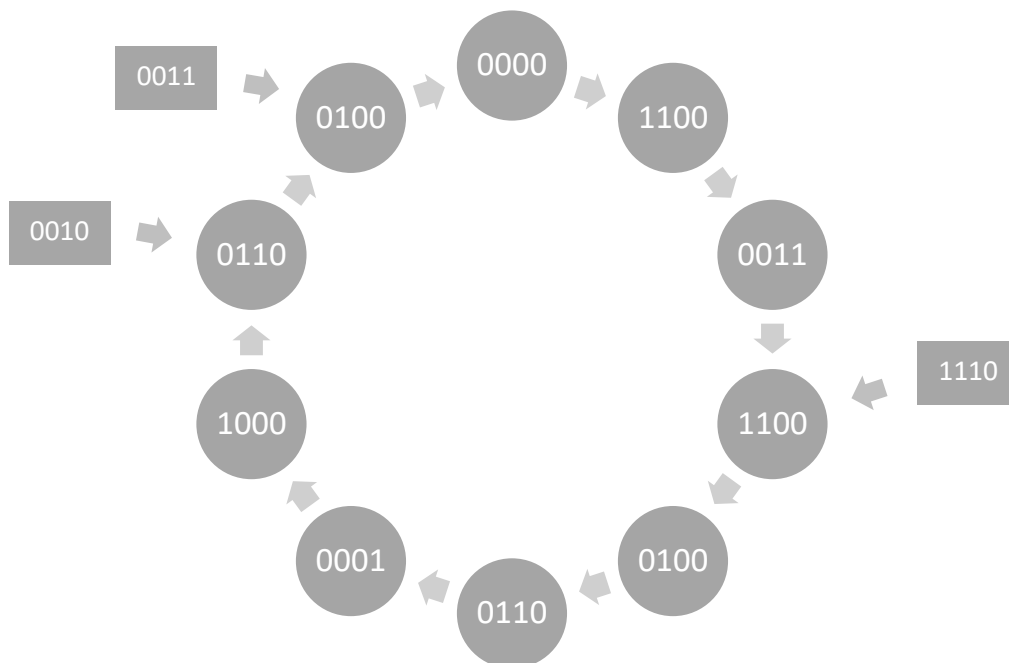


CONTADOR ARBITRARIO

1) Análisis del ejercicio a realizar

Para realizar un contador arbitrario analizaremos la secuencia dada que es la siguiente: **0-12-3-12-4-6-1-8-6-4**. Con esa secuencia no podremos realizar correctamente el contador ya que se repiten algunos números dentro de la secuencia, por tanto, deberemos cambiar algunos de ellos utilizando un conversor y utilizar inicialmente esta secuencia: **0-12-3-14-4-6-1-8-2-5**.



A la hora de realizar un contador arbitrario de 4 bits, tenemos que realizar una tabla de transiciones, de tal forma que los números de la secuencia vayan a los números asignados por nosotros.

Transiciones								J3	K3	J2	K2	J1	K1	J0	K0
Estado actual				Estado Siguiente											
Q3	Q2	Q1	Q0	Q'3	Q'2	Q'1	Q'0								
0	0	0	0	1	1	0	0	1	X	1	X	0	X	0	X
0	0	0	1	1	0	0	0	1	X	0	X	0	X	X	1
0	0	1	0	0	1	0	1	0	X	1	X	X	1	1	X
0	0	1	1	1	1	1	0	1	X	1	X	X	0	X	1
0	1	0	0	0	1	1	0	0	X	X	0	1	X	0	X
0	1	0	1	0	0	0	0	0	X	X	1	0	X	X	1
0	1	1	0	0	0	0	1	0	X	X	1	X	1	1	X
0	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X
1	0	0	0	0	0	1	0	X	1	0	X	1	X	0	X
1	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X
1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X
1	1	0	0	0	0	1	1	X	1	X	1	1	X	1	X
1	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	0	0	1	0	0	X	1	X	0	X	1	0	X
1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X

2) Obtención de las ecuaciones para el contador [Mapas de Karnaugh]

Para obtener las ecuaciones que debemos más tarde implantar en verilog debemos antes realizar algunos mapas de Karnaugh para que una vez las tengamos, nos sirvan para implantar las puertas del módulo ContadorArbitrario.

