

Universidad del Valle de Guatemala

Electrónica Digital 1

Francisco Montúfar 19379

Proyecto No.1 Máquinas de estados finitos

Link repositorio: <https://github.com/mon19379/PROYECTO.git>

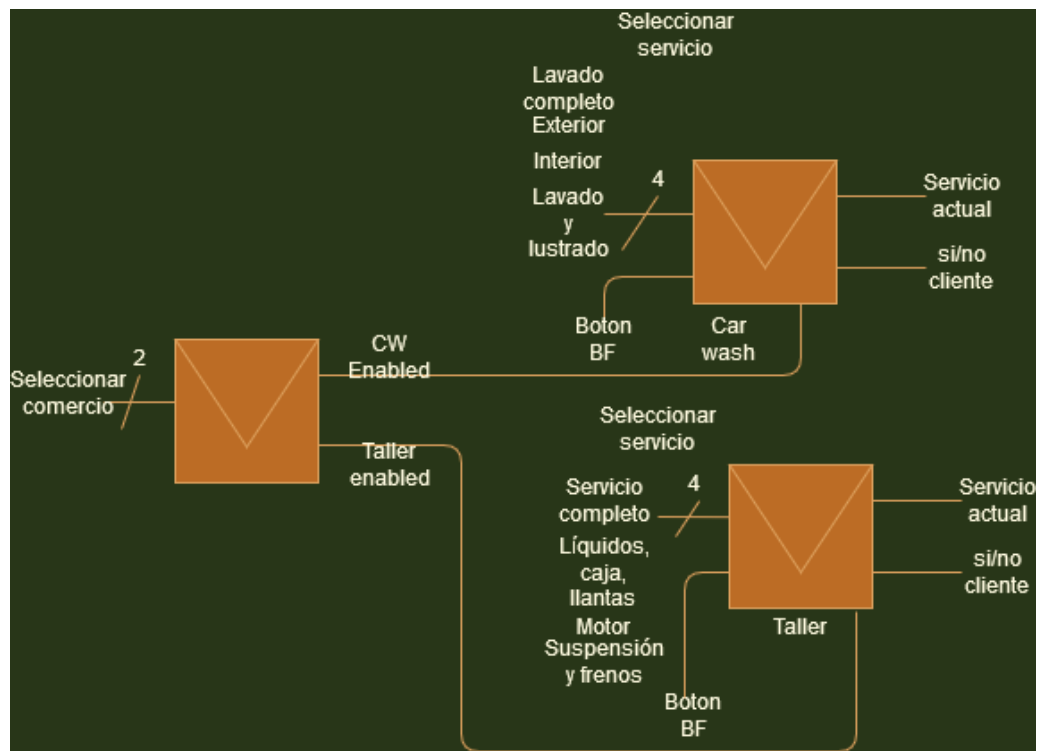
Link Video parte 1: https://youtu.be/40bcoT_z34Q

Link video parte 2: <https://youtu.be/a7FbT5QtsvY>

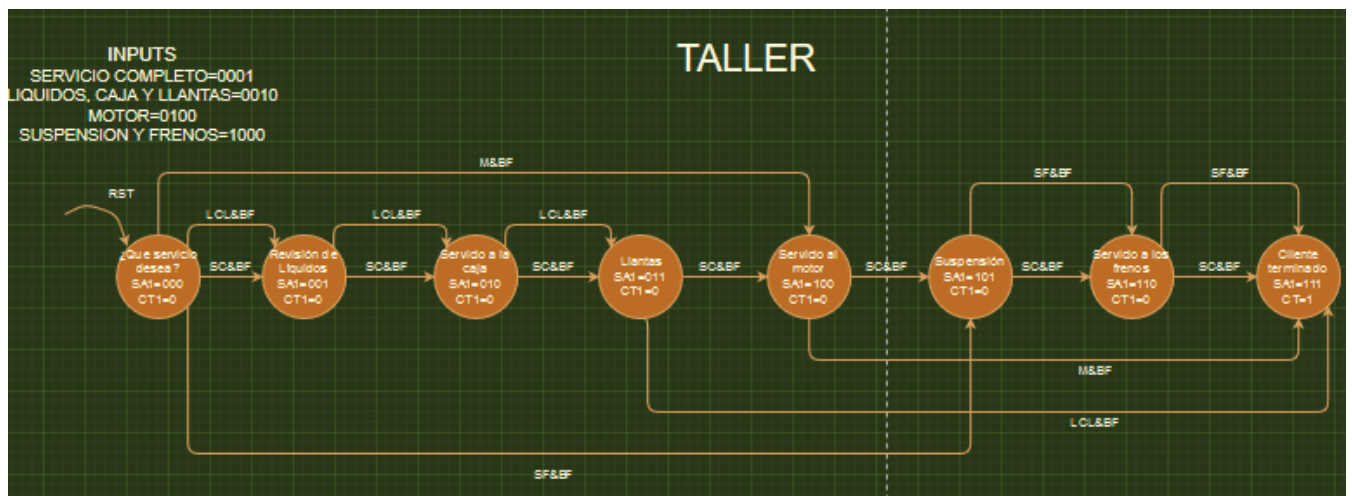
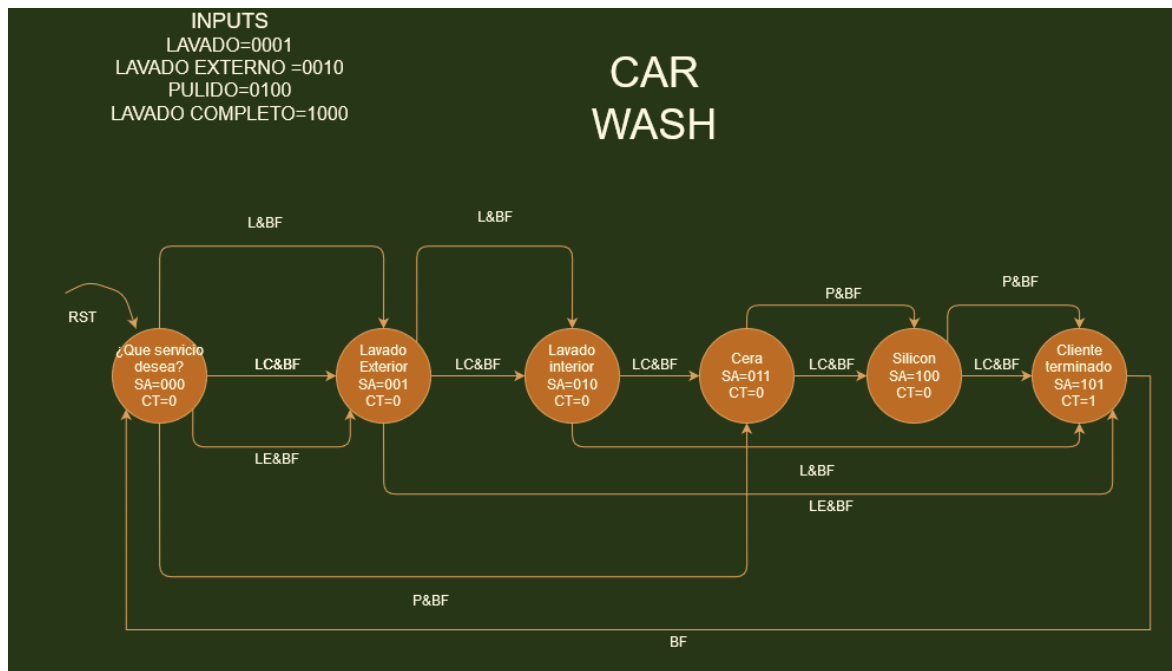
1.Resumen y descripción:

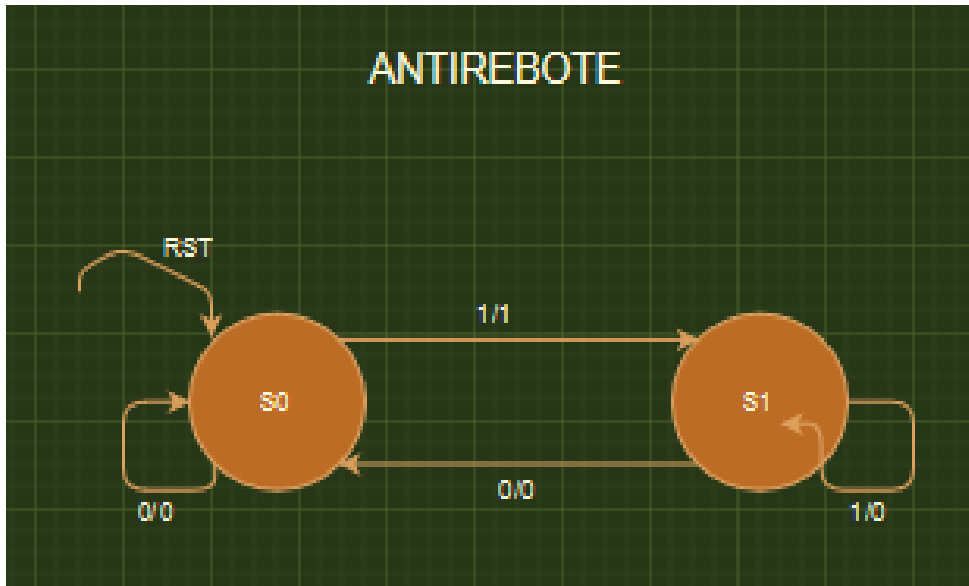
Este proyecto consiste en la representación de un complejo de car wash/taller por medio de una máquina de estados finitos. Primero se realizó un diagrama de cajas negras en donde se describe el funcionamiento general de la máquina. En el diagrama se puede observar que primero se escoge un complejo y luego en el complejo se escoge algún servicio. Depende al servicio escogido serán las transiciones entre estados y cuando este servicio finaliza, se enciende un indicador que da la señal de que ya puede entrar un cliente nuevo. En el diagrama de estados se puede observar que el selector consiste de 3 estados, uno donde no hace nada, otro en donde se habilita el car wash y otro en donde se habilita el taller. En el car wash hay 5 estados que dependen del servicio seleccionado, hay un estado de lavado interno, otro de lavado externo, otro de cera y otro de silicón, el último estado es el que indica que ya el cliente terminó de ser atendido. En el taller hay 8 estados que dependen del servicio seleccionado, hay un estado de revisión de líquidos, otro de servicio a la caja, otro de llantas, otro de servicio al motor, otro de suspensión, otro de servicio a los frenos, y el último estado que indica que el cliente terminó de ser atendido. Los servicios disponibles en el car wash son: lavado, lavado externo, pulido, lavado completo y los servicios disponibles para el taller son: servicio completo, revisión de líquidos, caja y llantas, servicio de motor y revisión de suspensión y frenos. Se implementaron máquinas de Moore, las cuales hacen que las salidas solo dependan de los estados actuales, es decir, el input no afecta directamente a la salida. Los inputs de esta máquina consisten en la selección de servicio y un botón que tiene la función de avisar que ya esta terminada una parte del servicio, es decir, un botón para cambiar de estados.

