



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 2:

REGISTROS

CIUDAD DE MEXICO MARZO DE 2021

Para esta práctica utilizaremos la GAL C22V10 en el Simulador Galaxy, por otro lado, esta práctica consistió en programar un Registro universal de 7 bits con una entrada serial, clear asíncrono, una señal de reloj y dos señales de operación, a continuación, las salidas y entradas de ocupamos de la GAL y el código correspondiente a la práctica.

Durante esta práctica utilizare las entradas clk (señal de reloj), clear (clr), ES (Entrada serial), OPER0 y OPER1 donde su funcionamiento esta descrito por la tabla de operación.

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

Funcionamiento del registro.

Además, utilizare las entradas/salidas D y Q que son del tipo INOUT de 7 bits.

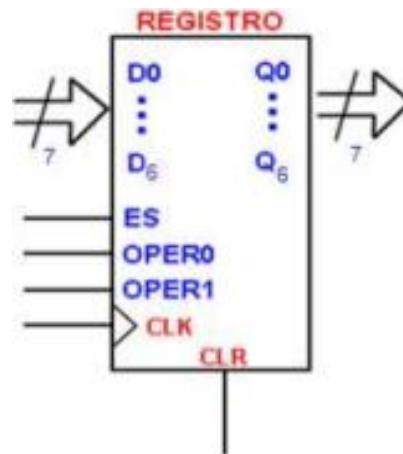


Diagrama de Esquemático

Desplazamiento a la derecha

En esta sección puede observarse el código para un registro de entrada paralelo y salida paralela de 7 bits, donde tendrá un desplazamiento hacia la derecha, además hacemos uso de la instrucción LOOP que nos permitirá realizar la asignación para cada señal de salida auxiliar correspondiente.

```
ELSIF (OPER(0) = '1') AND (OPER(1) = '1') THEN
    FOR i IN 0 TO 5 LOOP
        Q(i) <= Q(i+1);
    END LOOP;
    Q(6) <= ES;
END IF;
```

Desplazamiento a la izquierda

En esta sección puede observarse el código para un registro de entrada paralelo y salida paralela de 7 bits, donde tendrá un desplazamiento hacia la izquierda, note que hacemos uso de la instrucción LOOP que nos permitirá realizar la asignación para cada señal de salida auxiliar correspondiente.

```
ELSIF (OPER(0) = '0') AND (OPER(1) = '1') THEN
    FOR i IN 1 TO 6 LOOP
        Q(i) <= Q(i-1);
    END LOOP;
    Q(0) <= ES;
```

Para cargar los datos

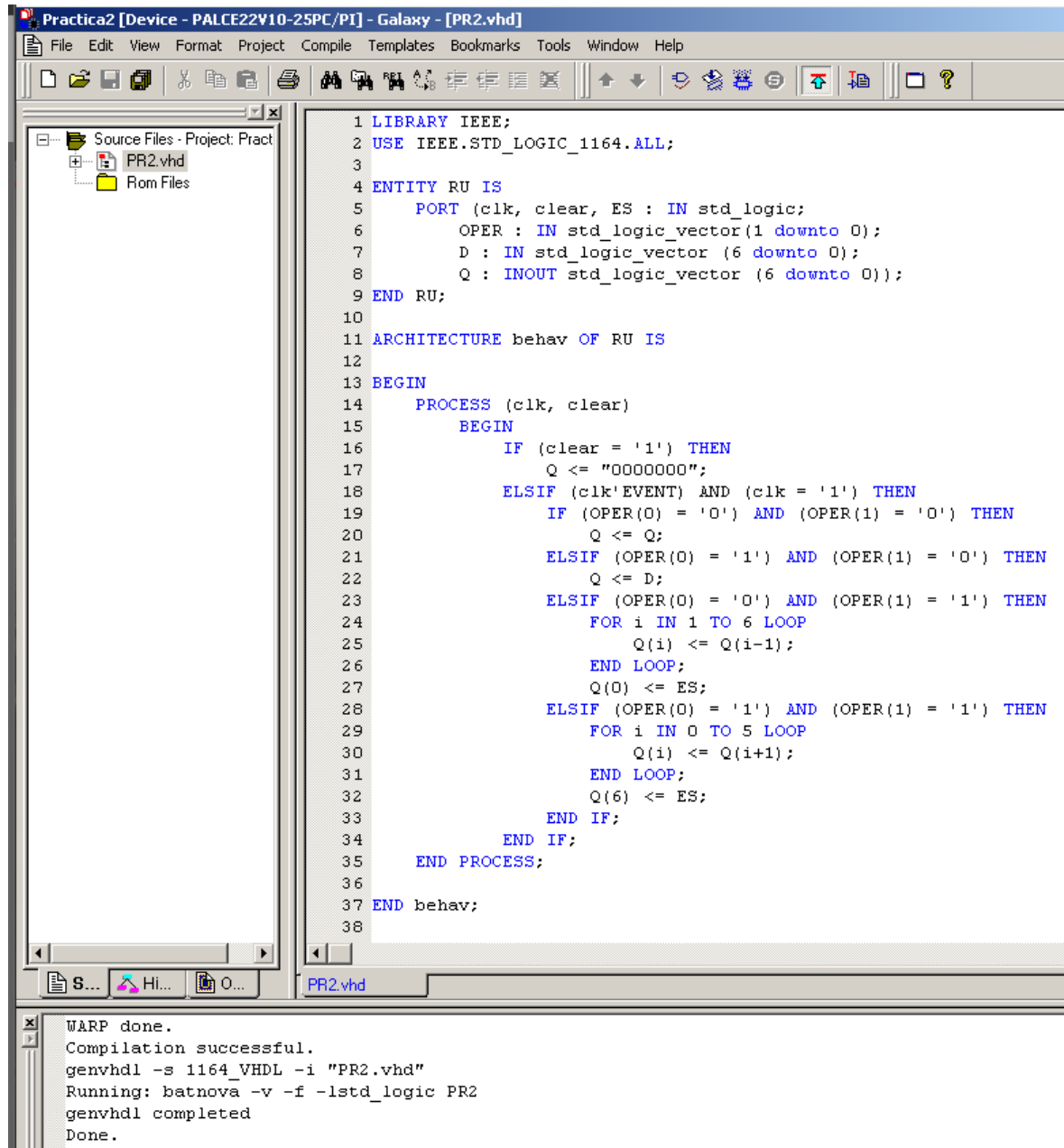
```
ELSIF (OPER(0) = '1') AND (OPER(1) = '0') THEN
    Q <= D;
```

