



INSTITUTO POLITECNICO
NACIONAL



ESCUELA SUPERIOR DE COMPUTO

Práctica 6

“Motor Bipolar con puente H”

Diseño de Sistemas Digitales

Rene Baltazar Jiménez Ruiz

Equipo 7

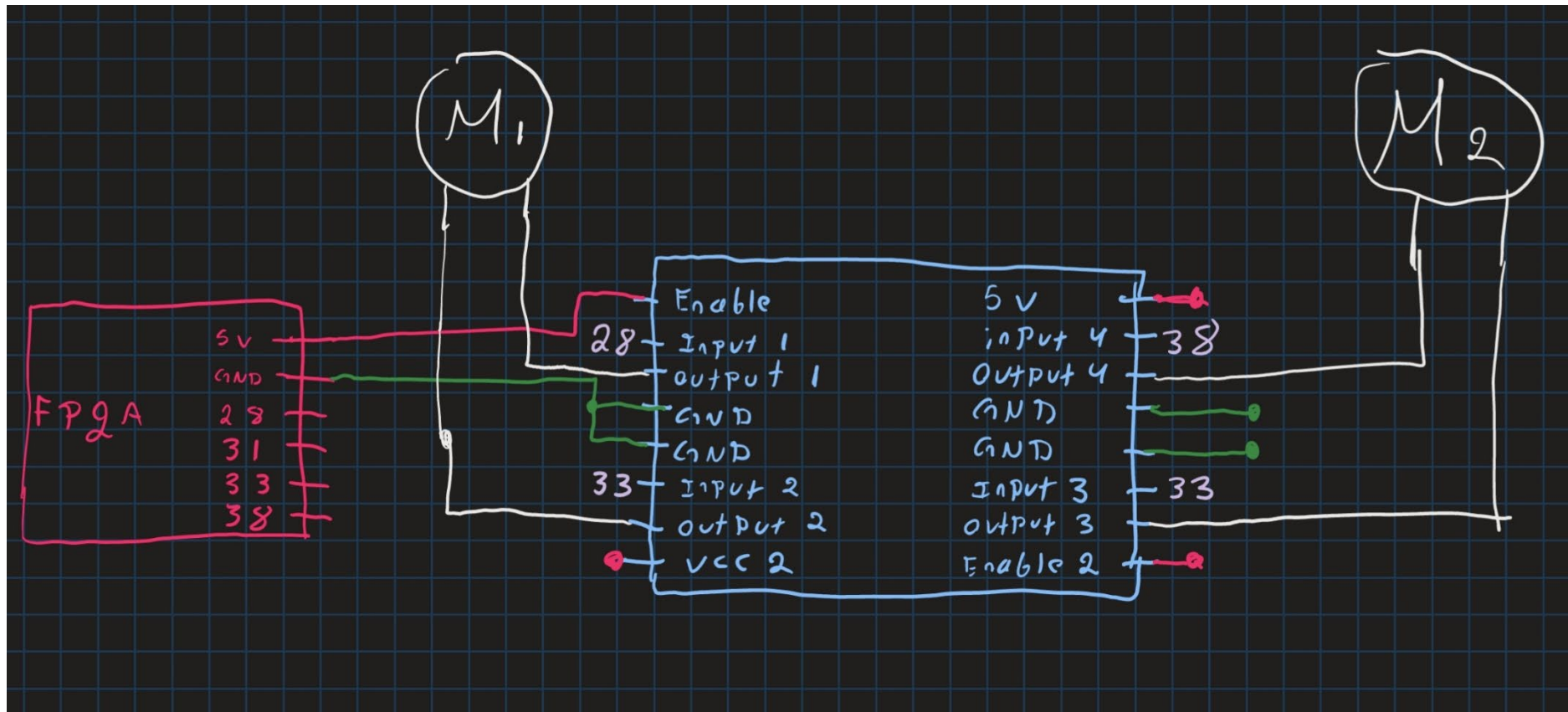
Hernández Hernández Roberto Issac

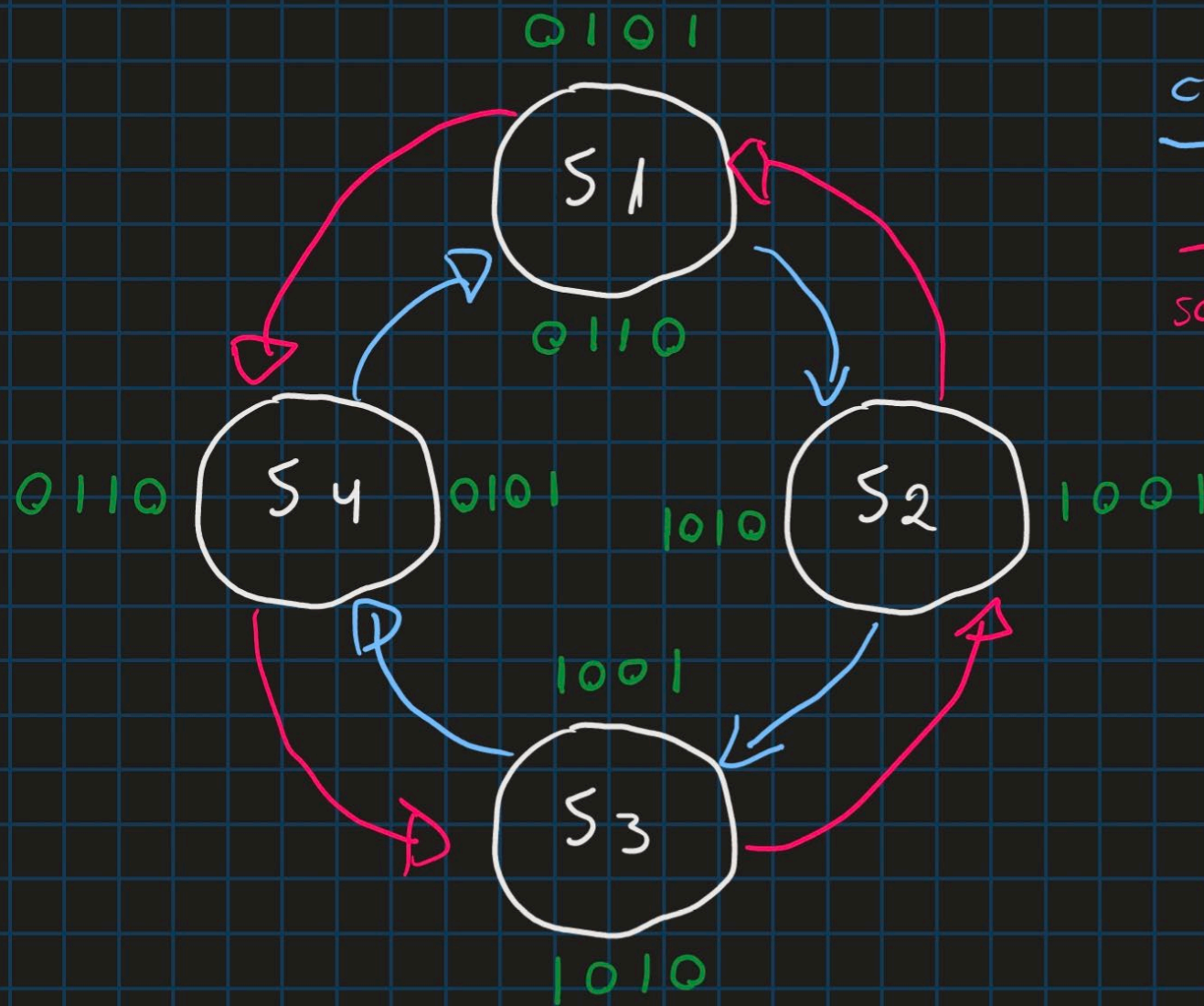
Navarrete Piña Arath Antonio

Yescas García Ulysses Aarón

4CV3

DIAGRAMA





contra reloj
— OPC = 1

— OPC = 0
sentido reloj

CODIGO VHDL

```

1  --Control completo del motor a pasos a 4 fases
2  --
3  --  Descripcion
4  --  CLK   Entrada    1      H      Reloj maestro
5  --  OPC   Entrada    3      H      Control de movimientos
6  --                                     1 sentido reloj
7  --                                     --0 contra reloj
8  --  P     Salida     4      H      Secuencia de fases entradas al punete h
9
10 library IEEE;
11 use IEEE.std_logic_1164.all;
12
13 entity MotorBiPasos is
14 port(
15     CLK   : in std_logic;
16     OPC   : in std_logic;
17     leds: out std_logic_vector(3 downto 0);
18     P     : out std_logic_vector(3 downto 0)
19 );
20 end MotorBiPasos;
21
22 architecture Control of MotorBiPasos is
23     type estado is (S1,S2,S3,S4); --Estados de la maquina
24     signal CONT: integer range 0 to 19999999:=0; --velocidad del motor
25     signal Qn, Qp : estado;
26 begin
27
28     PROCESS(CLK)
29     BEGIN
30         IF CLK 'EVENT AND CLK = '1' THEN
31             CONT <= CONT + 1;
