



INSTITUTO POLITÉCNICO NACIONAL



ESCUELA SUPERIOR DE CÓMPUTO

INGENIERIA EN SISTEMAS COMPUTACIONALES

MATERIA: DISEÑO DE SISTEMAS DIGITALES

PROFESOR: TESTA NAVA ALEXIS

PRESENTA:

RAMIREZ BENITEZ BRAYAN

GRUPO: 2CV18

PRACTICA 6:

DETECTOR DE SECUENCIA CON MODELO DE MEALY

CIUDAD DE MEXICO MAYO DE 2021

Para esta práctica utilizaremos la GAL C22V10 en el Simulador Galaxy, esta práctica consistió en programar un detector de secuencia con modelo mealy haciendo uso de la metodología previamente vista en clase donde mostrara en un display una A si acepta la secuencia o una E en caso de no aceptarla, a continuación, las salidas y entradas que ocupamos de la GAL y el código correspondiente para la práctica.

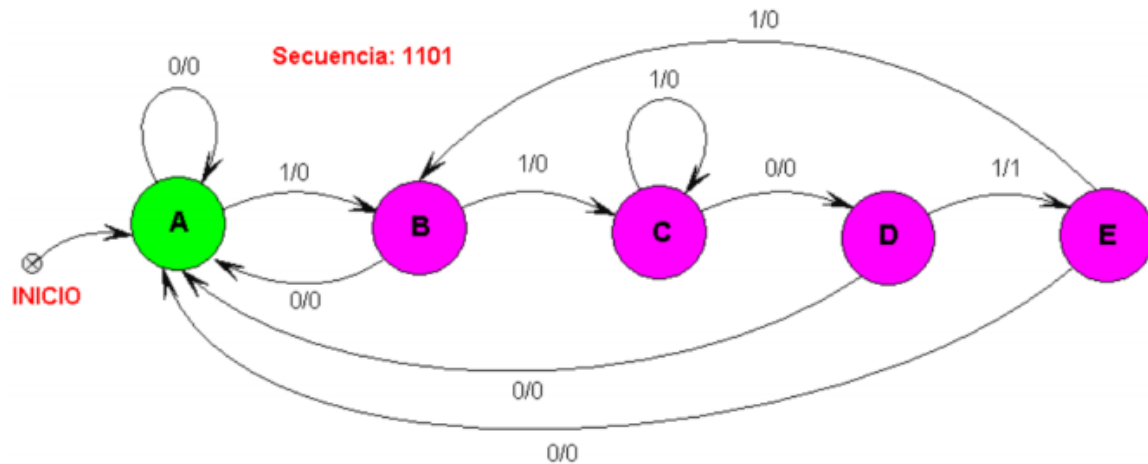
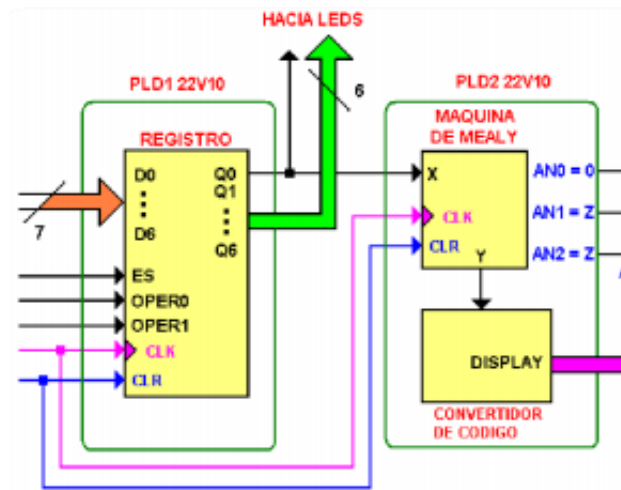


Diagrama de transición de estados

### Procedimiento

- Aplicar toda la metodología de diseño para circuitos secuenciales y diseñar un circuito que detecte la secuencia  $1 \rightarrow 1 \rightarrow 0 \rightarrow 1$ .
- Observe la máquina de Mealy que se muestra en la ilustración anterior.
- Realizar el diseño por medio de FF de su preferencia que permitan implementar la FSM Mealy anterior en Falstad. Además, de realizar un programa en HDL utilizando los tipos de datos enumerados (TYPE) y el diagrama a bloques que se muestra a continuación.