Para mantener los datos

```
IF (OPER(0) = '0') AND (OPER(1) = '0') THEN
    Q <= Q;
```

Código y simulación Practica 2.

Nota: El desplazamiento a la derecha corresponde a un cambio hacia abajo y el desplazamiento a la izquierda corresponde a un cambio hacia arriba.



The screenshot displays the Galaxy IDE interface. The main window shows the VHDL code for 'PR2.vhd'. The code defines an entity 'RU' with three ports: 'clk', 'clear', and 'ES' (all inputs of type 'std_logic'), and two 6-bit output vectors 'OPER' and 'Q' (both of type 'std_logic_vector'). The architecture 'behav' contains a process 'clk, clear' that implements a shift register. It uses conditional logic to handle 'clear', 'load' (when both 'OPER(0)' and 'OPER(1)' are '0'), and 'shift' (when 'OPER(0)' is '1' and 'OPER(1)' is '0'). The shift operation is performed using a loop to shift the data from 'OPER' into 'Q'. The output 'Q(6)' is assigned the value of 'ES'. The simulation output at the bottom shows a successful compilation and simulation run.

```
1 LIBRARY IEEE;
2 USE IEEE.STD_LOGIC_1164.ALL;
3
4 ENTITY RU IS
5     PORT (clk, clear, ES : IN std_logic;
6           OPER : IN std_logic_vector(1 downto 0);
7           D : IN std_logic_vector (6 downto 0);
8           Q : INOUT std_logic_vector (6 downto 0));
9 END RU;
10
11 ARCHITECTURE behav OF RU IS
12
13 BEGIN
14     PROCESS (clk, clear)
15     BEGIN
16         IF (clear = '1') THEN
17             Q <= "00000000";
18         ELSIF (clk'EVENT) AND (clk = '1') THEN
19             IF (OPER(0) = '0') AND (OPER(1) = '0') THEN
20                 Q <= Q;
21             ELSIF (OPER(0) = '1') AND (OPER(1) = '0') THEN
22                 Q <= D;
23             ELSIF (OPER(0) = '0') AND (OPER(1) = '1') THEN
24                 FOR i IN 1 TO 6 LOOP
25                     Q(i) <= Q(i-1);
26                 END LOOP;
27                 Q(0) <= ES;
28             ELSIF (OPER(0) = '1') AND (OPER(1) = '1') THEN
29                 FOR i IN 0 TO 5 LOOP
30                     Q(i) <= Q(i+1);
31                 END LOOP;
32                 Q(6) <= ES;
33             END IF;
34         END IF;
35     END PROCESS;
36
37 END behav;
38
```

WARP done.
Compilation successful.
genvhdl -s 1164_VHDL -i "PR2.vhd"
Running: batnova -v -f -lstd_logic PR2
genvhdl completed
Done.