2. Diagrama de cajas:

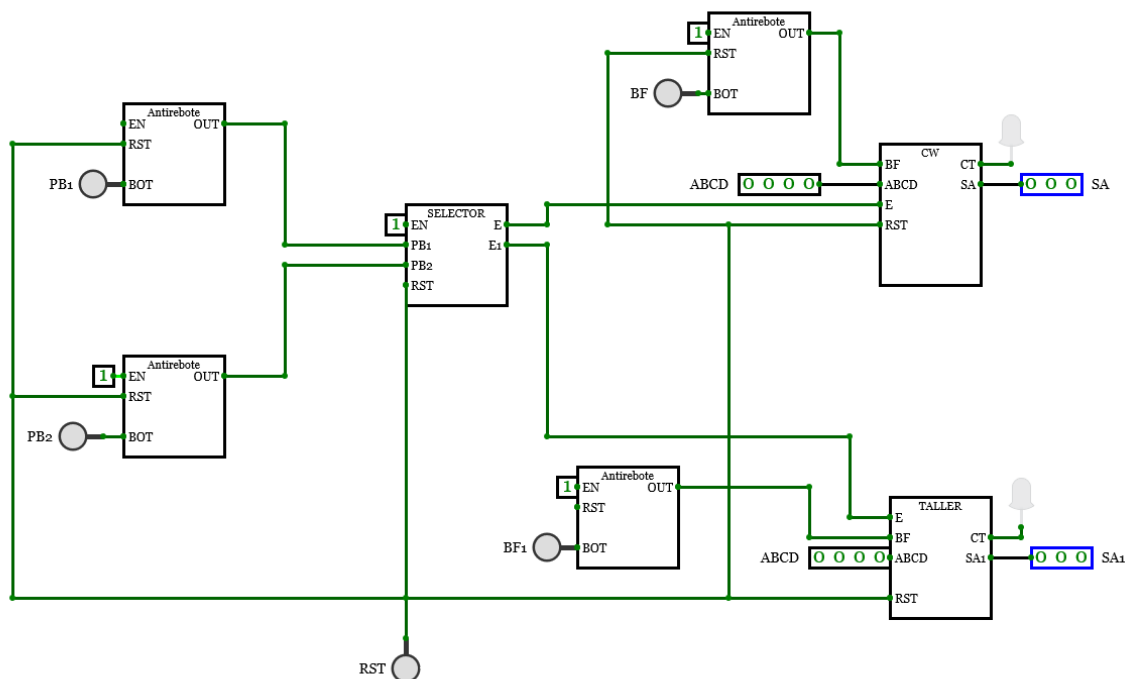


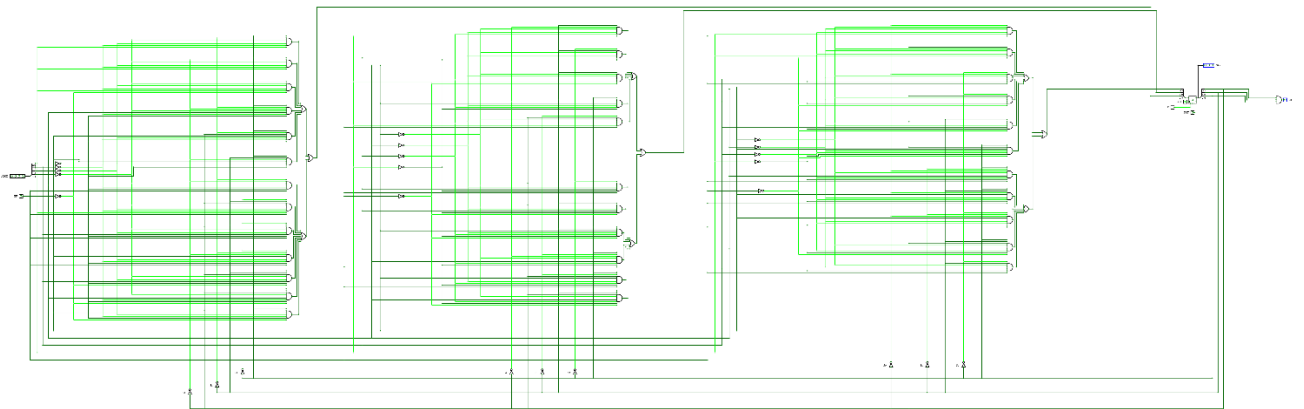
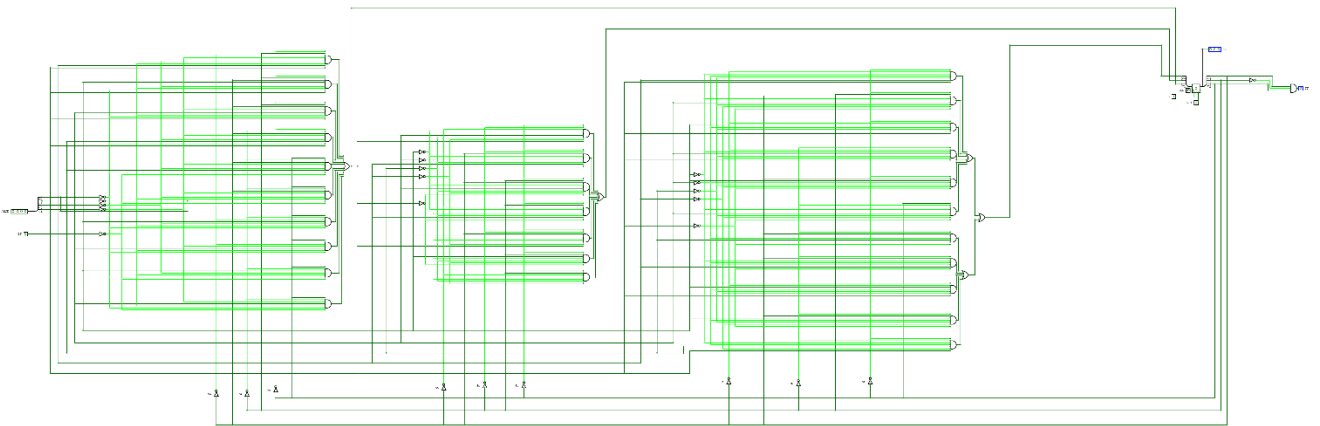
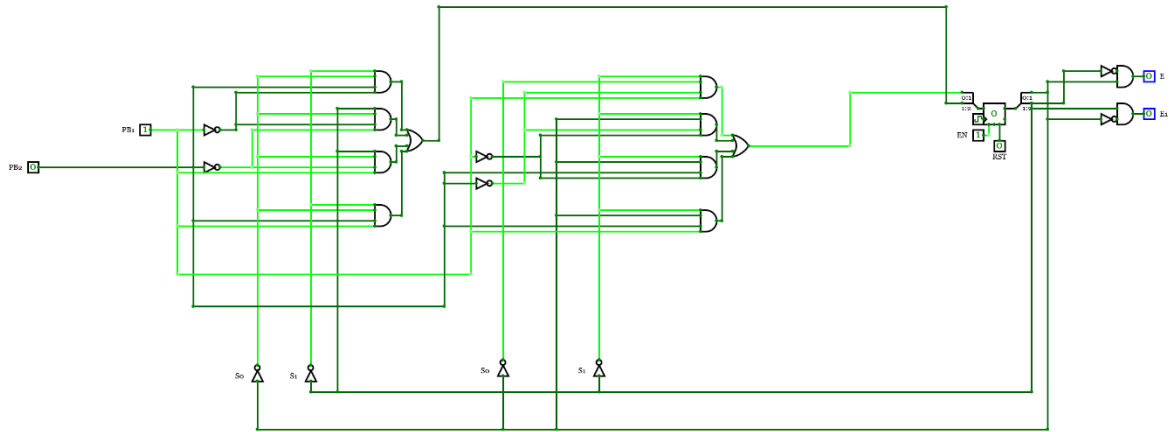
3. Diagramas de estado:

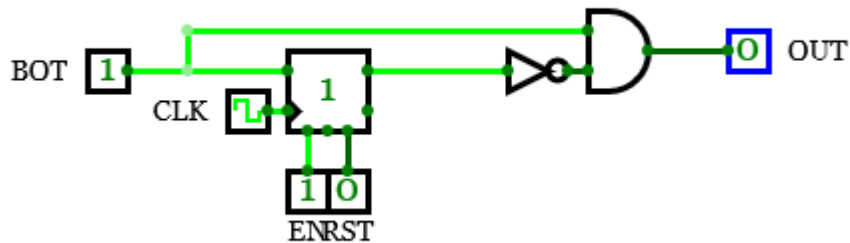




4.Simulación en CircuitVerse:







5.Ecuaciones Booleanas

Selector:

Car Wash:

- Estados Futuros:

Minimized:

$$\begin{aligned}
 SF2 &= S2' S1 S0' A' B' C' D BF + S2' S1 S0 A B' C' D' BF + S2' S1 S0 A' B C' D' BF \\
 &+ S2' S1' S0 A' B' C D' BF + S2 S1' S0 A' B' C D' BF' + S2 S1' S0 A' B' C' D BF' + \\
 &S2 S1' S0' A B' C' D' + S2 S1' S0' A' B C' D' + S2 S1' A B' C' D' BF' + S2 S1' \\
 &A' B C' D' BF'; \\
 SF1 &= S2' S1' S0' A' B C' D' BF + S2' S1' S0 A' B' C' D BF + S2' S1 S0 A' B C' D' \\
 &BF' + S2' S1 S0' A' B' C' D BF' + S2' S1' S0 A B' C' D' BF + S2' S1 A B' C' D' BF' \\
 &+ S2' S1 S0' A B' C' D'; \\
 SF0 &= S2' S0' A' B' C' D BF + S2' S1 S0 A' B C' D' BF' + S2' S0' A B' C' D' BF + \\
 &S1' S0' A' B C' D' BF + S2' S0 A B' C' D' BF' + S2 S1' S0 A' B C' D' BF' + S1' S0 \\
 &A' B' C D' BF' + S1' S0 A' B' C' D BF' + S1' S0' A B' C' D' BF + S1' S0 A B' C' D' \\
 &BF' + S2' S1' A' B' C D' BF;
 \end{aligned}$$

- Salidas:

$$CT = S2 S1' S0;$$

Taller:

- **Estados Futuros:**

Minimized:

SF2 = S2 S1' A' B' C' D + S2 S0' A' B' C' D + S2 A' B' C' D BF1' + S2 S1' S0' A' B C' D' + S2 S1' S0 A B' C' D' + S2 S1 S0' A B' C' D' + S1' S0' A' B C' D' BF1 + S2' S1 S0 A' B' C' D BF1 + S2' S1 S0 A' B' C' D' BF1 + S2' S1' S0' A B' C' D' BF1 + S2 S1 S0 A' B' C D' BF1' + S2 S1 S0 A' B C' D' BF1' + S2 S0 A B' C' D' BF1';

SF1 = S1 S0' A' B' C' D + S2' S1 A' B' C D' + S1 A' B' C' D BF1' + S2 S1 S0' A B' C' D' + S1' S0 A' B' C' D BF1 + S2' S0 A' B' C D' BF1 + S1 S0 A' B' C D' BF1' + S2 S1 A B' C' D' BF1' + S2 S1' S0' A' B C' D' BF1 + S2 S1' S0 A B' C' D' BF1 + S2 S1 S0 A' B C' D' BF1';

SF0 = S0' A' B' C' D BF1 + S0 A' B' C' D BF1' + S2' S0' A' B' C D' BF1 + S2' S0 A' B' C D' BF1' + S1 S0 A' B' C D' BF1' + S2 S0 A B' C' D' BF1' + S2 S1' S0' A' B C' D' BF1 + S2 S1 S0' A B' C' D' BF1 + S2' S1' S0' A B' C' D' BF1 + S2 S1 S0 A' B C' D' BF1' + S2' S1 A' B' C D' BF1;