OPER1	OPER0	OPERACIÓN
0	0	RETENCIÓN
0	1	CARGA
1	0	CORRIMIENTO A LA IZQUIERDA
1	1	CORRIMIENTO A LA DERECHA

Tabla 1 Funcionamiento del registro.

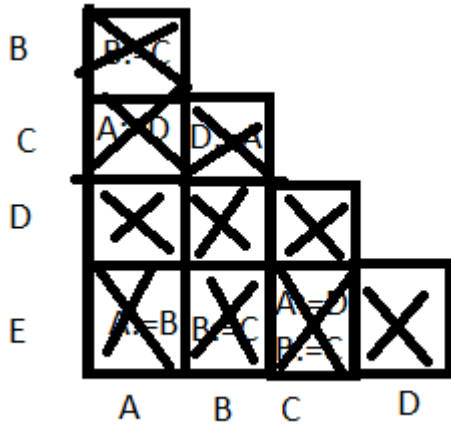
- El bloque de la máquina de Mealy y el convertidor de código. La señal de entrada X toma los bits del registro genérico a través del bit Q0.
- El convertidor de código convierte el valor de salida 0 ó 1 de la máquina de estados en un valor de siete bits para mostrar en un display de siete segmentos de ánodo común. En el display se deben mostrar las letras A (Acierto) y E (Error). La letra A, debe mostrarse cuando se detecte la secuencia deseada y la letra E cuando no se haya detectado la secuencia.
- Verificar el correcto funcionamiento del sistema de la siguiente forma:  
 Cargar un número de siete bits en el registro genérico.  
 El valor de siete bits debe contener la secuencia 1101.  
 Por ejemplo: 0011010, 1110110, 1011011, etc
- Colocar la operación de corrimiento a la derecha en el registro genérico. Con esta operación empezarán a introducirse los bits de forma serial a la máquina Mealy para que comience a detectar la secuencia.
- Verificar la detección de la secuencia en el display.

#### Descripción de la máquina de Mealy por la séxtupla

$M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$   
 $Q = \{A, B, C, D, E\}$   
 $\Sigma = \{0, 1\}$   
 $\Delta = \{0, 1\}$   
 $Edo\_inicial = \{A\}$

$\delta(A, 0) = A, \delta(A, 1) = B$	$\lambda(A, 0) = 0, \lambda(A, 1) = 0$
$\delta(B, 0) = A, \delta(B, 1) = C$	$\lambda(B, 0) = 0, \lambda(B, 1) = 0$
$\delta(C, 0) = D, \delta(C, 1) = C$	$\lambda(C, 0) = 0, \lambda(C, 1) = 0$
$\delta(D, 0) = A, \delta(D, 1) = E$	$\lambda(D, 0) = 0, \lambda(C, 1) = 1$
$\delta(E, 0) = A, \delta(E, 1) = B$	$\lambda(E, 0) = 0, \lambda(E, 1) = 0$

X Estados	0	1
A	A/0	B/0
B	A/0	C/0
C	D/0	C/0
D	A/0	E/1
E	A/0	B/0



Estado	Q2	Q1	Q0
A	0	0	0
B	0	0	1
C	0	1	0
D	0	1	1
E	1	0	0

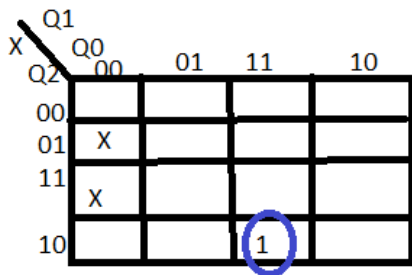
Estado Actual				Entrada	Estado siguiente				Salida	Excitaciones					
	Q2	Q1	Q0	X		*Q2	*Q1	*Q0	Z	J2	K2	J1	K1	J0	K0
A	0	0	0	0	A	0	0	0	0	0	X	0	X	0	X
	0	0	0	1	B	0	0	1	0	0	X	0	X	1	X
B	0	0	1	0	A	0	0	0	0	0	X	0	X	X	1
	0	0	1	1	C	0	1	0	0	0	X	1	X	X	1
C	0	1	0	0	D	0	1	1	0	0	X	X	0	1	X
	0	1	0	1	C	0	1	0	0	0	X	X	0	0	X
D	0	1	1	0	A	0	0	0	0	0	X	X	1	X	1
	0	1	1	1	E	1	0	0	1	1	X	X	1	X	1
E	1	0	0	0	A	0	0	0	0	X	1	0	X	0	X
	1	0	0	1	B	0	0	1	0	X	1	0	X	1	X