$$J3 = \overline{Q1} * \overline{Q2} + \overline{Q2} * Q0$$

Q3 Q2 →	00	01	11	10
Q1 Q0 ↓				
00	1	0	X	X
01	1	0	X	X
11	1	X	X	X
10	0	0	X	X

Q3 Q2 →	00	01	11	10
Q1 Q0 ↓				
00	1	0	X	X
01	1	0	X	X
11	1	X	X	X
10	0	0	X	X

$$K3 = 1$$

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	X	X	1	1
01	X	X	X	X
11	X	X	X	X
10	X	X	1	X

$$J2 = Q1 * \overline{Q3} + \overline{Q0}$$

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	1	X	X	0
01	0	X	X	X
11	1	X	X	X
10	1	X	X	X

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	1	X	X	0
01	0	X	X	X
11	1	X	X	X
10	1	X	X	X

$$K2 = Q3 * \overline{Q1} + Q0 + \overline{Q3} * Q1$$

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	X	0	1	X
01	X	1	X	X
11	X	X	X	X
10	X	1	0	X

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	X	0	1	X
01	X	1	X	X
11	X	X	X	X
10	X	1	0	X

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	X	0	1	X
01	X	1	X	X
11	X	X	X	X
10	X	1	0	X

$$J1 = Q3 + \overline{Q0} * Q2$$

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	0	1	1	1
01	0	0	X	X
11	X	X	X	X
10	X	X	X	X

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	0	1	1	1
01	0	0	X	X
11	X	X	X	X
10	X	X	X	X

$$K1 = \overline{Q0}$$

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	0	X	X	X
10	1	1	1	X

$$J0 = Q3 * Q2 * \overline{Q1} + \overline{Q3} * Q1$$

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	0	0	1	0
01	X	X	X	X
11	X	X	X	X
10	1	1	0	X

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	0	0	1	0
01	X	X	X	X
11	X	X	X	X
10	1	1	0	X

$$K0 = 1$$

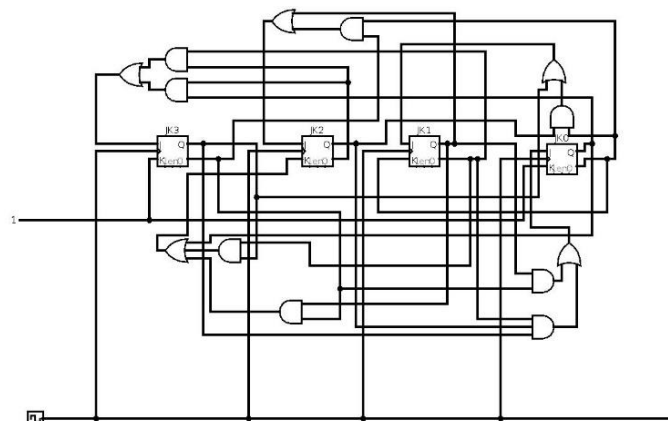
Q3 Q2 →	00	01	11	10
Q1 Q0 ↓				
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

