Digital System Design Course

Problemas de diseño, maquinas de estado e implementadas en Quartus II

Javier Felipe Pérez Código : 0523037

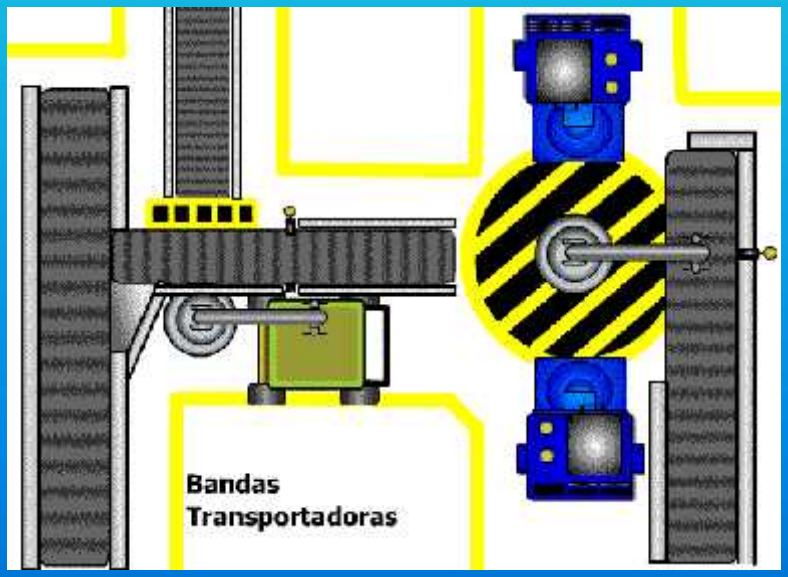
La temática se dividen en los siguientes pasos :

- Descripción de los problemas a resolver.
- Implementación de la solución al problema por diseño de diagrama de estados.
- Implementación de la solución al problema por VHDL
- Verificación de resultados mediante simulación.

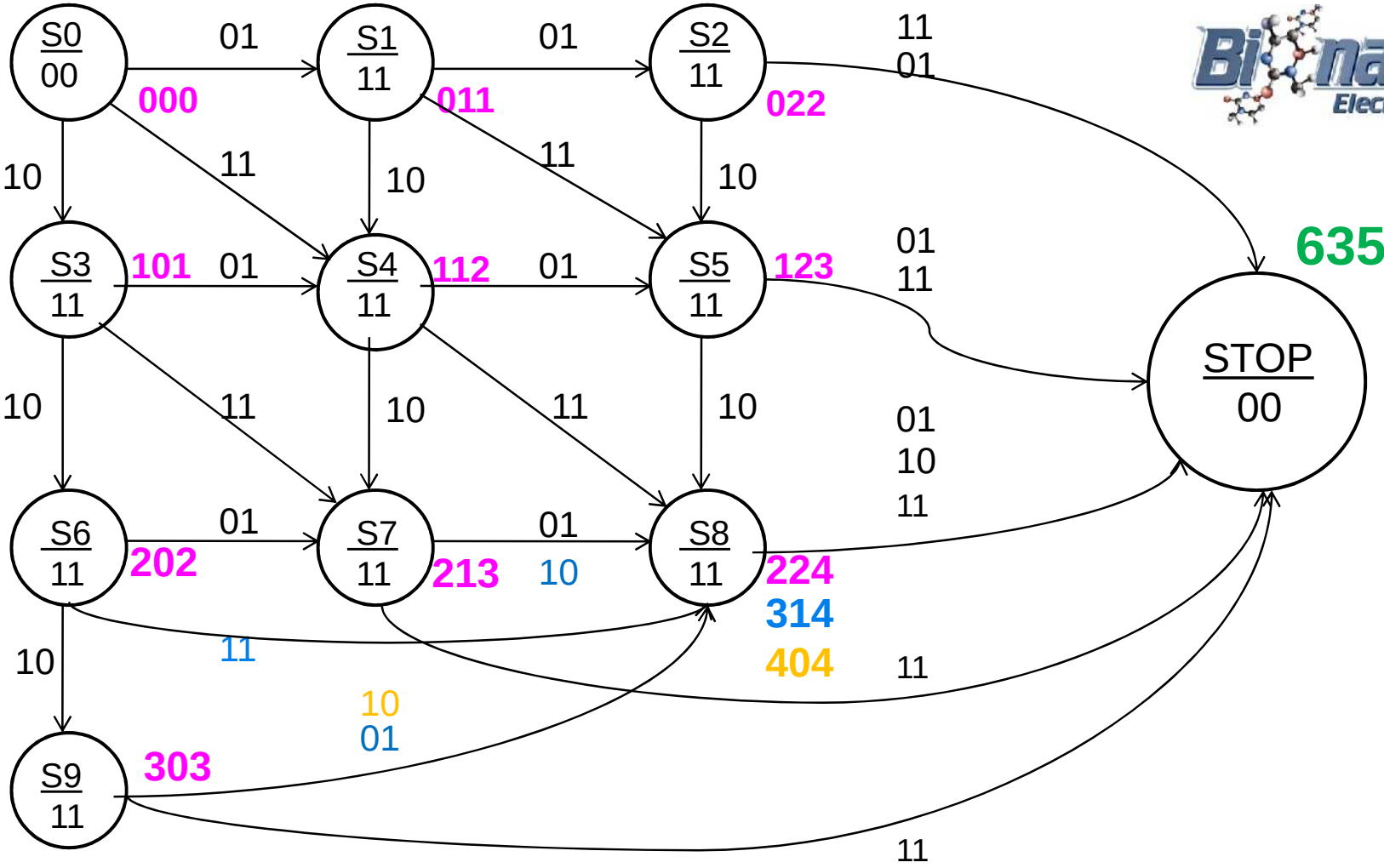
I. Descripción de los problemas a resolver.

- .

Diseñar una FSM para controlar los motores de dos bandas de transporte en un proceso industrial. El proceso se detiene si por la banda 1 (B1) pasan 3 objetos o si por la banda 2 (B2) pasan 6 objetos o si por las dos bandas (B2 Y B1) pasan 5 objetos.



I.Solución mediante diagrama de estados



I. DISEÑO TABLA DE ESTADOS Y FLIP FLOPS J- K

Y	X	q3	q2	q1	q0	Q3	Q2	Q1	Q0	J3	K3	J2	K2	J1	K1	J0	K0
0	0	0	0	0	0	0	0	0	0	0	X	0	X	0	X	0	X
0	0	0	0	0	1	0	0	0	1	0	X	0	X	0	X	X	0
0	0	0	0	1	0	0	0	1	0	0	X	0	X	X	0	0	X
0	0	0	0	1	1	0	0	1	1	0	X	0	X	X	0	X	0
0	0	0	1	0	0	0	1	0	0	0	X	X	0	0	X	0	X
0	0	0	1	0	1	0	1	0	1	0	X	X	0	0	X	X	0
0	0	0	1	1	0	0	1	1	0	0	X	X	0	X	0	0	X
0	0	0	1	1	1	0	1	1	1	0	X	X	0	X	0	X	0
0	0	1	0	0	0	1	0	0	0	X	0	0	X	0	X	0	X
0	0	1	0	0	1	1	0	0	1	X	0	0	X	0	X	X	0
0	0	1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X
0	0	1	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X
0	0	1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X
0	0	1	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X
0	0	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X