Expresiones:

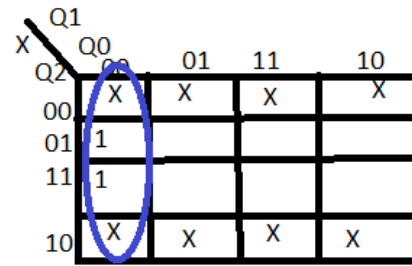
$$Z = XQ2'Q1Q0$$

Flip flop J-K

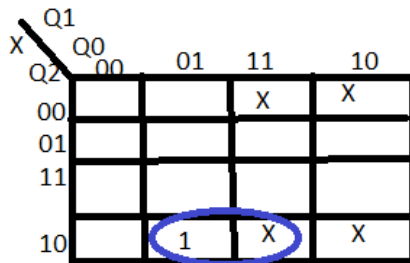
Q	Q (sig)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



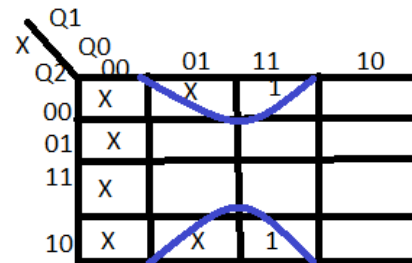
$$J2 = XQ2'Q1Q0$$



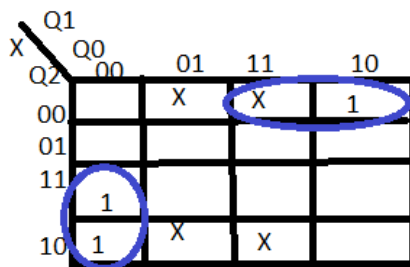
$$K2 = Q1'Q0'$$



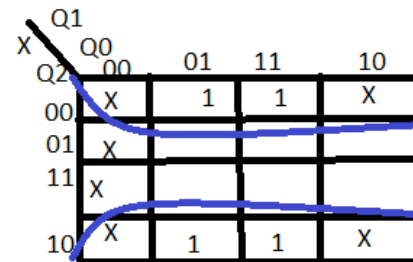
