

always @ (posedge clk, posedge reset)
        if (reset) state <= S1;
        else       state <= nextstate;

always @ (*)
        case (state)
        S1: if((E==1)&(C==1)) nextstate = S2;
            else nextstate = S1;
        S2: if((E==1)&(C==1)) nextstate = S3;
            else nextstate = S2;
        S3: if((E==1)&(C==1)) nextstate = S4;
            else nextstate = S3;
        S4: if((E==1)&(C==1)) nextstate = S5;
            else nextstate = S4;
        S5: if((E==1)&(C==1)) nextstate = S6;
            else nextstate = S5;
        S6: if((E==1)&(C==1)) nextstate = S1;
            else nextstate = S6;
        default: nextstate = S1;
        endcase


// decodificador binario a 7 segmentos

assign SEG[6] = (state==2) | (state==3) | (state==5) | (state==6);  // segmento a
```

```
assign SEG [5] = (state==1) | (state==2) | (state==3) | (state==4);  // segmento b
assign SEG [4] = (state==1) | (state==3) | (state==4) | (state==5) | (state==6);  // segmento c
assign SEG [3] = S[6];  // segmento d
assign SEG [2] = (state==2) | (state==6);  // segmento e
assign SEG [1] = (state==4) | (state==5) | (state==6);  // segmento f
assign SEG [0] = (state==2) | (state==3) | (state==4) | (state==5) | (state==6);  // segmento g
```

Start / Stop

```
module maquina2 (input clk,
                 input reset,
                 input A,
                 output E);

reg state, nextstate;

parameter S0 = 1'b0;
parameter S1 = 1'b1;

always @ (posedge clk, posedge reset)
        if (reset) state <= S0;
        else       state <= nextstate;

always @ (*)
        case (state)
        S0: if (A) nextstate = S1;
            else nextstate = S0;
        S1: if (A) nextstate = S0;
            else nextstate = S1;
        endcase

assign E = (state == 0) & (A==1) | (state == 1) & (A==0);
```

Divisor clock (2x)

```
module maquina3 (input clk,
                 input reset,
                 input S,
                 output C);

reg state, nextstate;

parameter S0 = 1'b0;
parameter S1 = 1'b1;

always @ (posedge clk, posedge reset)
        if (reset) state <= S0;
        else       state <= nextstate;

always @ (*)
case (state)
        S0: if (S) nextstate = S0;
            else nextstate = S1;
        S1: nextstate = S0;
        endcase

assign C = (state == 0) & (S == 1) | (state == 1);
```