I. DISEÑO TABLA DE ESTADOS Y FLIP FLOPS J- K

Y	X	q3	q2	q1	q0	Q3	Q2	Q1	Q0	J3	K3	J2	K2	J1	K1	J0	K0
0	1	0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	1	0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	1	0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	1	0	0	1	1	0	0	1	1	0	X	0	X	X	0	X	0
0	1	0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	0	1	1	0	0	0	1	1	0	X	X	1	X	0	1	X
0	1	0	1	1	1	1	0	1	1	1	X	X	1	X	0	X	0
0	1	1	0	0	0	0	0	1	1	X	1	0	X	1	X	1	X
0	1	1	0	0	1	1	0	0	0	X	0	0	X	0	0	0	X
0	1	1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X

I. DISEÑO TABLA DE ESTADOS Y FLIP FLOPS J- K

Y	X	q3	q2	q1	q0	Q3	Q2	Q1	Q0	J3	K3	J2	K2	J1	K1	J0	K0
1	0	0	0	0	0	0	1	0	0	X	0	1	X	0	X	0	X
1	0	0	0	0	1	0	1	0	1	X	0	1	X	0	X	X	0
1	0	0	0	1	0	0	1	1	0	0	X	1	X	X	0	X	0
1	0	0	0	1	1	0	0	1	1	0	X	0	X	X	0	X	0
1	0	0	1	0	0	0	1	1	1	0	X	X	0	1	X	1	X
1	0	0	1	0	1	1	0	1	1	1	X	X	1	1	X	X	0
1	0	0	1	1	0	1	0	0	0	1	X	X	1	X	1	0	X
1	0	0	1	1	1	1	0	0	1	1	1	X	X	1	X	1	X
1	0	1	0	0	0	0	0	1	1	X	1	0	X	1	X	1	X
1	0	1	0	0	1	1	0	0	0	X	0	0	X	1	X	1	X
1	0	1	0	1	0	1	0	1	0	X	1	0	X	X	0	1	X
1	0	1	0	1	1	X	X	X	X	X	0	0	X	X	1	X	1
1	0	1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X
1	0	1	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X
1	0	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X

I. DISEÑO TABLA DE ESTADOS Y FLIP FLOPS J- K

Y	X	q3	q2	q1	q0	Q3	Q2	Q1	Q0	J3	K3	J2	K2	J1	K1	J0	K0
1	1	0	0	0	0	0	1	0	1	X	1	X	0	X	1	X	0
1	1	0	0	0	1	0	1	1	0	0	X	1	X	1	X	X	1
1	1	0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
1	1	0	0	1	1	0	0	1	1	0	X	0	X	X	0	X	0
1	1	0	1	0	0	1	0	1	1	1	X	X	1	1	X	X	1
1	1	0	1	0	1	1	0	0	0	1	X	X	1	0	X	X	1
1	1	0	1	1	0	0	0	1	1	0	X	X	1	X	0	1	X
1	1	0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	1	1	0	0	0	0	0	1	1	X	1	0	X	1	X	1	X
1	1	1	0	0	1	0	1	1	1	X	1	0	X	1	X	X	0
1	1	1	0	1	0	1	0	1	0	X	1	0	X	1	X	X	0
1	1	1	0	1	1	0	0	1	1	X	1	0	X	X	0	X	0
1	1	1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X

I. Proceso de simplificación de tablas por el método de Mapas-K.

Ecuaciones de los FLIP-FLOPS J-K :

- $J_0 = (Y^*/X^*q_2^*/q_1)+(Y^*q_3^*/q_1)+(Y^*X^*q_1)+(X^*/q_2^*/q_1)+(/Y^*X^*/q_3)$

- $K_0 = (X^*/q_3^*/q_1)+(X^*q_2)+(Y^*q_2^*/q_1)+(Y^*/X^*q_3)+(/Y^*X^*/q_1)$

- $J_1 = (X^*q_3^*/q_0)+(Y^*q_2^*/q_0)+(Y^*/X^*q_2)+(Y^*q_3^*/q_0)+(/Y^*X^*/q_3^*q_0)+(Y^*X^*/q_2^*q_0)$
- $K_1 = (Y^*/X^*q_2)+(Y^*/X^*q_3)+(/Y^*X^*q_3)+(Y^*q_2^*q_0)$

- $J_2 = (Y^*/q_3^*/q_1)+(Y^*/X^*/q_3^*/q_0)$
- $K_2 = (X^*q_1)+(Y^*q_0)+(Y^*q_1)+(Y^*X)$

- $J_3 = (X^*q_2^*q_1^*q_0)+(Y^*q_2^*q_0)+(Y^*X^*q_2^*/q_1)+(Y^*/X^*q_2^*q_1)$
- $K_3 = (X^*/q_1^*/q_0)+(Y^*/q_1^*/q_0)+(Y^*X)$

Ecuaciones de salida:

- $S_0 = (q_1^*/q_0)+(/q_1^*q_0)+(q_2)+(q_3)$
- $S_1 = (q_1^*/q_0)+(/q_1^*q_0)+(q_2)+(q_3)$

I. SOLUCIÓN MEDIANTE VHDL

```
library IEEE;  -- Se invoca Librería IEEE
use IEEE.STD_LOGIC_1164.ALL  -- Bibliotecas de la librería
use IEEE.STD_LOGIC_ARITH.ALL  --Bibliotecas de la librería
use IEEE.STD_LOGIC_UNSIGNED.ALL  --Bibliotecas de la librería
entity  bandas1 is  -----Definición de la entidad

                        inicio: in std_logic;
                        ck  in std_logic;
                        X: in std_logic;
                        Y  in std_logic;
                        S0: out std_logic;
                        S1: out std_logic
                        );
end bandas1;  --Fin de la entidad
architecture architecture behavioral of bandas1 isDefinición de la arquitectura
de la entidad.
type nombres estados is (Q0, Q1, Q2, Q10, Q3, Q4, Q5, Q6, Q8, Q9, Q7); - -
Definición de las variables
signal estado: nombres estados; - -Definición de las variables.
signal entrada_aux: std_logic_vector (1 downto 0);   --Variable.
```

I. SOLUCIÓN MEDIANTE VHDL

```
process (inicio, ck)    --Inicio del ciclo "Decisión transición de estados"
begin
if inicio='1  then
                estado<=Q0;
elsif ck='1' and ck'event then
                case estado is   -- Casos
                        when Q0 =>
                                case entrada_aux is
                                        when "00" => estado<=Q0;
                                        when "01" => estado<=Q1;
                                        when "10" => estado<=Q3;
                                        when "11" => estado<=Q4;

                                end case;
                        when Q1 =>
                                case entrada_aux is
                                        when "00" => estado<=Q1;
                                        when "01" => estado<=Q2;
                                        when "10" => estado<=Q4;
                                        when "11" => estado<=Q5;
                                end case;
```

I. SOLUCIÓN MEDIANTE VHDL

```
            when Q2 =>
                case entrada_aux is
                            when "00" => estado<=Q2;
                            when "01" => estado<=Q10;
                            when "10" => estado<=Q5;
                            when "11" => estado<=Q10;
                    end case;
when Q10 =>
                case entrada_aux is
                            when "00" => estado<=Q10;
                            when "01" => estado<=Q10;
                            when "10" => estado<=Q10;
                            when "11" => estado<=Q10;
                            when others => estado<=Q0;
                end case;
when Q3 =>
                case entrada_aux is
                            when "00" => estado<=Q3;
                            when "01" => estado<=Q4;
                            when "10" => estado<=Q6;
                            when "11" => estado<=Q7;
                end case;
```

I. SOLUCIÓN MEDIANTE VHDL

```
when Q4 =>
        case entrada_aux is
                when "00" => estado<=Q4;
                when "01" => estado<=Q5;
                when "10" => estado<=Q7;
                when "11" => estado<=Q8;

        end case;
when Q5 =>
        case entrada_aux is
                when "00" => estado<=Q5;
                when "01" => estado<=Q10;
                when "10" => estado<=Q8;
                when "11" => estado<=Q10;

    when Q6 =>
        case entrada_aux is
                when "00" => estado<=Q6;
                when "01" => estado<=Q7;
                when  10" => estado<=Q9;
                when "11" => estado<=Q8;

        end case;
```