$$J1 = XQ2'Q0$$



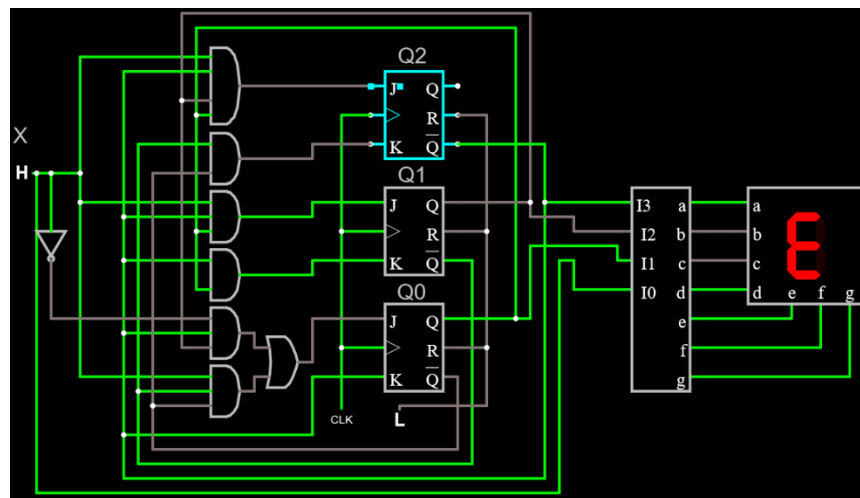
$$K1 = Q2'Q0$$



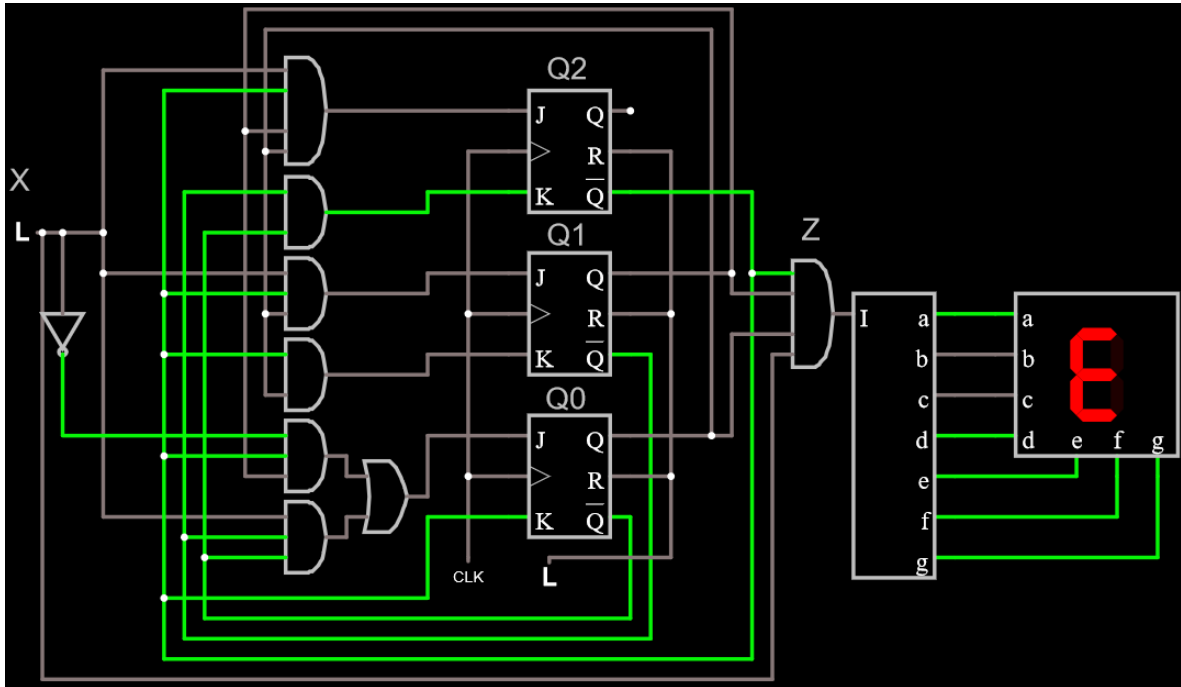
$$J0 = X'Q2'Q1 + XQ1'Q0'$$



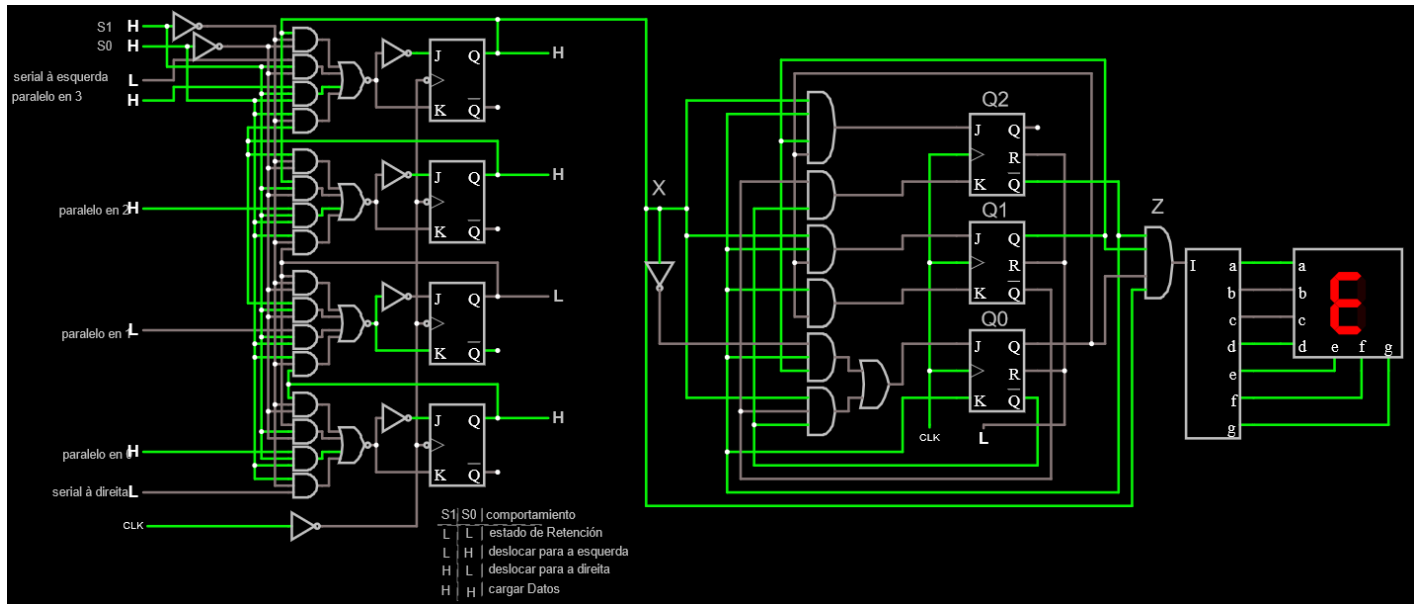