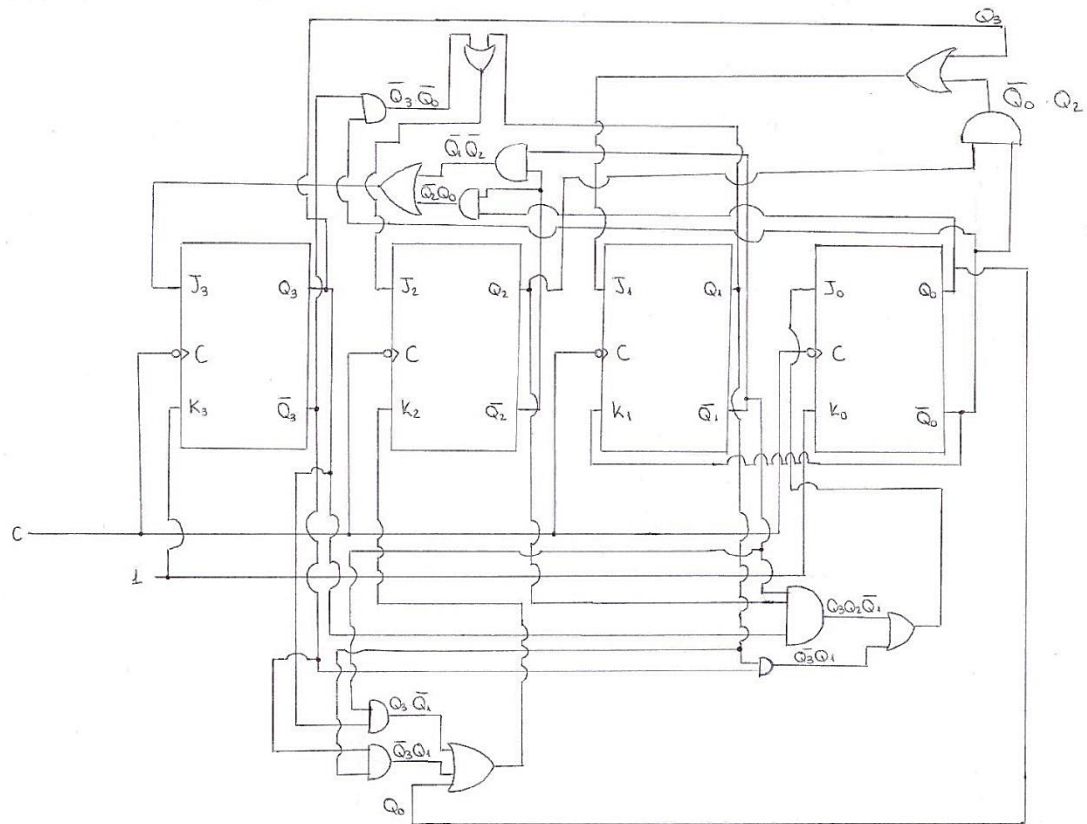
3) Comprobación de las ecuaciones

En este paso deberemos comprobar que de verdad con las ecuaciones obtenidas llevan a los números que nosotros queremos.

Estado Inicial				J3	K3	J2	K2	J1	K1	J0	K0	Estado Final			
Q3	Q2	Q1	Q0									Q'3	Q'2	Q'1	Q'0
0	0	0	0	1	1	1	0	0	1	0	1	1	1	0	0
0	0	0	1	1	1	0	1	0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	1	0	1	1	1	0	1	0	1
0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	0
0	1	0	0	0	1	1	0	1	1	0	1	0	1	1	0
0	1	0	1	0	1	0	1	0	0	0	1	0	0	0	0
0	1	1	0	0	1	1	1	1	1	1	1	0	0	0	1
0	1	1	1	0	1	1	1	0	0	1	1	0	0	0	0
1	0	0	0	1	1	0	1	1	1	0	1	0	0	1	0
1	0	0	1	1	1	0	1	1	0	0	1	0	0	1	0
1	0	1	0	0	1	1	0	1	1	0	1	0	1	0	0
1	0	1	1	1	1	1	1	1	0	0	1	0	0	1	0
1	1	0	0	0	1	0	1	1	1	1	1	0	0	1	1
1	1	0	1	0	1	0	1	1	0	1	1	0	0	1	0
1	1	1	0	0	1	1	0	1	1	0	1	0	1	0	0
1	1	1	1	0	1	1	1	1	0	0	1	0	0	1	0

4) Esquema del módulo ContadorArbitrario con sus puertas





5) Obtención de las ecuaciones para el contador [Mapas de Karnaugh]

Para realizar el conversor (nosotros lo hemos llamado Transformador) deberemos realizar una tabla con las entradas y las salidas del conversor. Mas tarde tendremos que hacer los correspondientes mapas de Karnaugh para conseguir las ecuaciones que tenemos que implementar con puertas lógicas.

Entrada				Salida Transformador			
Q3	Q2	Q1	Q0	O3	O2	O1	O0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	1	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	1
1	1	0	0	1	1	0	0
1	1	0	1	1	1	0	1
1	1	1	0	1	1	0	0
1	1	1	1	1	1	1	1