I. SOLUCIÓN MEDIANTE VHDL

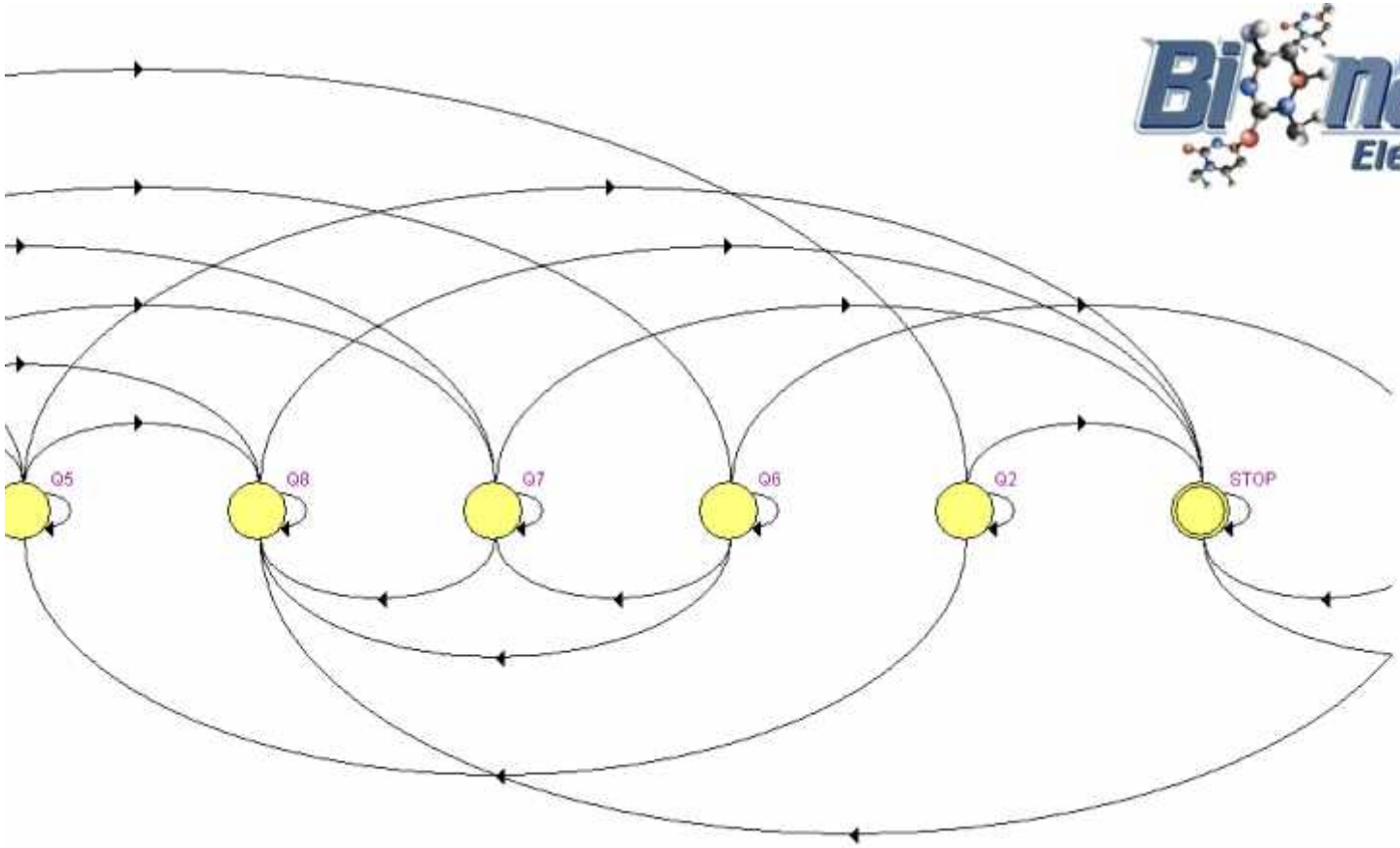
```
                when Q7 =>
                                case entrada_aux is
                                        when "00" => estado<=Q7;
                                        when "01" => estado<=Q8;
                                        when "10" => estado<=Q8;
                                        when "11" => estado<=Q10;
                                end case;
                 when Q8 =>
                                case entrada_aux is
                                        when "00" => estado<=Q8;
                                        when "01" => estado<=Q10;
                                        when "10" => estado<=Q10;
                                        when "11" => estado<=Q10;
                when Q9 =>
                                case entrada_aux is
                                        when "00" => estado<=Q9;
                                        when "01" => estado<=Q8;
                                        when "10" => estado<=Q8;
                                        when "11" => estado<=Q10;
               when STOP =>
                                case entrada_aux is
                                        when "00" => estado<= STOP;
                                        when "01" => estado<= STOP;
                                        when "10" => estado<= STOP;
                                        when "11" => estado<= STOP;
                                end case;
                        when others => estado<=Q0;
                end case;
        end if;
end process; -- Fin del ciclo
```

I. SOLUCIÓN MEDIANTE VHDL

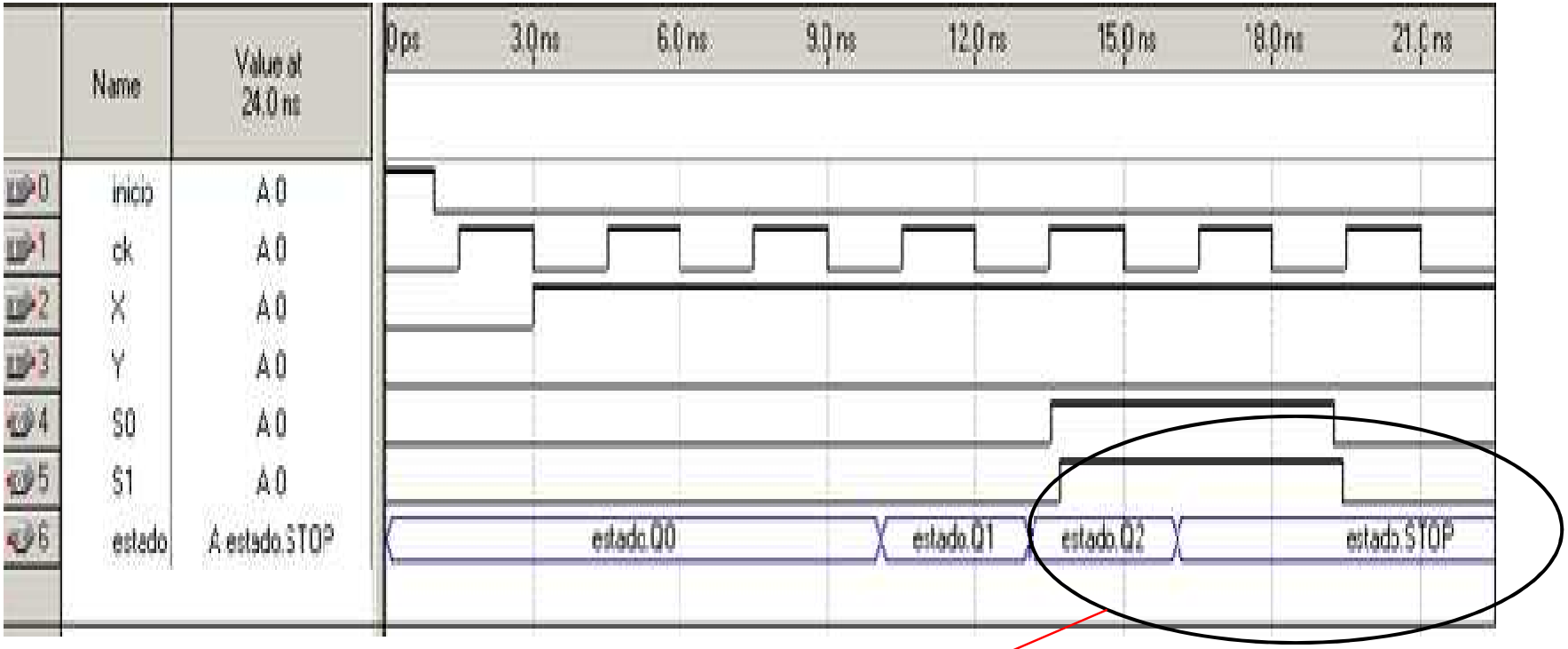
```
Process (estado) –- Ciclo para las salidas de la FSM
 begin
        case estado is
                when Q0 =>
                        S0<='0';
                        S1<='0';
                 when Q1 =>
                        S0<='1';
                        S1<='1';
                 when Q2 =>
                        S0<='1';
                        S1<='1';
                when Q10 =>
                         S0<='0';
                        S1<='0';
                 when Q3 =>
                        S0<='1';
                        S1<='1';
                 when Q4 =>
                        S0<='1';
                        S1<='1';
                 when Q5 =>
                        S0<='1';
                        S1<='1';
```

```
                when Q6 =>
                    S0<='0';
                    S1<='0';
                when Q7 =>
                    S0<='1';
                    S1<='1';
                when Q8 =>
                    S0<='1';
                    S1<='1';
                when Q9 =>
                    S0<='0';
                    S1<='0';
                when STOP=>
                    S0<='0';
                    S1<='0'
end case;
end process; --Fin ciclo de salidas
        end behavioral;-- Fin  de la arquitectura de la entidad
```