$$K0 = Q2'$$



Link: <https://tinyurl.com/ydwrff42>

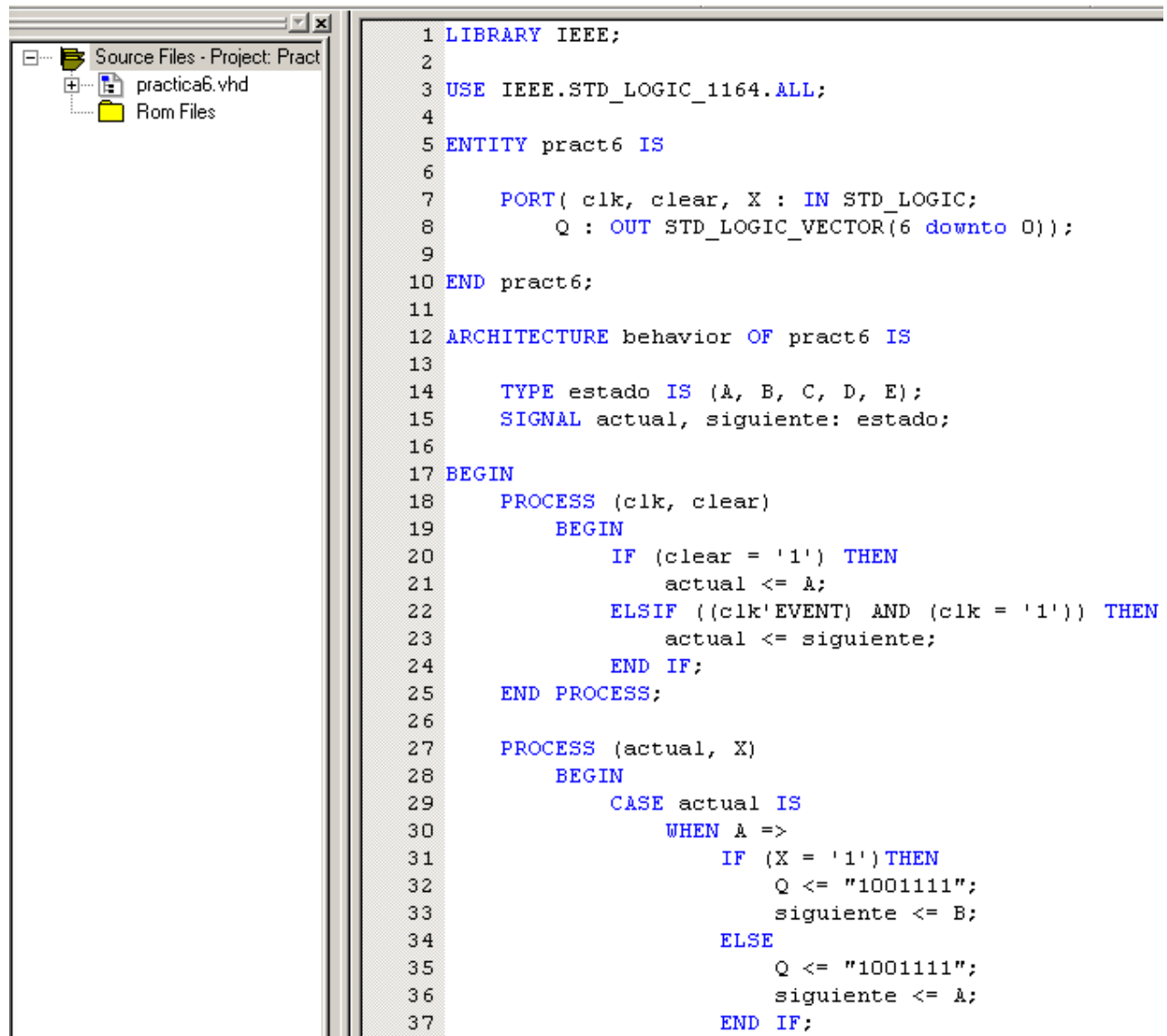


Link: <https://tinyurl.com/yj7d574l>



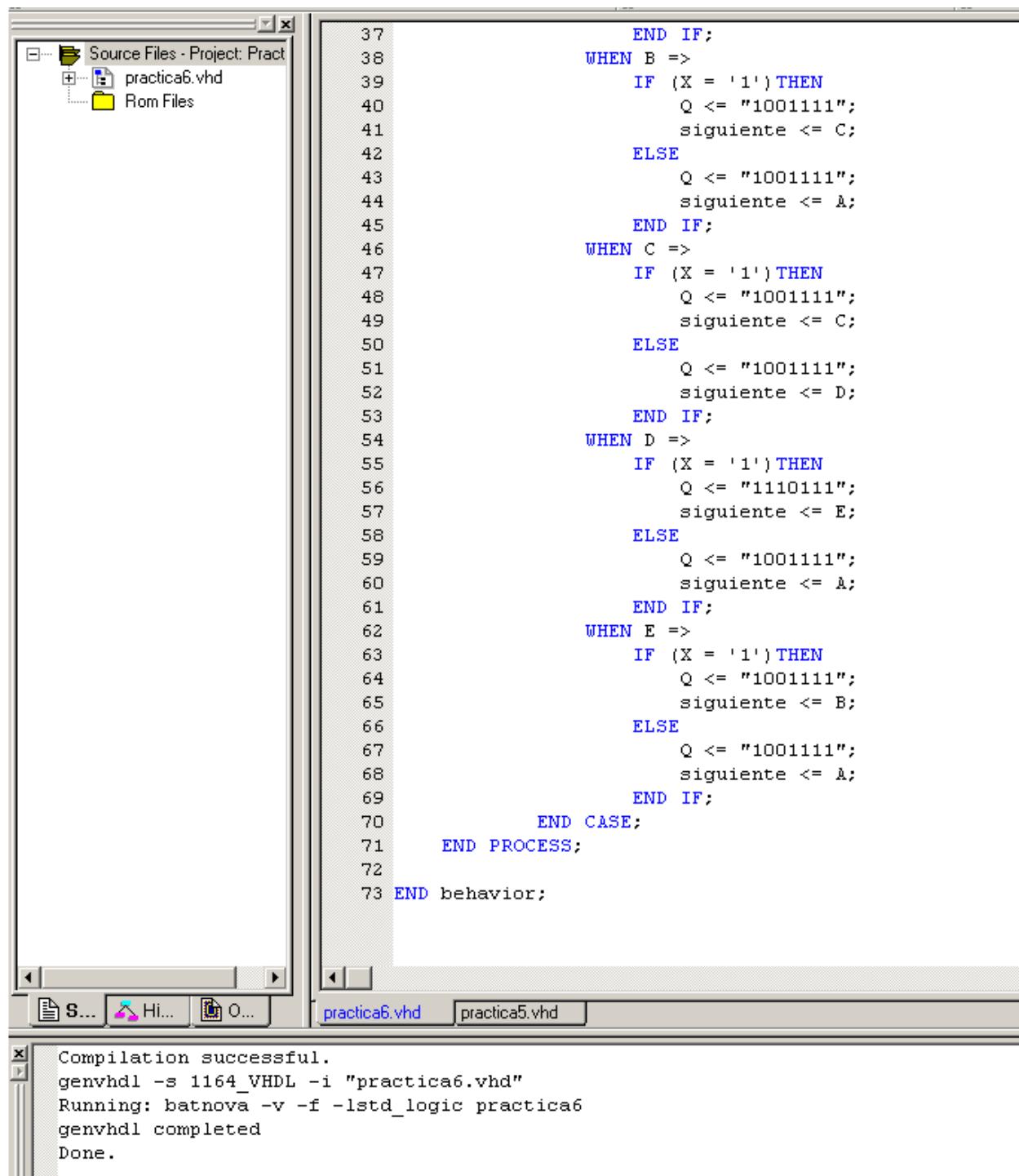