Clear activado para asignar un valor conocido

The screenshot displays the Active-HDL Sim (PR2) - Waveform2 * interface. The main window shows a simulation waveform with a time axis from 0 to 350 ns. A red vertical line marks the 70 ns point. The waveform shows signals D, D(6), D(5), D(4), D(3), D(2), D(1), D(0), ES, OPER, Q, Q(6), Q(5), Q(4), Q(3), Q(2), Q(1), Q(0), clear, and clk. The 'clear' signal is set to 1 at 70 ns. The 'clk' signal is a clock signal. The 'D' signal is a vector of 7 bits, with D(6) to D(0) shown. The 'ES' signal is 0. The 'OPER' signal is 0. The 'Q' signal is 00. The 'Q(6)' to 'Q(0)' signals are 0. The 'clear' signal is 1. The 'clk' signal is 0.

The Stimulators dialog box is open, showing the Signals tab. The Signals list includes:

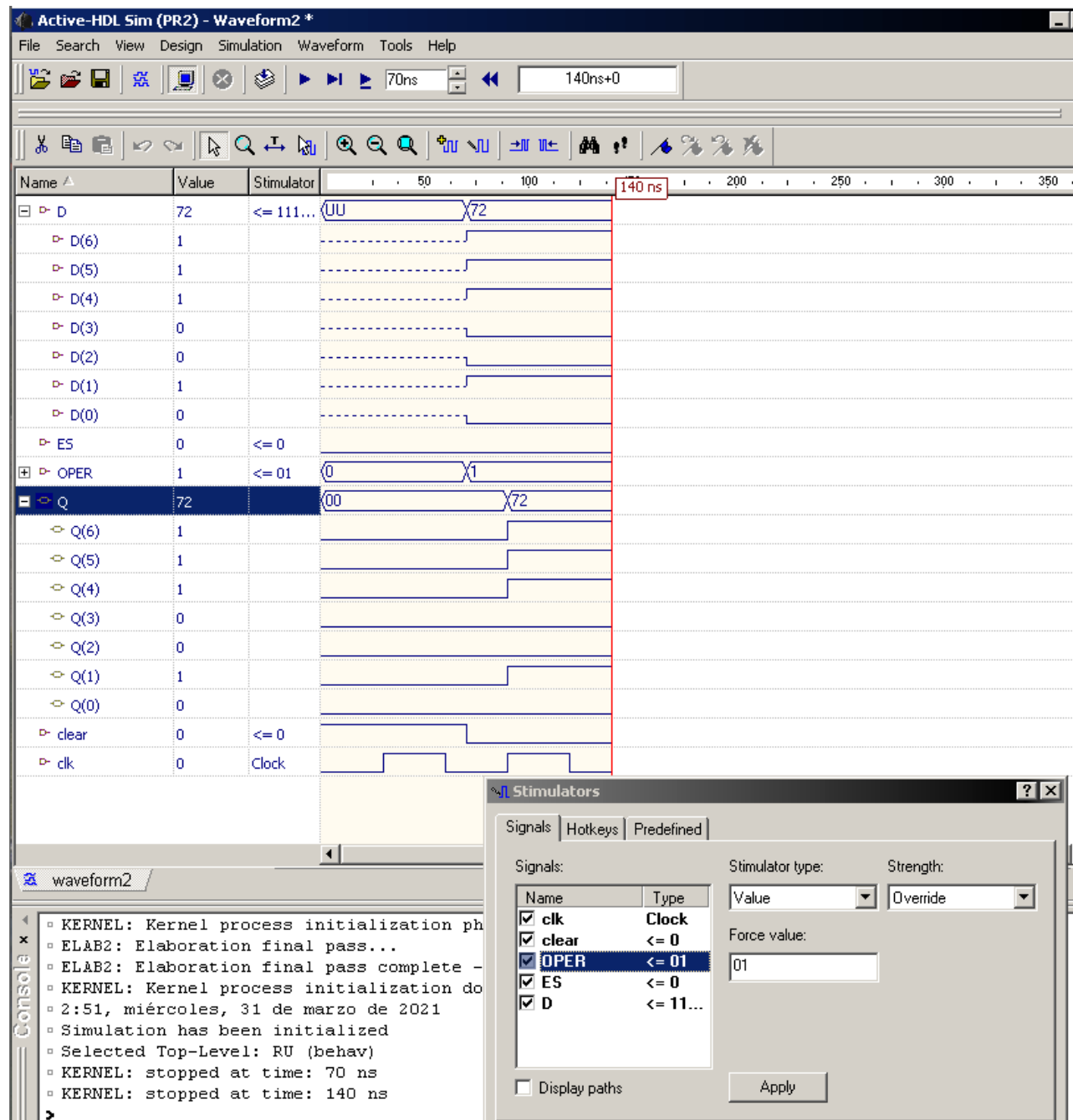
Name	Type
<input checked="" type="checkbox"/> clk	Clock
<input checked="" type="checkbox"/> clear	<= 1
<input checked="" type="checkbox"/> OPER	<= 00
<input checked="" type="checkbox"/> ES	<= 0
<input type="checkbox"/> D	