Diagrama de estados bandas en Quartus II

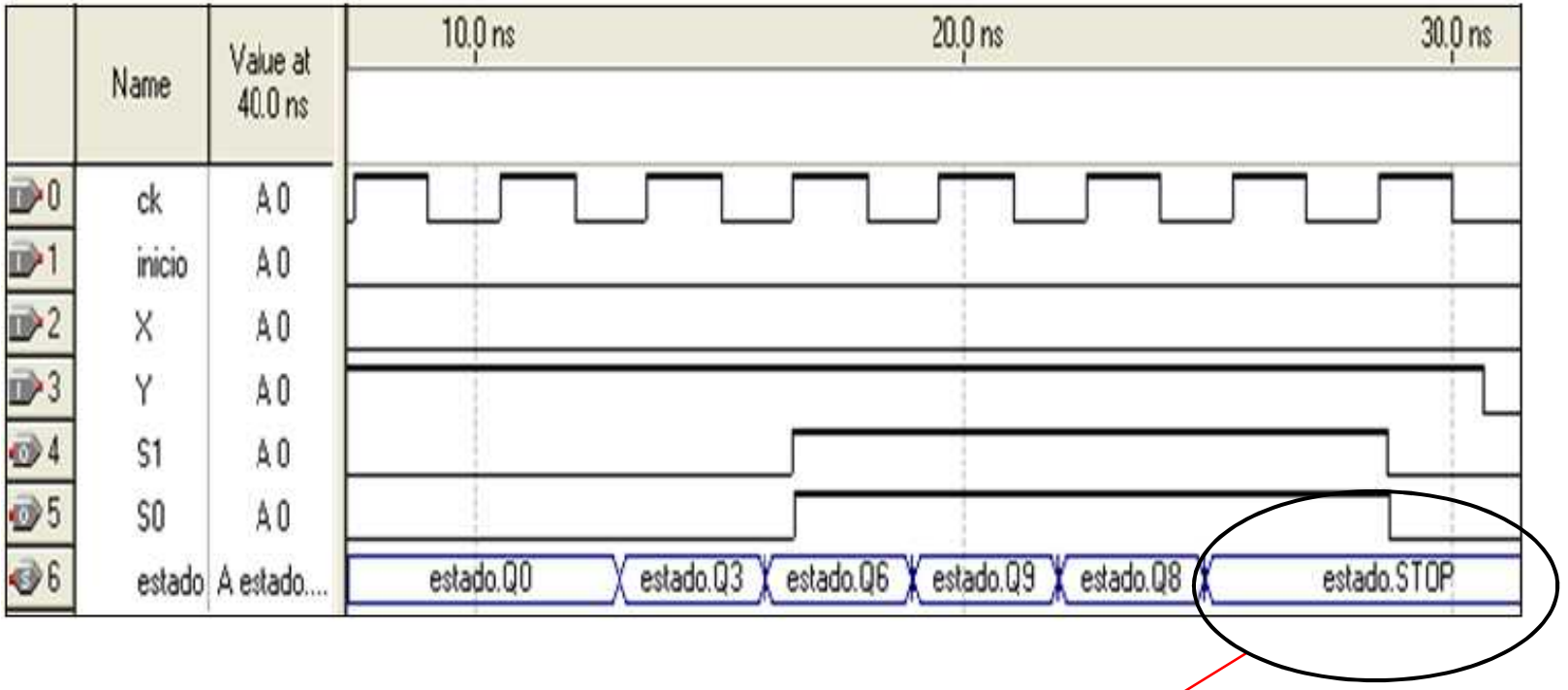


Simulación y verificación de resultados



Parada por la banda 1

Simulación y verificación de resultados



Parada por la banda 2

Descripción de problema 2.

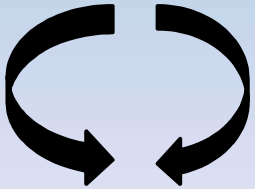
Diseñar una FSM para controlar un motor D.C. mediante diagramas de estados:

- El motor arranca y gira en sentido normal cuando detecta la secuencia X: 1,0,1,0,1