32             IF CONT = 19999999 THEN
33                 Qp <= Qn; --QP estado actual QN estado siguiente
34                 CONT <= 0;
35             END IF;
36         END IF;
37     END PROCESS;
38

```

Se establecen las librerías y la entidad
(las entradas y salidas que se utilizaran
para el Código).

En la primera parte de la arquitectura se establece las
señales que se utilizaran para realizar los procesos de
la frecuencia de reloj y de los estados que se realizan
para ejecutar la secuencia.

Sabemos que obtenemos el divisor de frecuencia
estableciendo un contador que tiene un intervalo 0 a
19999999... donde se va incrementando y al llegar a
su máximo valor se reinicia así, se controla la
velocidad del giro del motor.

(Entre mayor sea el rango menor será la velocidad de
giro y viceversa).

```

39 Combinational: process (qp, opc)
40 begin
41   case qp is
42     when S1 => --00 01
43       if(opc='1')then
44         P <= "0101"; --BB-AA --0001
45         leds<= "0101";
46         Qn <= S2;
47       else
48         P <= "0110"; --BB-AA --0001
49         leds<= "1110";
50         Qn <= S2;
51       end if;
52     when S2 =>
53       if(opc='1')then
54         P <= "1001";
55         leds <= "1001";
56         Qn <= S3;
57       else
58         P <= "1010";
59         leds <= "1101";
60         Qn <= S3;
61       end if;
62     when S3 =>
63       if(opc='1')then
64         P <= "1010";
65         leds <= "1010";
66         Qn <= S4;