$$O3 = Q3$$

Q3 Q2 →	00	01	11	10
Q1 Q0 ↓				
00	0	0	1	1
01	0	0	1	1
11	0	0	1	1
10	0	0	1	1

$$O2 = Q2 + Q1 * \overline{Q0} * \overline{Q3}$$

Q3 Q2 →	00	01	11	10
Q1 Q0 ↓				
00	0	1	1	0
01	0	1	1	0
11	0	1	1	0
10	1	1	1	0

Q3 Q2 →	00	01	11	10
Q1 Q0 ↓				
00	0	1	1	0
01	0	1	1	0
11	0	1	1	0
10	1	1	1	0

$$Q1 = Q1 * \overline{Q3} + Q1 * \overline{Q2} + Q1 * Q0$$

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	0	1

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	0	1

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	0	1

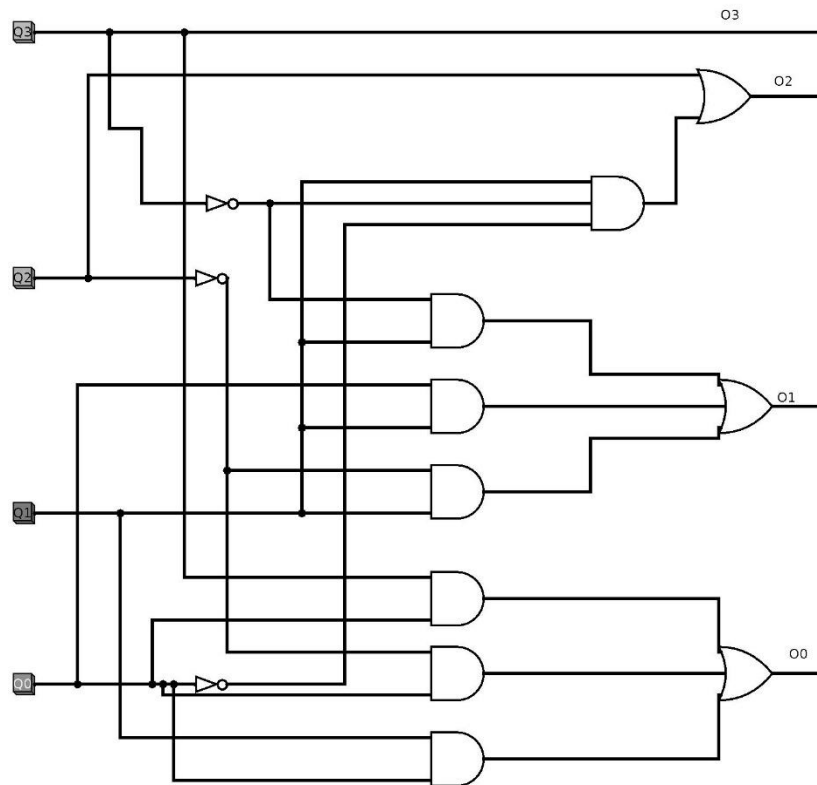
$$Q0 = Q0 * \overline{Q2} + Q0 * Q3 + Q1 * Q0$$

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	0	0	0	0
01	1	0	1	1
11	1	1	1	1
10	0	0	0	0

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	0	0	0	0
01	1	0	1	1
11	1	1	1	1
10	0	0	0	0

Q3 Q2 → Q1 Q0 ↓	00	01	11	10
00	0	0	0	0
01	1	0	1	1
11	1	1	1	1
10	0	0	0	0

6) Esquema del módulo Transformador con sus puertas



7) Código de Verilog

```

module BiestableJK(output reg Q, output wire nQ, input wire J, input wire K, input wire C);
  not(nQ,Q);
  initial
  begin
    Q=0;
  end
  always @(posedge C)
  case ({J,K})
    2'b10: Q=1;
    2'b01: Q=0;
    2'b11: Q=~Q;
  endcase
endmodule

```

```

module ContadorArbitrario(output wire [3:0]Q, input wire C);
  wire[3:0]nQ;
  wire J3, J3a, J3b, J2, J2a;

  // Puertas logicas para J3
  and (J3a, nQ[1], nQ[2]);
  and (J3b, nQ[2], Q[0]);
  or (J3, J3a, J3b);