The Stimulator type is set to Clock. The Strength is set to 1. The Display paths checkbox is unchecked. The Apply button is visible.

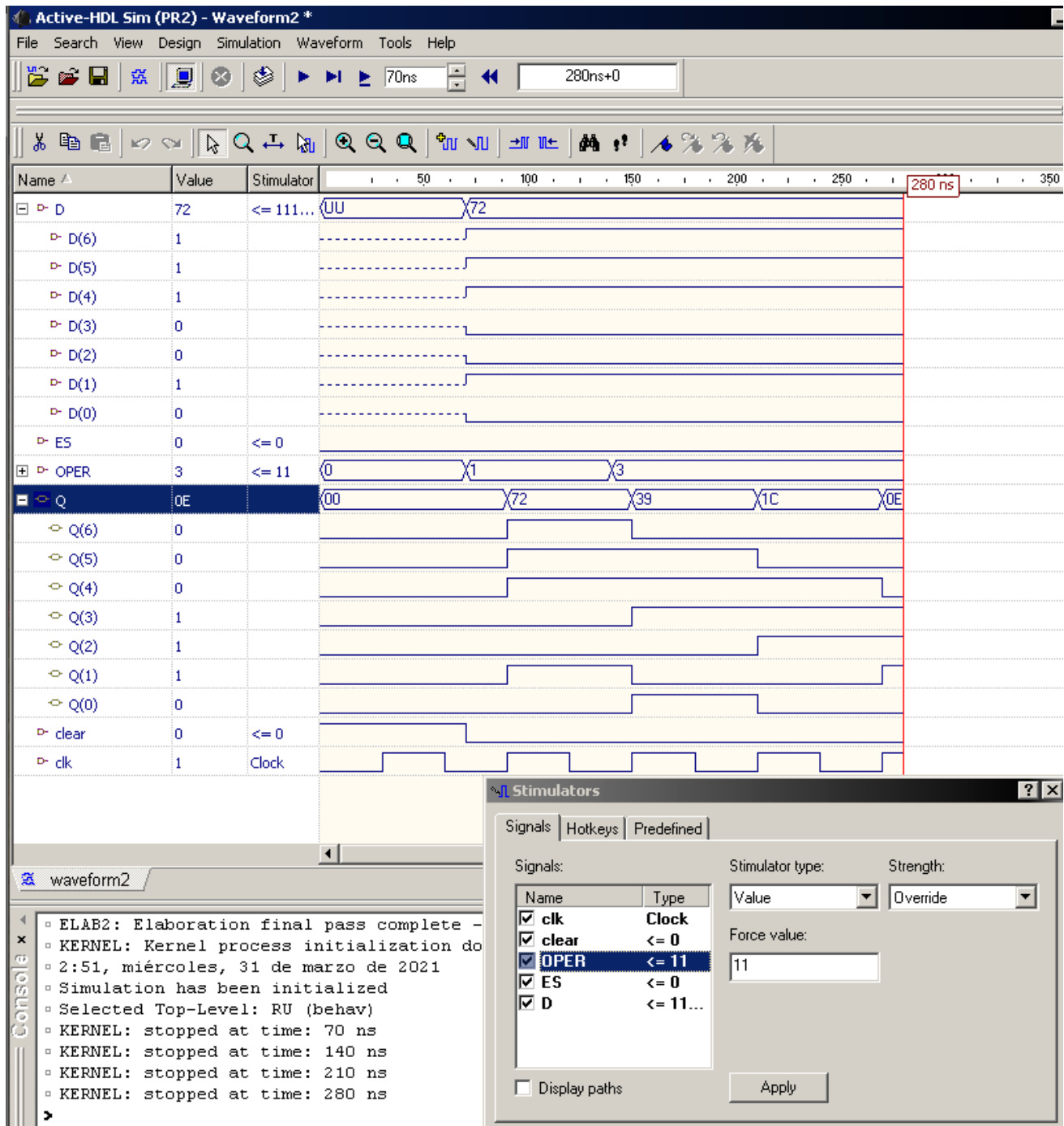
The Console window shows the following output:

```
o KERNEL: Main thread initiated.
o KERNEL: Kernel process initialization ph
o ELAB2: Elaboration final pass...
o ELAB2: Elaboration final pass complete -
o KERNEL: Kernel process initialization do
o 2:51, miércoles, 31 de marzo de 2021
o Simulation has been initialized
o Selected Top-Level: RU (behav)
o KERNEL: stopped at time: 70 ns
```