- **Salidas:**

CT1 = S2 S1 S0;

Tablas de estados de Moore:

Selector:

Estado actual	Input		Estado futuro
	PB1	PB2	
S0	0	0	S0
S0	1	0	S1
S0	0	1	S2
S1	0	0	S1
S1	1	0	S0
S2	0	0	S2
S2	0	1	S0

Salidas:

Estado	E1	E0
S0	0	0
S1	0	1
S2	1	0

Servicios:

Lavado	0001
Lavado externo	0010
Pulido	0100
Lavado completo	1000

Servicio completo	0001
Líquidos, caja y llantas	0010
Motor	0100
Suspensión y frenos	1000

Car wash:

Estado actual	Input					Estado futuro
	Servicio				BF	
S0	0	0	0	1	0	S0
S0	0	0	0	1	1	S1
S0	0	0	1	0	0	S0
S0	0	0	1	0	1	S1
S0	0	1	0	0	0	S0
S0	0	1	0	0	1	S3
S0	1	0	0	0	0	S0
S0	1	0	0	0	1	S1
S1	0	0	0	1	0	S1
S1	0	0	0	1	1	S2
S1	0	0	1	0	0	S1
S1	0	0	1	0	1	S5
S1	1	0	0	0	0	S1
S1	1	0	0	0	1	S2
S2	0	0	0	1	0	S2
S2	0	0	0	1	1	S5
S2	1	0	0	0	0	S2
S2	1	0	0	0	1	S3
S3	0	1	0	0	0	S3
S3	0	1	0	0	1	S4
S3	1	0	0	0	0	S3
S3	1	0	0	0	0	S4
S4	0	1	0	0	0	S4
S4	0	1	0	0	1	S5
S4	1	0	0	0	0	S4
S4	1	0	0	0	1	S5
S5	0	0	0	1	0	S5

S5	0	0	0	1	1	S0
S5	0	0	1	0	0	S5
S5	0	0	1	0	1	S0
S5	0	1	0	0	0	S5
S5	0	1	0	0	1	S0
S5	1	0	0	0	0	S5
S5	1	0	0	0	1	S0

Salidas:

Estado	CT
S0	0
S1	0
S2	0
S3	0
S4	0
S5	1

Taller:

Estado actual	Input					Estado futuro
	Servicio			BF		
S0	0	0	0	1	0	S0
S0	0	0	0	1	1	S1
S0	0	0	1	0	0	S0
S0	0	0	1	0	1	S1
S0	0	1	0	0	0	S0
S0	0	1	0	0	1	S4
S0	1	0	0	0	0	S0
S0	1	0	0	0	1	S5
S1	0	0	0	1	0	S1
S1	0	0	0	1	1	S2
S1	0	0	1	0	0	S1
S1	0	0	1	0	1	S2
S2	0	0	0	1	0	S2
S2	0	0	0	1	1	S3

S2	0	0	1	0	0	S2
S2	0	0	1	0	1	S3
S3	0	0	0	1	0	S3
S3	0	0	0	1	1	S4
S3	0	0	1	0	0	S3
S3	0	0	1	0	0	S7
S4	0	0	0	1	0	S4
S4	0	0	0	1	1	S5
S4	0	1	0	0	0	S4
S4	0	1	0	0	1	S7
S5	0	0	0	1	0	S5
S5	0	0	0	1	1	S6
S5	1	0	0	0	0	S5
S5	1	0	0	0	1	S6
S6	0	0	0	1	0	S6
S6	0	0	0	1	1	S7
S6	1	0	0	0	0	S6
S6	1	0	0	0	1	S7
S7	0	0	0	1	0	S7
S7	0	0	0	1	1	S0
S7	0	0	1	0	0	S7
S7	0	0	1	0	1	S0
S7	0	1	0	0	0	S7
S7	0	1	0	0	1	S0
S7	1	0	0	0	0	S7
S7	1	0	0	0	1	S0

Salida:

Estado	CT1
S0	0
S1	0
S2	0
S3	0
S4	0
S5	0
S6	0
S7	1

Tablas de estados de Moore codificadas

Selector:

Estado actual	Input		Estado futuro
	PB1	PB2	
00	0	0	00
00	1	0	01
00	0	1	10
01	0	0	01
01	1	0	00
10	0	0	10
10	0	1	0

Salidas:

Estado	E1	E0
00	0	0
01	0	1
10	1	0

Car wash:

Estado actual	Input					Estado futuro
	Servicio				BF	
000	0	0	0	1	0	000
000	0	0	0	1	1	001
000	0	0	1	0	0	000
000	0	0	1	0	1	001
000	0	1	0	0	0	000
000	0	1	0	0	1	011

000	1	0	0	0	0	000
000	1	0	0	0	1	001
001	0	0	0	1	0	001
001	0	0	0	1	1	010
001	0	0	1	0	0	001
001	0	0	1	0	1	101
001	1	0	0	0	0	001
001	1	0	0	0	1	010
010	0	0	0	1	0	010
010	0	0	0	1	1	101
010	1	0	0	0	0	010
010	1	0	0	0	1	011
011	0	1	0	0	0	011
011	0	1	0	0	1	100
011	1	0	0	0	0	011
011	1	0	0	0	0	100
100	0	1	0	0	0	100
100	0	1	0	0	1	101
100	1	0	0	0	0	100
100	1	0	0	0	1	101
101	0	0	0	1	0	101
101	0	0	0	1	1	000
101	0	0	1	0	0	101
101	0	0	1	0	1	000
101	0	1	0	0	0	101
101	0	1	0	0	1	000
101	1	0	0	0	0	101
101	1	0	0	0	1	000