67       else
68         P <= "1001";
69         leds <= "1011";
70         Qn <= S4;
71       end if;
72     when S4 =>
73       if(opc='1')then
74         P <= "0110";
75         leds <= "0110";
76         Qn <= S1;
77       else
78         P <= "0101";
79         leds <= "0111";
80         Qn <= S1;
81       end if;
82     end case;
83   end process Combinational;
84 end Control;

```

-En esta primera parte se muestra el process de nombre combinational el cual solo requiere como parámetros Qp y OPC, realizamos un case de Qp para saber en qué estado se encuentra el motor, si el motor esta en S1 verificamos con un if a que sentido debe de girar, en este caso será a contra sentido del reloj, después de saber a que sentido girara el motor, asignamos la salida al motor con la variable P, leds nos mostrada en que sentido esta girando prendiendo y apagando unos leds que tendrán la misma salida que el motor y por ultimo asignamos a Qn el siguiente estado que será S2, si es a favor del sentido del reloj los leds prenderán solo 1 en cada estado siendo el primer estado el led de la derecha y el estado S4 el led de la izquierda para así diferenciar en que sentido gira el motor en caso de que no se aprecie de forma visual en el motor. En cada uno de los estados se realiza este proceso.

En esta parte se muestra el final del process de nombre combinational y con esto terminaría el código para girar el motor bipolar.