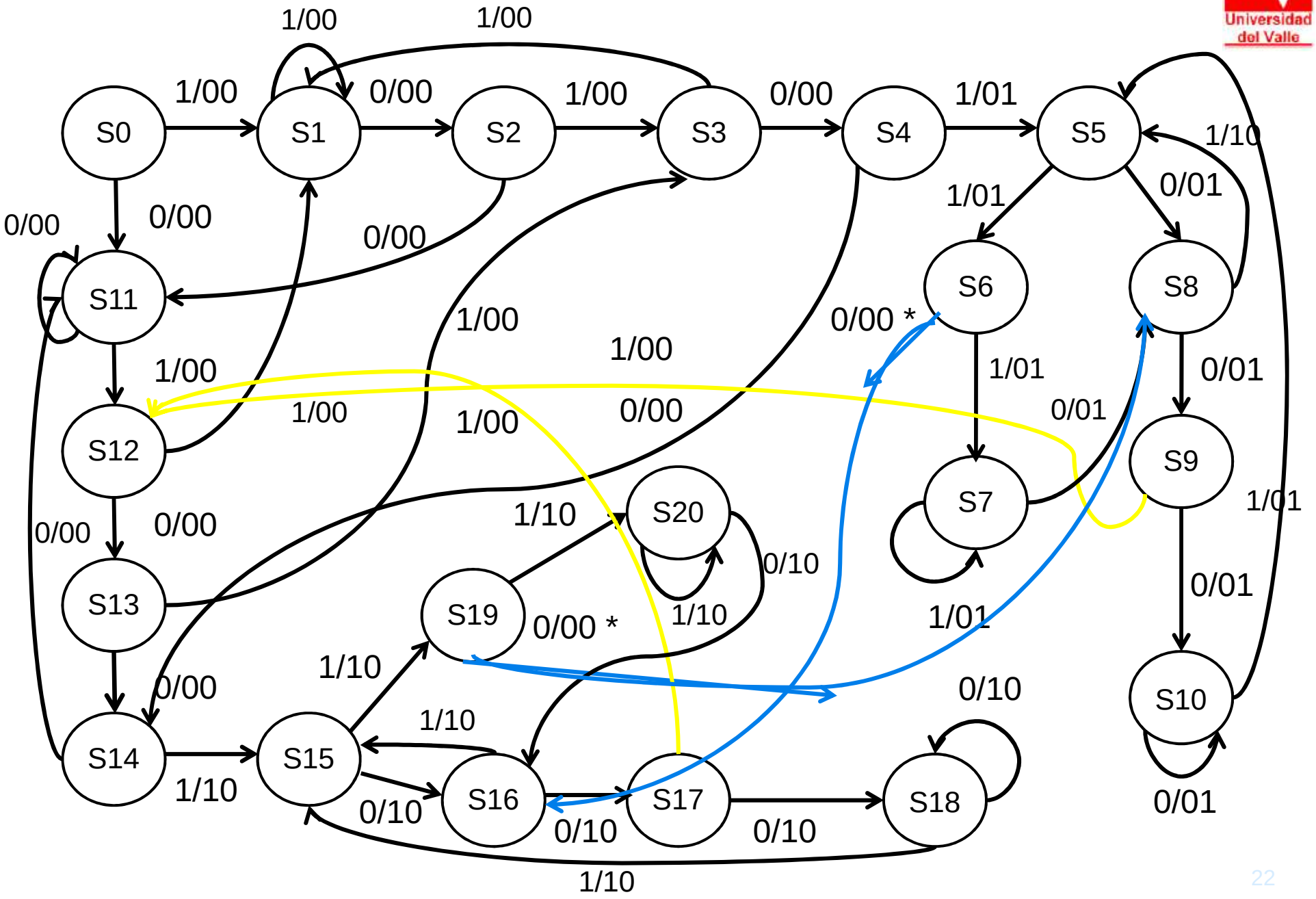
- El motor arranca y gira en sentido inverso cuando detecta la secuencia X: 0,1,0,0,1

- El motor se detiene después de arrancar, si el circuito detecta la secuencia X: 1,0,0,1 y permanece detenido hasta detectar de nuevo una secuencia de arranque.

- Sin embargo, después de arrancar el motor , cada vez que el circuito detecta la secuencia X: 0,1,1,0 el motor debe girar en sentido contrario, pero primero debe parar un instante.



Diseño diagramas de estado del motor D.C



II. SOLUCIÓN MEDIANTE VHDL

```
Library IEEE; -- Se invoca Librería IEEE
use IEEE.STD_LOGIC_1164.ALL; -- Bibliotecas de la librería
use IEEE.STD_LOGIC_ARITH.ALL; Bibliotecas de la librería
use IEEE.STD_LOGIC_UNSIGNED.ALL; Bibliotecas de la librería

entity motorV1 is Definición  e inicio de la entidad
        Port (
                inicio: in std_logic;
                ck: in std_logic;
                x: in std_logic;
                S0: out std_logic;
                S1: out std_logic
                );
end motorV1;

architecture behavioral of motorV1 is Definición de la arquitectura de la entidad.

type nombres_estados is (Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q9, Q10, Q08, Q11, Q12, Q13, Q14, Q15,
Q16, Q17, Q18, Q19, Q20);
signal estado: nombres_estados;
signal entrada_aux: std_logic;
```

II. SOLUCIÓN MEDIANTE VHDL

```
begin
entrada_aux<=x;
process(inicio,ck)
begin
if inicio='1' then
                estado<=Q0;
elsif ck='1' and ck'event then
                case estado is

                        when Q0 =>
                                case entrada_aux is
                                        when '0' => estado<=Q11;
                                        when '1' => estado<=Q1;
                                        when others => estado<=Q0;
                                end case;
                        when Q1 =>
                                case entrada_aux is
                                        when '0' => estado<=Q2;
                                        when '1' => estado<=Q1;
                                        when others => estado<=Q0;
                                end case;
                        when Q2 =>
                                case entrada_aux is
                                        when '0' => estado<=Q11;
                                        when '1' => estado<=Q3;
                                        when others => estado<=Q0;
                                end case;
```

II. SOLUCIÓN MEDIANTE VHDL

```
when Q3 =>
        case entrada_aux is
                when ’0’ => estado<=Q4;
                when ’1’ => estado<=Q1;
                when others => estado<=Q0;
        end case;
when Q4 =>
        case entrada_aux is
                when ’0’ => estado<=Q14;
                when ’1’ => estado<=Q5;
                when others => estado<=Q0;
        end case;
when Q5 =>
        case entrada_aux is
                when ’0’ => estado<=Q08;
                when ’1’ => estado<=Q6;
                when others => estado<=Q0;
        end case;

 when Q6 =>
        case entrada_aux is
                when ’0’ => estado<=Q16;
                when ’1’ => estado<=Q7;
                when others => estado<=Q0;
        end case;
```

Solución 2 mediante VHDL

```
when Q7 =>
        case entrada_aux is
                when ’0’ => estado<=Q08;
                when ’1’ => estado<=Q7;
                when others => estado<=Q0;
        end case;
when Q9 =>
        case entrada_aux is
                when ’0’ => estado<=Q10;
                when ’1’ => estado<=Q12;
                when others => estado<=Q0;
        end case;
when Q10 =>
        case entrada_aux is
                when ’0’ => estado<=Q10;
                when ’1’ => estado<=Q5;
                when others => estado<=Q0;
        end case;
when Q08 =>
        case entrada_aux is
                when ’0’ => estado<=Q9;
                when ’1’ => estado<=Q5;
                when others => estado<=Q0;
        end case;
```

Solución 2 mediante VHDL

```
when Q11 =>
        case entrada_aux is
                when ’0’ => estado<=Q11;
                when ’1’ => estado<=Q12;
                when others => estado<=Q0;
        end case;
when Q12 =>
        case entrada_aux is
                when ’0’ => estado<=Q13;
                when ’1’ => estado<=Q1;
                when others => estado<=Q0;
        end case;
when Q13 =>
        case entrada_aux is
                when ’0’ => estado<=Q14;
                when ’1’ => estado<=Q3;
                when others => estado<=Q0;
        end case;
when Q14 =>
        case entrada_aux is
                when ’0’ => estado<=Q11;
                when ’1’ => estado<=Q15;
                when others => estado<=Q0;
        end case;
```

SOLUCIÓN II MEDIANTE VHDL

```
when Q11 =>
        case entrada_aux is
                when ’0’ => estado<=Q11;
                when ’1’ => estado<=Q12;
                when others => estado<=Q0;
        end case;
when Q12 =>
        case entrada_aux is
                when ’0’ => estado<=Q13;
                when ’1’ => estado<=Q1;
                when others => estado<=Q0;
        end case;
when Q13 =>
        case entrada_aux is
                when ’0’ => estado<=Q14;
                when ’1’ => estado<=Q3;
                when others => estado<=Q0;
        end case;
when Q14 =>
        case entrada_aux is
                when ’0’ => estado<=Q11;
                when ’1’ => estado<=Q15;
                when others => estado<=Q0;
        end case;
```