Salidas:

Estado	CT
000	0
001	0
010	0
011	0
100	0
101	1

Taller:

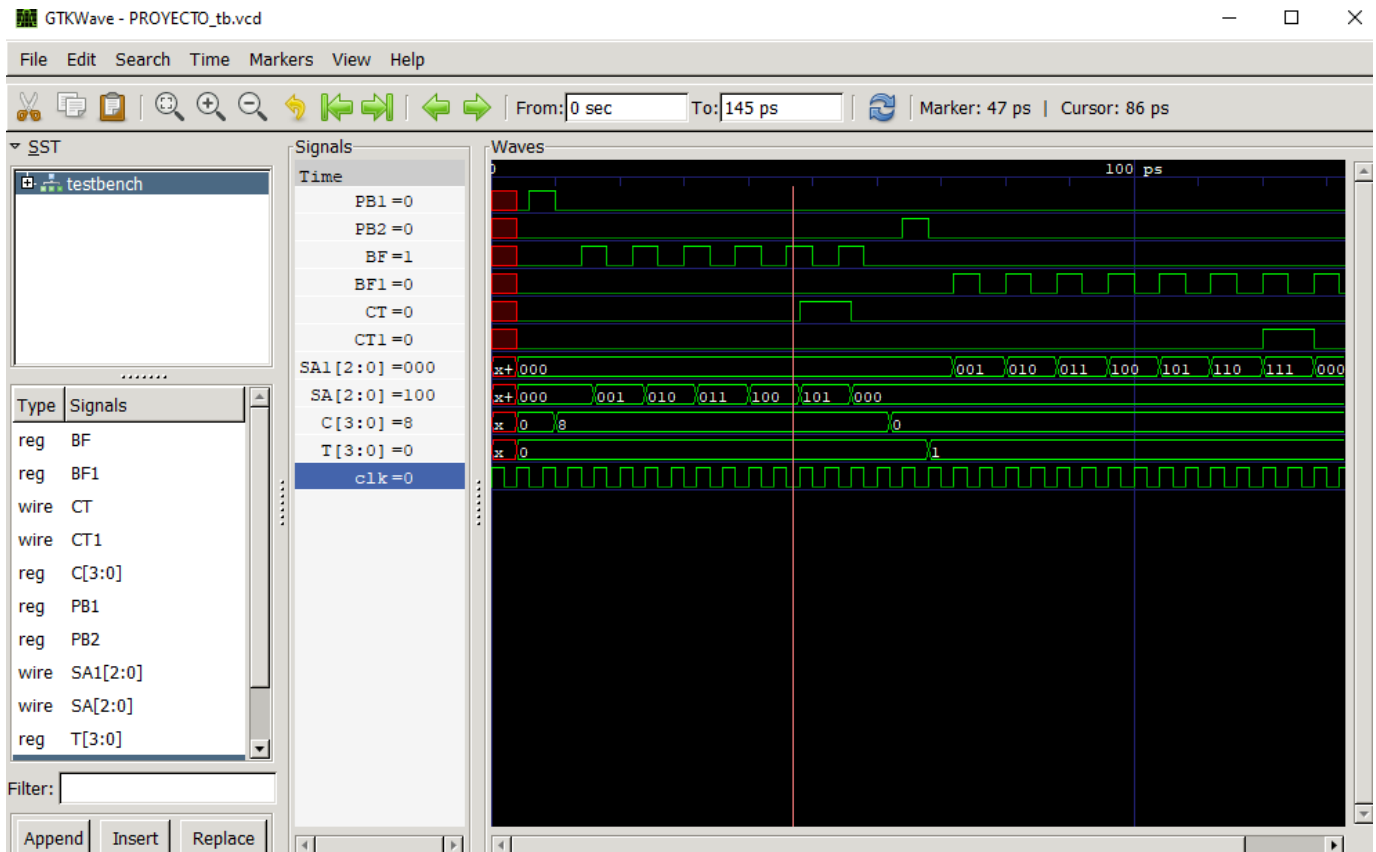
Estado actual	Input					Estado futuro
	Servicio				BF	
000	0	0	0	1	0	000
000	0	0	0	1	1	001
000	0	0	1	0	0	000
000	0	0	1	0	1	001
000	0	1	0	0	0	000
000	0	1	0	0	1	100
000	1	0	0	0	0	000
000	1	0	0	0	1	101
001	0	0	0	1	0	001
001	0	0	0	1	1	010
001	0	0	1	0	0	001
001	0	0	1	0	1	010
010	0	0	0	1	0	010
010	0	0	0	1	1	011
010	0	0	1	0	0	010
010	0	0	1	0	1	011
011	0	0	0	1	0	011
011	0	0	0	1	1	100
011	0	0	1	0	0	011
011	0	0	1	0	0	111
100	0	0	0	1	0	100
100	0	0	0	1	1	101
100	0	1	0	0	0	100
100	0	1	0	0	1	111
101	0	0	0	1	0	101
101	0	0	0	1	1	110

101	1	0	0	0	0	101
101	1	0	0	0	1	110
110	0	0	0	1	0	110
110	0	0	0	1	1	111
110	1	0	0	0	0	110
110	1	0	0	0	1	111
111	0	0	0	1	0	111
111	0	0	0	1	1	000
111	0	0	1	0	0	111
111	0	0	1	0	1	000
S7	0	1	0	0	0	111
S7	0	1	0	0	1	000
S7	1	0	0	0	0	111
S7	1	0	0	0	1	000

Salida:

Estado	CT1
000	0
001	0
010	0
011	0
100	0
101	0
110	0
111	1

Diagrama de timing:



Verilog

```
1 //Proyecto
2 //Flip flop tipo D de 3 bits, reset asíncrono
3 module FlipF3(input wire clk, reset, E,
4               input wire [2:0]D,
5               output reg [2:0]Q);
6 always @ (posedge clk or posedge reset)begin
7     if (reset) begin
8         Q <= 3'b0;
9     end
10    else if (E) begin
11        Q <= D;
12    end
13 end
14 endmodule
15
16 //Flip flop tipo D de 2 bits, reset asíncrono
17 module FlipF2(input wire clk, reset, E,
18               input wire [1:0]D,
19               output reg [1:0]Q);
20 always @ (posedge clk or posedge reset)begin
21     if (reset) begin
22         Q <= 2'b0;
23     end
24     else if (E) begin
25         Q <= D;
26     end
27 end
28 endmodule
29
30 //Flip flop tipo D de 1 bit, reset asíncrono
```

```

29
30 //Flip flop tipo de de 1 bit, reset asíncrono
31 module FlipF1(input wire clk, reset,E,
32               input wire D,
33               output reg Q);
34 always @ (posedge clk or posedge reset)begin
35     if (reset) begin
36         Q <= 1'b0;
37     end
38     else if (E) begin
39         Q <= D;
40     end
41 end
42
43 endmodule

```

```

44
45 //Antirebote
46 module antireb (input wire clk, reset, PB, output wire Q);
47 wire Y1, Y2;
48 assign Y2 = (PB);
49 FlipF1 F1(clk, reset, 1'b1 , Y2, Y1);
50 assign Q = (~Y1 & PB);
51 endmodule
52
53 //selector de Local
54 module selector(input wire reset, clk, PB1, PB2, output wire E, E1);
55 wire s0, s1;
56 wire [1:0]s;
57 wire [1:0]sf;
58 assign s1 = s[1];
59 assign s0 = s[0];
60 assign sf[1] = (~s1 & ~s0 & PB2 & ~PB1) | (s1 & ~s0 & ~PB2 & ~ PB1) | (s1 & ~s0 & ~PB2 & PB1) | (s1 & ~s0 & PB2 & PB1);
61 assign sf[0] = (~s1 & ~s0 & ~PB2 & PB1) | (~s1 & s0 & ~PB2 & ~PB1) | (~s1 & s0 & PB2 & PB1) | (~s1 & s0 & PB2 & PB1);
62 FlipF2 t1(clk, reset, 1'b1, sf, s);
63 assign E = (~s1 & s0);
64 assign E1 = (s1 & ~s0);
65 endmodule
66

```

```

66 //modulo del car carwash
67 module carwash(input wire reset, clk, EN, input wire[3:0]L, input wire BF, output wire CT0, output wire [2:0]SA);
68 wire sc0, sc1, sc2;
69 wire [2:0]s_c;
70 wire [2:0]sf_c;
71 assign sc2 = s_c[2];
72 assign sc1 = s_c[1];
73 assign sc0 = s_c[0];
74
75 assign sf_c[2] = (~sc2 & sc1 & ~sc0 & ~L[3] & ~L[2] & ~L[1] & L[0] & BF) | (~sc2 & sc1 & sc0 & L[3] & ~L[2] & ~L[1] & ~L[0] & BF) | (~sc2 & sc1 & sc0 & ~L[3] & L[2] & ~L[1] & ~L[0] & BF) |
76 (~sc2 & ~sc1 & sc0 & ~L[3] & ~L[2] & L[1] & ~L[0] & BF) | (sc2 & ~sc1 & sc0 & ~L[3] & ~L[2] & L[1] & ~L[0] & ~BF) | (sc2 & ~sc1 & sc0 & ~L[3] & ~L[2] & ~L[1] & L[0] & ~BF) |
77 (sc2 & ~sc1 & ~sc0 & L[3] & ~L[2] & ~L[1] & ~L[0]) | (sc2 & ~sc1 & ~sc0 & ~L[3] & L[2] & ~L[1] & ~L[0]) | (sc2 & ~sc1 & L[3] & ~L[2] & ~L[1] & ~L[0] & ~BF) |
78 (sc2 & ~sc1 & ~L[3] & L[2] & ~L[1] & ~L[0] & ~BF);
79
80 assign sf_c[1] = (~sc2 & ~sc1 & ~sc0 & ~L[3] & L[2] & ~L[1] & ~L[0] & BF) | (~sc2 & ~sc1 & sc0 & ~L[3] & ~L[2] & ~L[1] & L[0] & BF) | (~sc2 & sc1 & sc0 & ~L[3] & L[2] & ~L[1] & ~L[0] & ~BF) |
81 (~sc2 & sc1 & ~sc0 & ~L[3] & ~L[2] & ~L[1] & L[0] & ~BF) | (~sc2 & ~sc1 & sc0 & L[3] & ~L[2] & ~L[1] & ~L[0] & BF) | (~sc2 & sc1 & L[3] & ~L[2] & ~L[1] & ~L[0] & ~BF) |
82 (~sc2 & sc1 & ~sc0 & L[3] & ~L[2] & ~L[1] & ~L[0]);
83
84 assign sf_c[0] = (~sc2 & ~sc0 & ~L[3] & ~L[2] & ~L[1] & L[0] & BF) | (~sc2 & sc1 & sc0 & ~L[3] & L[2] & ~L[1] & ~L[0] & ~BF) | (~sc2 & ~sc0 & L[3] & ~L[2] & ~L[1] & ~L[0] & BF) |
85 (~sc1 & ~sc0 & ~L[3] & L[2] & ~L[1] & ~L[0] & BF) | (~sc2 & sc0 & L[3] & ~L[2] & ~L[1] & ~L[0] & ~BF) | (sc2 & ~sc1 & sc0 & ~L[3] & L[2] & ~L[1] & ~L[0] & ~BF) |
86 (~sc1 & sc0 & ~L[3] & ~L[2] & L[1] & ~L[0] & ~BF) | (~sc1 & sc0 & ~L[3] & ~L[2] & ~L[1] & L[0] & ~BF) | (~sc1 & ~sc0 & L[3] & ~L[2] & ~L[1] & ~L[0] & BF) |
87 (~sc1 & sc0 & L[3] & ~L[2] & ~L[1] & ~L[0] & ~BF) | (~sc2 & ~sc1 & ~L[3] & ~L[2] & L[1] & ~L[0] & BF);
88
89 FlipF3 t2(clk, reset, EN, sf_c, s_c);
90 assign CT0 = (sc2 & ~sc1 & sc0);
91 assign SA = {sc2, sc1, sc0};
92 endmodule
93
94