Link: <https://tinyurl.com/yjfi7anf>

## Código y simulación.

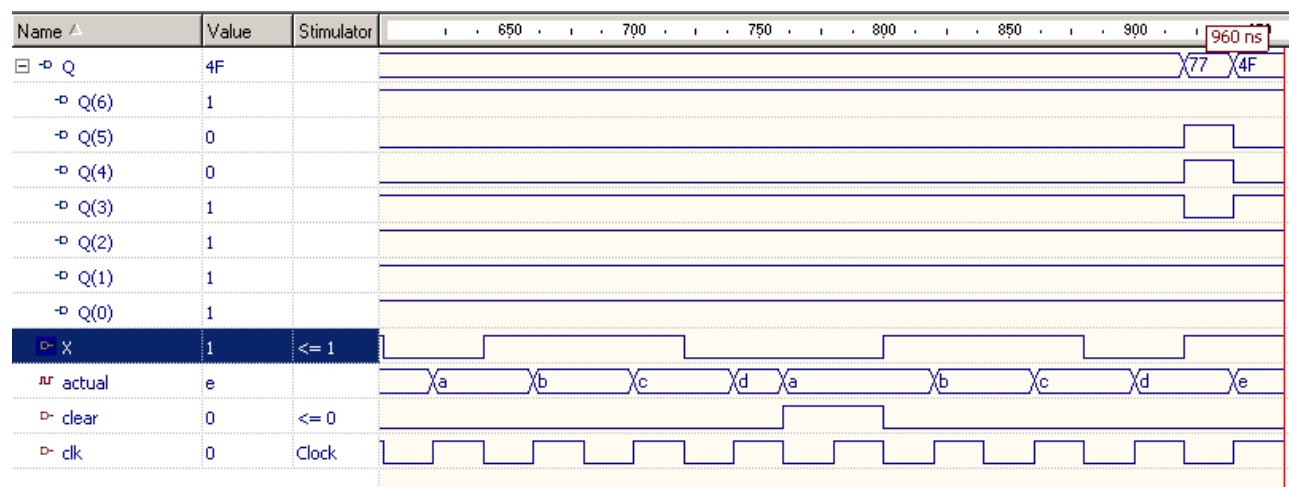
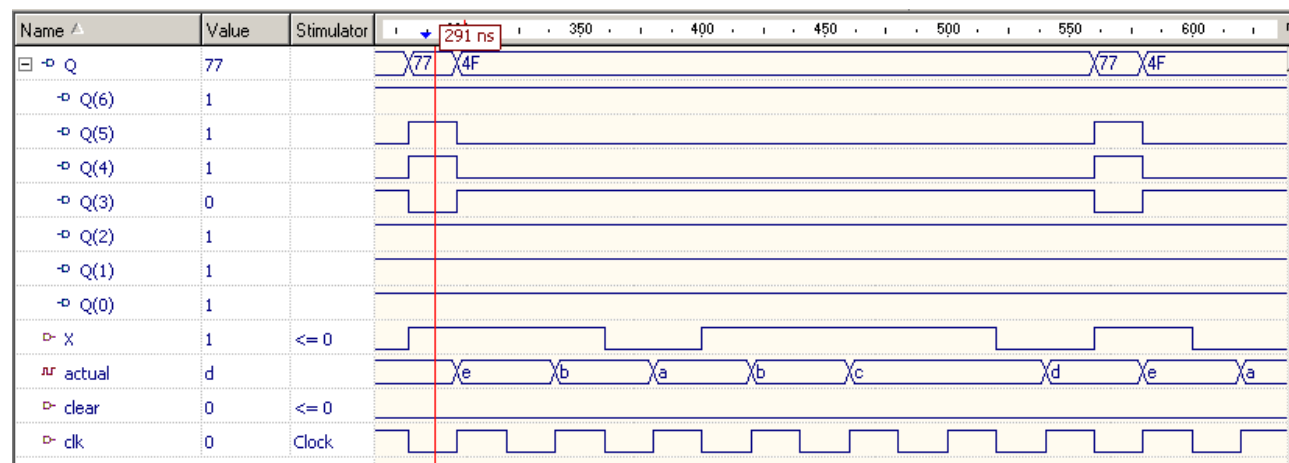
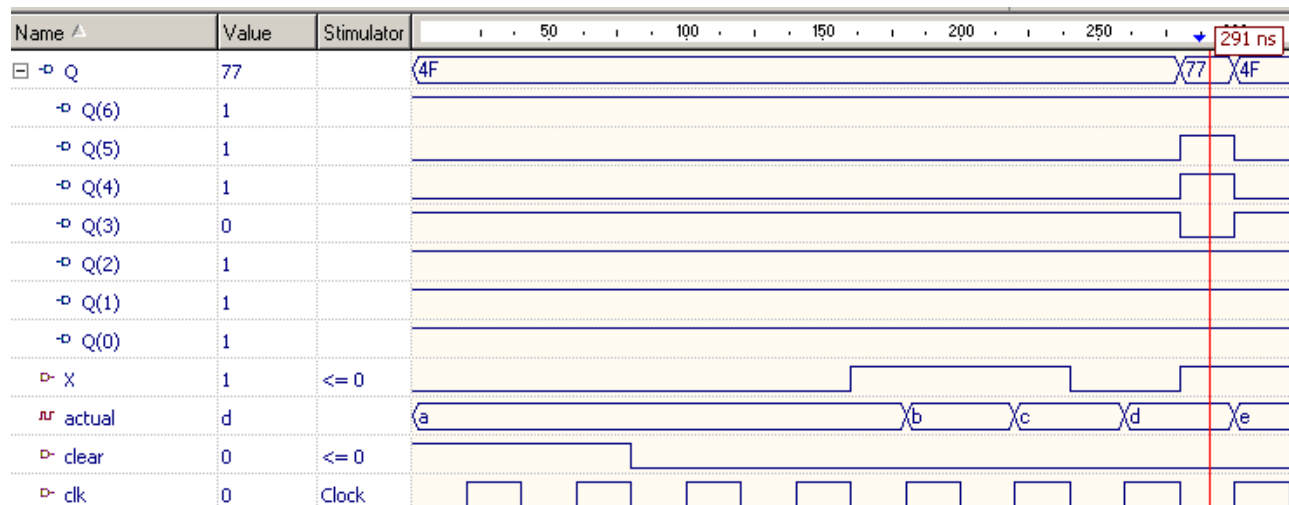