SOLUCIÓN II MEDIANTE VHDL

```
when Q11 =>
        case entrada_aux is
                when ’0’ => estado<=Q11;
                when ’1’ => estado<=Q12;
                when others => estado<=Q0;
        end case;
when Q12 =>
        case entrada_aux is
                when ’0’ => estado<=Q13;
                when ’1’ => estado<=Q1;
                when others => estado<=Q0;
        end case;
when Q13 =>
        case entrada_aux is
                when ’0’ => estado<=Q14;
                when ’1’ => estado<=Q3;
                when others => estado<=Q0;
        end case;
when Q14 =>
        case entrada_aux is
                when ’0’ => estado<=Q11;
                when ’1’ => estado<=Q15;
                when others => estado<=Q0;
        end case;
```

SOLUCIÓN II MEDIANTE VHDL

```
when Q15 =>
        case entrada_aux is
                when '0' => estado<=Q16;
                when '1' => estado<=Q19;
                when others => estado<=Q0;
        end case;
when Q16 =>
        case entrada_aux is
                when '0' => estado<=Q17;
                when '1' => estado<=Q15;
                when others => estado<=Q0;
        end case;
when Q17 =>
        case entrada_aux is
                when '0' => estado<=Q18;
                when '1' => estado<=Q12;
                when others => estado<=Q0;
        end case;
when Q18 =>
        case entrada_aux is
                when '0' => estado<=Q18;
                when '1' => estado<=Q15;
                when others => estado<=Q0;
        end case;
```

SOLUCIÓN II MEDIANTE VHDL

```
                when Q19 =>
                        case entrada_aux is
                                when ’0’ => estado<=Q08;
                                when ’1’ => estado<=Q20;
                                when others => estado<=Q0;
                        end case;
                when Q20 =>
                        case entrada_aux is
                                when ’0’ => estado<=Q16;
                                when ’1’ => estado<=Q20;
                                when others => estado<=Q0;
                        end case;
                when others => estado<=Q0;
        end case;
end if;
end process;

process(estado, entrada_aux) -- Lógica de las  salidas de la FSM
Begin
```

SOLUCIÓN II MEDIANTE VHDL

```
case estado is
        when Q0 =>
                case entrada_aux is
                        when '0' =>
                                S0<='0';
                                S1<='0';
                        when '1' =>
                                S0<='0';
                                S1<='0';
                end case;
        when Q1 =>
                case entrada_aux is
                        when '0' =>
                                S0<='0';
                                S1<='0';
                        when '1' =>
                                S0<='0';
                                S1<='0';
                end case;
        when Q2 =>
                case entrada_aux is
                        when '0' =>
                                S0<='0';
                                S1<='0';
                        when '1' =>
                                S0<='0';
                                S1<='0';
                end case;
```

SOLUCIÓN II MEDIANTE VHDL

```
when Q3 =>
        case entrada_aux is
                when ’0’ =>
                        S0<=’0’;
                        S1<=’0’;
                when ’1’ =>
                        S0<=’0’;
                        S1<=’0’;
        end case;
when Q4 =>
        case entrada_aux is
                when ’0’ =>
                        S0<=’0’;
                        S1<=’0’;
                when ’1’ =>
                        S0<=’0’;
                        S1<=’1’;
        end case;
when Q5 =>
        case entrada_aux is
                when ’0’ =>
                        S0<=’0’;
                        S1<=’1’;
                when ’1’ =>
                        S0<=’0’;
                        S1<=’1’;
        end case;
```

SOLUCIÓN II MEDIANTE VHDL

```
when Q6 =>
        case entrada_aux is
                when ’0’ =>
                        S0<=’1’;
                        S1<=’1’;
                when ’1’ =>
                        S0<=’0’;
                        S1<=’1’;
        end case;
when Q7 =>
        case entrada_aux is
                when ’0’ =>
                        S0<=’0’;
                        S1<=’1’;
                when ’1’ =>
                        S0<=’0’;
                        S1<=’1’;
        end case;
when Q9 =>
        case entrada_aux is
                when ’0’ =>
                        S0<=’0’;
                        S1<=’1’;
                when ’1’ =>
                        S0<=’0’;
                        S1<=’0’;
        end case;
```

SOLUCIÓN II MEDIANTE VHDL

```
when Q10 =>
        case entrada_aux is
                when ’0’ =>
                        S0<=’0’;
                        S1<=’1’;
                when ’1’ =>
                        S0<=’0’;
                        S1<=’1’;
        end case;
when Q08 =>
        case entrada_aux is
                when ’0’ =>
                        S0<=’0’;
                        S1<=’1’;
                when ’1’ =>
                        S0<=’0’;
                        S1<=’1’;
        end case;
when Q11 =>
        case entrada_aux is
                when ’0’ =>
                        S0<=’0’;
                        S1<=’0’;
                when ’1’ =>
                        S0<=’0’;
                        S1<=’0’;
        end case;
```

SOLUCIÓN II MEDIANTE VHDL

```
 when Q12 =>
                case entrada_aux is
                                when ’0’ =>
                                                S0<=’0’;
                                                S1<=’0’;
                                when ’1’ =>
                                                S0<=’0’;
                                                S1<=’0’;
                end case;
when Q13 =>
                case entrada_aux is
                                when ’0’ =>
                                                S0<=’0’;
                                                S1<=’0’;
                                when ’1’ =>
                                                S0<=’0’;
                                                S1<=’0’;
                end case;
when Q14 =>
                case entrada_aux is
                                when ’0’ =>
                                                S0<=’0’;
                                                S1<=’0’;
                                when ’1’ =>
                                                S0<=’1’;
                                                S1<=’0’;
                end case;
```

SOLUCIÓN II MEDIANTE VHDL

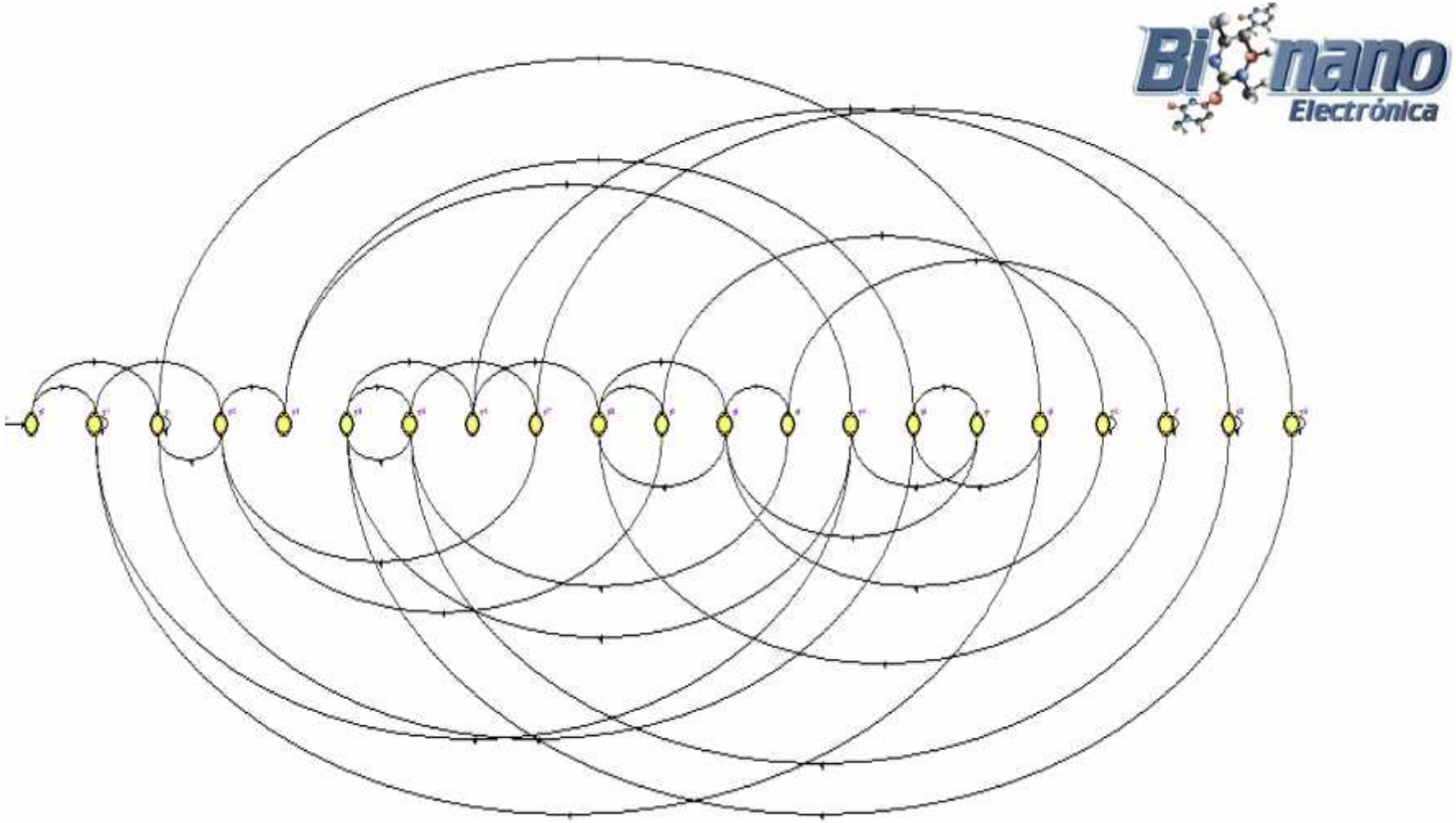
```
 when Q15 =>
                case entrada_aux is
                                when ’0’ =>
                                                S0<=’1’;
                                                S1<=’0’;
                                when ’1’ =>
                                                S0<=’1’;
                                                S1<=’0’;
                end case;
when Q16 =>
                case entrada_aux is
                                when ’0’ =>
                                                S0<=’1’;
                                                S1<=’0’;
                                when ’1’ =>
                                                S0<=’1’;
                                                S1<=’0’;
                end case;
when Q17 =>
                case entrada_aux is
                                when ’0’ =>
                                                S0<=’1’;
                                                S1<=’0’;
                                when ’1’ =>
                                                S0<=’0’;
                                                S1<=’0’;
                end case;
```