```

```

//modulo del taller
module taller(input wire reset, clk, EN, input wire [3:0]T, input wire BF1, output wire CT1, output wire [2:0]SA1);
wire st0, st1, st2;
wire [2:0]s_t;
wire [2:0]sf_t;
assign st2 = s_t[2];
assign st1 = s_t[1];
assign st0 = s_t[0];
assign sf_t[2] = (st2 & ~st1 & ~T[3] & ~T[2] & ~T[1] & T[0]) | (st2 & ~st0 & ~T[3] & ~T[2] & ~T[1] & T[0]) | (st2 & ~T[3] & ~T[2] & ~T[1] & T[0] & ~BF1) | (st2 & ~st1 & ~st0 & ~T[3] & T[2] & ~T[1] & ~T[0]) |
(st2 & ~st1 & st0 & T[3] & ~T[2] & ~T[1] & ~T[0]) | (st2 & st1 & ~st0 & T[3] & ~T[2] & ~T[1] & ~T[0]) | (~st1 & ~st0 & ~T[3] & T[2] & ~T[1] & ~T[0] & BF1) |
(~st2 & st1 & st0 & ~T[3] & ~T[2] & ~T[1] & ~T[0] & BF1) | (~st2 & st1 & st0 & ~T[3] & ~T[2] & T[1] & ~T[0] & BF1) | (~st2 & ~st1 & ~st0 & T[3] & ~T[2] & ~T[1] & ~T[0] & BF1) |
(st2 & st1 & st0 & ~T[3] & T[2] & ~T[1] & ~T[0] & ~BF1) | (st2 & st0 & T[3] & ~T[2] & ~T[1] & ~T[0] & ~BF1);
assign sf_t[1] = (st1 & ~st0 & ~T[3] & ~T[2] & ~T[1] & T[0]) | (~st2 & st1 & ~T[3] & ~T[2] & T[1] & ~T[0]) | (st1 & ~T[3] & ~T[2] & ~T[1] & T[0] & ~BF1) | (st2 & st1 & ~st0 & T[3] & ~T[2] & ~T[1] & ~T[0]) |
(~st1 & st0 & ~T[3] & ~T[2] & ~T[1] & T[0] & BF1) | (~st2 & st0 & ~T[3] & ~T[2] & T[1] & ~T[0] & BF1) | (st1 & st0 & ~T[3] & ~T[2] & T[1] & ~T[0] & ~BF1) | (st2 & st1 & T[3] & ~T[2] & ~T[1] & ~T[0] & ~BF1) |
(st2 & ~st1 & ~st0 & ~T[3] & T[2] & ~T[1] & ~T[0] & BF1) | (st2 & ~st1 & st0 & T[3] & ~T[2] & ~T[1] & ~T[0] & BF1) |
(st2 & st1 & st0 & ~T[3] & T[2] & ~T[1] & ~T[0] & ~BF1);
assign sf_t[0] = (~st0 & ~T[3] & ~T[2] & ~T[1] & T[0] & BF1) | (st0 & ~T[3] & ~T[2] & ~T[1] & T[0] & ~BF1) | (~st2 & ~st0 & ~T[3] & ~T[2] & T[1] & ~T[0] & BF1) | (~st2 & st0 & ~T[3] & ~T[2] & T[1] & ~T[0] & ~BF1) |
(st1 & st0 & ~T[3] & ~T[2] & T[1] & ~T[0] & ~BF1) | (st2 & st0 & T[3] & ~T[2] & ~T[1] & ~T[0] & ~BF1) | (st2 & ~st1 & ~st0 & ~T[3] & T[2] & ~T[1] & ~T[0] & BF1) | (st2 & st1 & ~st0 & T[3] & ~T[2] & ~T[1] & ~T[0] & BF1) |
(~st2 & ~st1 & ~st0 & T[3] & ~T[2] & ~T[1] & ~T[0] & BF1) | (st2 & st1 & st0 & ~T[3] & T[2] & ~T[1] & ~T[0] & ~BF1) | (~st2 & st1 & ~T[3] & ~T[2] & T[1] & ~T[0] & BF1);
FlipF3 t3(clk, reset, EN, sf_t, s_t);
assign CT1 = (st2 & st1 & st0);
assign SA1 = {st2, st1, st0};
endmodule

```

```

121 //modulo donde se unen todas las fsm
122 module megafsm(input wire reset, clk, PB1, PB2, BF, BF1, input wire [3:0]T, input wire[3:0]C, output wire CT, CT1, output wire [2:0]SA, output wire [2:0]SA1);
123
124
125 wire C1, C2, T1, T2, E, E1;
126
127 antireb a1(clk, reset, PB1, C1);
128 antireb a2(clk, reset, PB2, T1);
129 antireb a3(clk, reset, BF, C2);
130 antireb a4(clk, reset, BF1, T2);
131
132
133 selector m1(reset, clk, C1, T1, E, E1);
134 carwash m2(reset, clk, E, C, C2, CT, SA);
135 taller m3(reset, clk, E1, T, T2, CT1, SA1);
136
137 endmodule

```