The image shows a screenshot of a VHDL code editor. On the left, a 'Source Files - Project: Pract' window displays a file tree with 'practica6.vhd' and 'Rom Files'. The main editor area shows the VHDL code for 'pract6.vhd', which is a counter entity. The code is as follows:

```
1  LIBRARY IEEE;
2
3  USE IEEE.STD_LOGIC_1164.ALL;
4
5  ENTITY pract6 IS
6
7      PORT( clk, clear, X : IN STD_LOGIC;
8            Q : OUT STD_LOGIC_VECTOR(6 downto 0));
9
10 END pract6;
11
12 ARCHITECTURE behavior OF pract6 IS
13
14     TYPE estado IS (A, B, C, D, E);
15     SIGNAL actual, siguiente: estado;
16
17 BEGIN
18     PROCESS (clk, clear)
19     BEGIN
20         IF (clear = '1') THEN
21             actual <= A;
22         ELSIF ((clk'EVENT) AND (clk = '1')) THEN
23             actual <= siguiente;
24         END IF;
25     END PROCESS;
26
27     PROCESS (actual, X)
28     BEGIN
29         CASE actual IS
30             WHEN A =>
31                 IF (X = '1') THEN
32                     Q <= "1001111";
33                     siguiente <= B;
34                 ELSE
35                     Q <= "1001111";
36                     siguiente <= A;
37                 END IF;
```







### **Observaciones y conclusiones.**

Durante el desarrollo de esta práctica no surgieron problemas, la Gal si me permitía ingresar todas las salidas del display, además, de las entradas clk, clear y X, las cuales nos permiten lograr un correcto funcionamiento para la máquina de estados tipo Mealy, con respecto a la implementación en falstad fue un poco laboriosa, sin embargo si tenemos el análisis para cada flip – flop y las entradas del decodificador implementarlo solo requiere un poco de tiempo y paciencia, por otra parte mostrar en el display cada una de las letras necesarias para el detector de secuencia fue bastante sencillo al utilizar un único display, por ultimo no fue complicado realizar el análisis para el funcionamiento de la practica ya que contaba con el conocimiento para este análisis.

### **Anexos y bibliografía**

#### **Referencias bibliográficas:**

- Floyd, T. L. (2021). *Fundamentos De Sistemas Digitales* (9.<sup>a</sup> ed.) [Libro electrónico]. PRENTICE HALL/PEARSON.  
[https://www.academia.edu/34699883/Libro\\_fundamentos\\_de\\_sistemas\\_digitales\\_floyd\\_9ed\\_PDF](https://www.academia.edu/34699883/Libro_fundamentos_de_sistemas_digitales_floyd_9ed_PDF)
- Maxinez, D. (2013). *Programación de sistemas digitales con VHDL* (1.<sup>a</sup> ed.). Grupo Editorial Patria. <https://editorialpatria.com.mx/pdf/files/9786074386219.pdf>