SOLUCIÓN II MEDIANTE VHDL

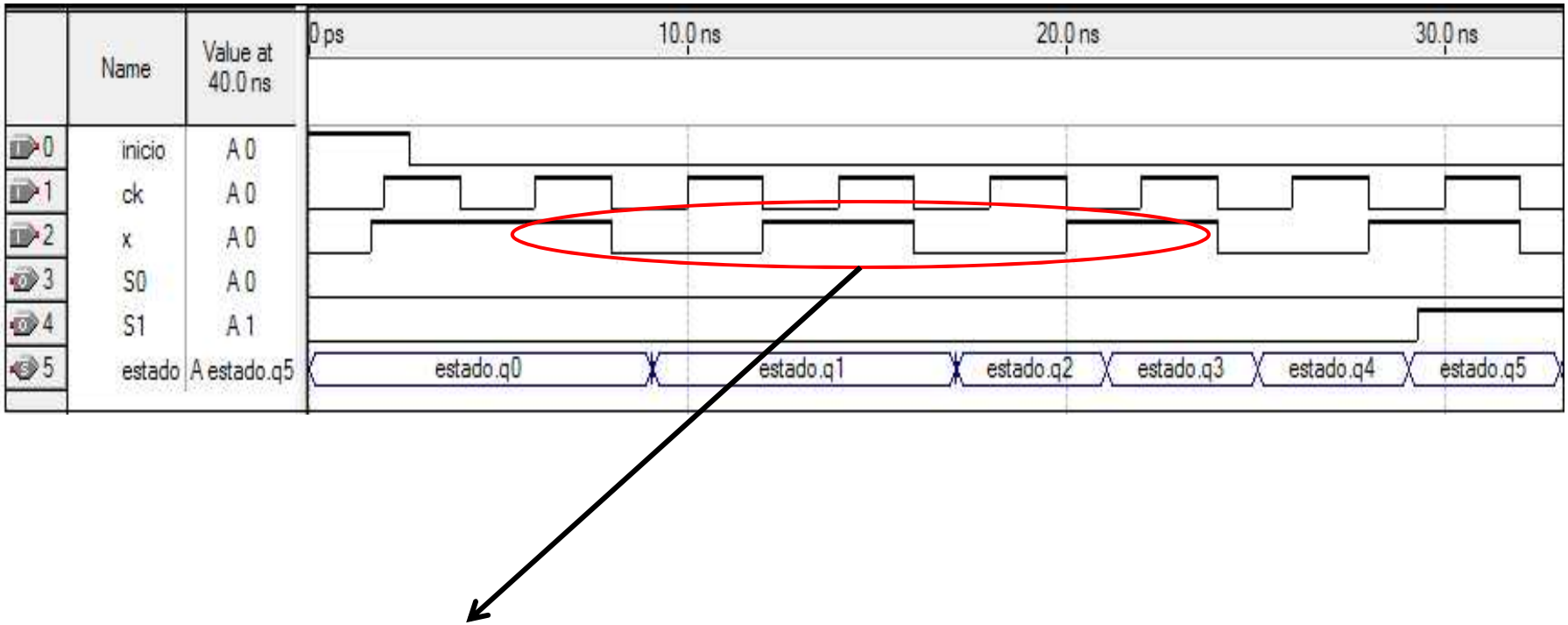
```
        when Q18 =>
                case entrada_aux is
                        when '0' =>
                                S0<='1';
                                S1<='0';
                        when '1' =>
                                S0<='1';
                                S1<='0';
                end case;
        when Q19 =>
                case entrada_aux is
                        when '0' =>
                                S0<='1';
                                S1<='1';
                        when '1' =>
                                S0<='1';
                                S1<='0';
                end case;
        when Q20 =>
                case entrada_aux is
                        when '0' =>
                                S0<='1';
                                S1<='0';
                        when '1' =>
                                S0<='1';
                                S1<='0';
                end case;
    end case;
end process;

end behavioral;
```