```

```
// Puertas logicas para J2
and(J2a, nQ[3], nQ[0]);
or(J2, J2a, Q[1]);

// Puertas logicas para K2
and(K2a, Q[3], nQ[1]);
and(K2b, Q[1], nQ[3]);
or(K2, K2a, K2b, Q[0]);

// Puertas logicas para J1
and(J1a, nQ[0], Q[2]);
or(J1, J1a, Q[3]);

// Puertas logicas para J0
and(J0a, nQ[1], Q[2], Q[3]);
and(J0b, nQ[3], Q[1]);
or(J0, J0a, J0b);

BiestableJK JK3(Q[3], nQ[3], J3, 1'b1, C); //Aquí introducimos un 1 en K3
BiestableJK JK2(Q[2], nQ[2], J2, K2, C);
BiestableJK JK1(Q[1], nQ[1], J1, nQ[0], C); //Aquí introducimos nQ[0] para K1
BiestableJK JK0(Q[0], nQ[0], J0, 1'b1, C); //Aquí introducimos un 1 en K0

endmodule

module Transformador(input wire [3:0]Q, output wire [3:0] O);

    wire nq3, nq2, nq1, nq0;
    wire a1s,a2s,a3s,a4s,a5s,a6s,a7s;
    wire O2, O1, O0;

    // Aquí declaramos puertas not para obtener entradas negadas en el conversor
    not(nq3,Q[3]);
    not(nq2,Q[2]);
    not(nq1,Q[1]);
    not(nq0,Q[0]);

    // A cada entrada O le asignamos su salida con distintas puertas que hemos
    implementado

    assign O[3] = Q[3];

    and(a1s,Q[1],nq0,nq3);
    or(O[2], Q[2], a1s);

    and(a2s,Q[1],nq3);
    and(a3s,Q[1],nq2);
    and(a4s,Q[1],Q[0]);
    or(O[1], a2s, a3s, a4s);
```

```
        and(a5s,Q[0],nq2);
    and(a6s,Q[0],Q[3]);
    and(a7s,Q[1],Q[0]);
    or(O[0], a5s, a6s, a7s);

endmodule

module Test;
    reg C;
    wire [3:0]Q;
    wire[3:0]O;

    ContadorArbitrario CA(Q,C);
    Transformador Transf(Q,O);

    always #10 C=~C;
    initial
    begin
        $dumpfile("salida.dmp");
        $dumpvars(2, CA, Transf);
        $dumpon;
        $monitor($time," C: %b , Q: %b (%d) || O: %b (%d)" ,C, Q, Q, O,O);
        C=1;
        #250;

        $dumpoff;
        $finish;
    end

endmodule

endmodule
```

8) Pruebas de Verilog

En verilog sacaremos a través de la terminal con display la secuencia sin repetir y la secuencia inicial. Además, mediante gtkwave podremos ver el cronograma de la secuencia y de esa forma vemos que todo está correcto y cada número va a lo que hemos asignado inicialmente.

Anexo – Fotos del Terminal y gtkwave

Secuencia Inicial
Secuencia final

0	C: 1	, Q: 0000 (0)		O: 0000 (0)
10	C: 0	, Q: 0000 (0)		O: 0000 (0)
20	C: 1	, Q: 1100 (12)		O: 1100 (12)
30	C: 0	, Q: 1100 (12)		O: 1100 (12)
40	C: 1	, Q: 0011 (3)		O: 0011 (3)
50	C: 0	, Q: 0011 (3)		O: 0011 (3)
60	C: 1	, Q: 1110 (14)		O: 1100 (12)
70	C: 0	, Q: 1110 (14)		O: 1100 (12)
80	C: 1	, Q: 0100 (4)		O: 0100 (4)
90	C: 0	, Q: 0100 (4)		O: 0100 (4)
100	C: 1	, Q: 0110 (6)		O: 0110 (6)
110	C: 0	, Q: 0110 (6)		O: 0110 (6)
120	C: 1	, Q: 0001 (1)		O: 0001 (1)
130	C: 0	, Q: 0001 (1)		O: 0001 (1)
140	C: 1	, Q: 1000 (8)		O: 1000 (8)
150	C: 0	, Q: 1000 (8)		O: 1000 (8)
160	C: 1	, Q: 0010 (2)		O: 0110 (6)
170	C: 0	, Q: 0010 (2)		O: 0110 (6)
180	C: 1	, Q: 0101 (5)		O: 0100 (4)
190	C: 0	, Q: 0101 (5)		O: 0100 (4)
200	C: 1	, Q: 0000 (0)		O: 0000 (0)
210	C: 0	, Q: 0000 (0)		O: 0000 (0)
220	C: 1	, Q: 1100 (12)		O: 1100 (12)
230	C: 0	, Q: 1100 (12)		O: 1100 (12)
240	C: 1	, Q: 0011 (3)		O: 0011 (3)
250	C: 0	, Q: 0011 (3)		O: 0011 (3)

En esta foto podemos ver como la el contador realiza la secuencia con números repetidos y después con el conversor vuelve a realizar la secuencia inicial, es decir la secuencia con los números repetidos.