Diagrama de estados motor D.C en Quartus II

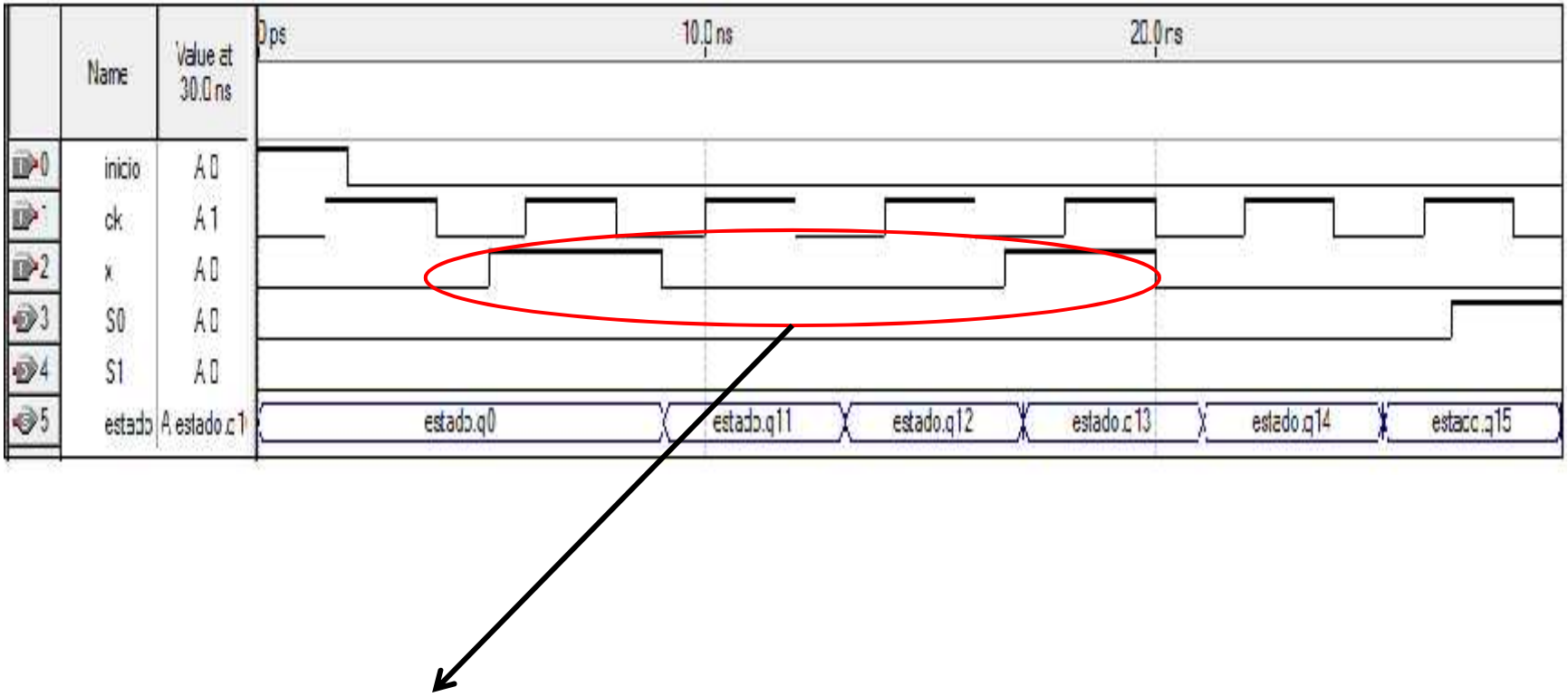


Simulación y verificación de resultados



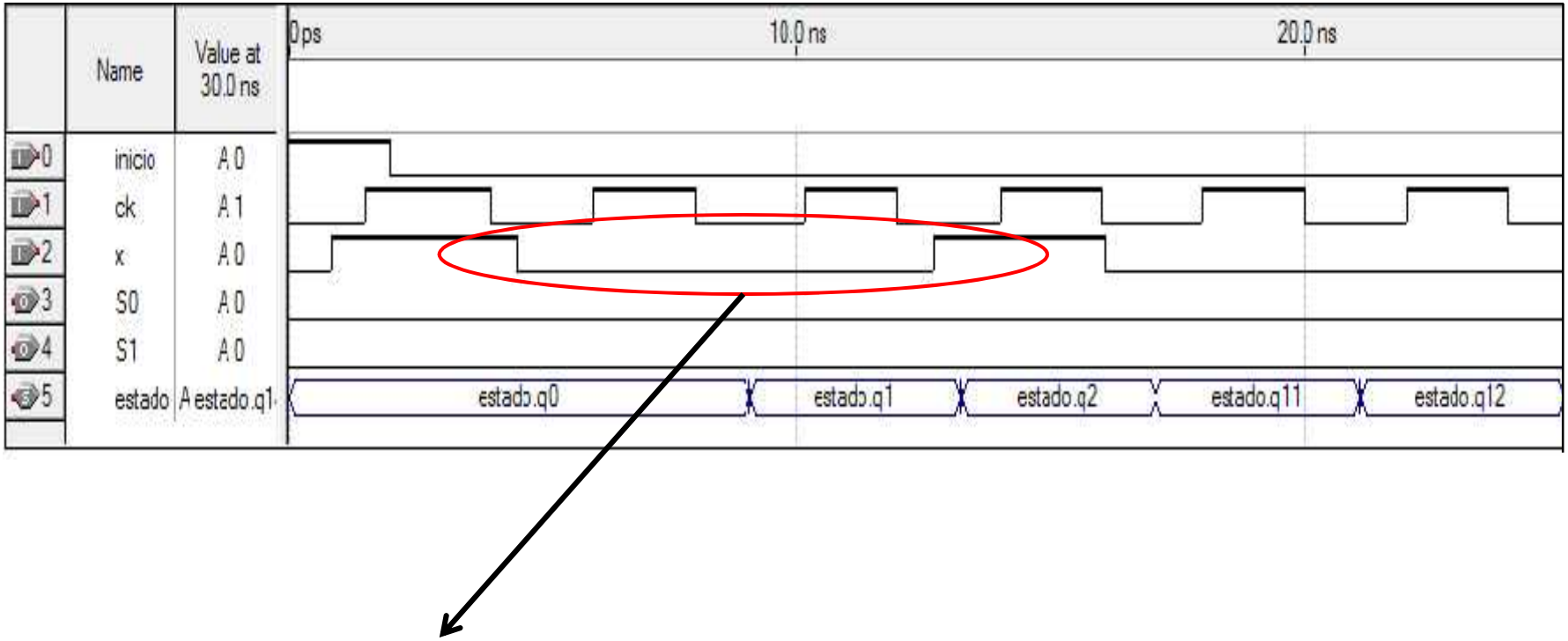
Arranque Sentido Directo

Simulación y verificación de resultados



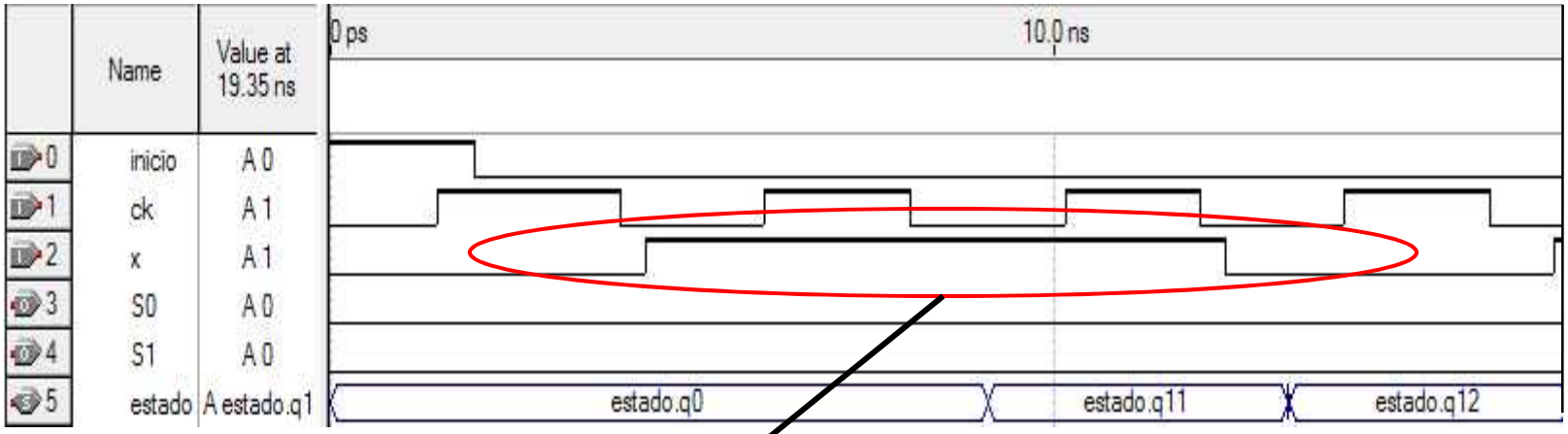
Arranque Sentido Inverso

Simulación y verificación de resultados



Parada

Simulación y verificación de resultados



Cambio De Giro