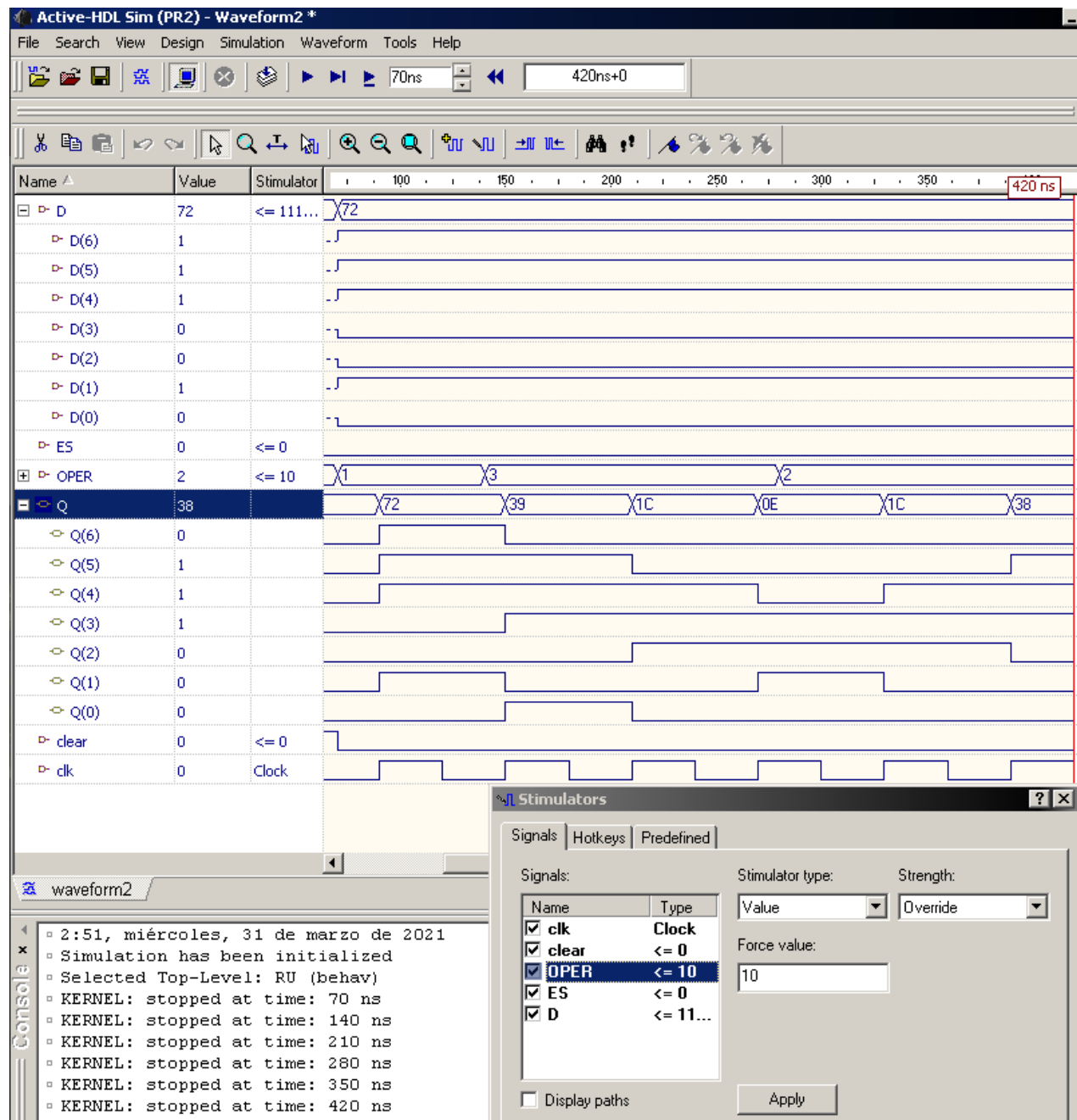
Cargamos el valor para D = "1110010" de 7 bits y asignamos clear = 0



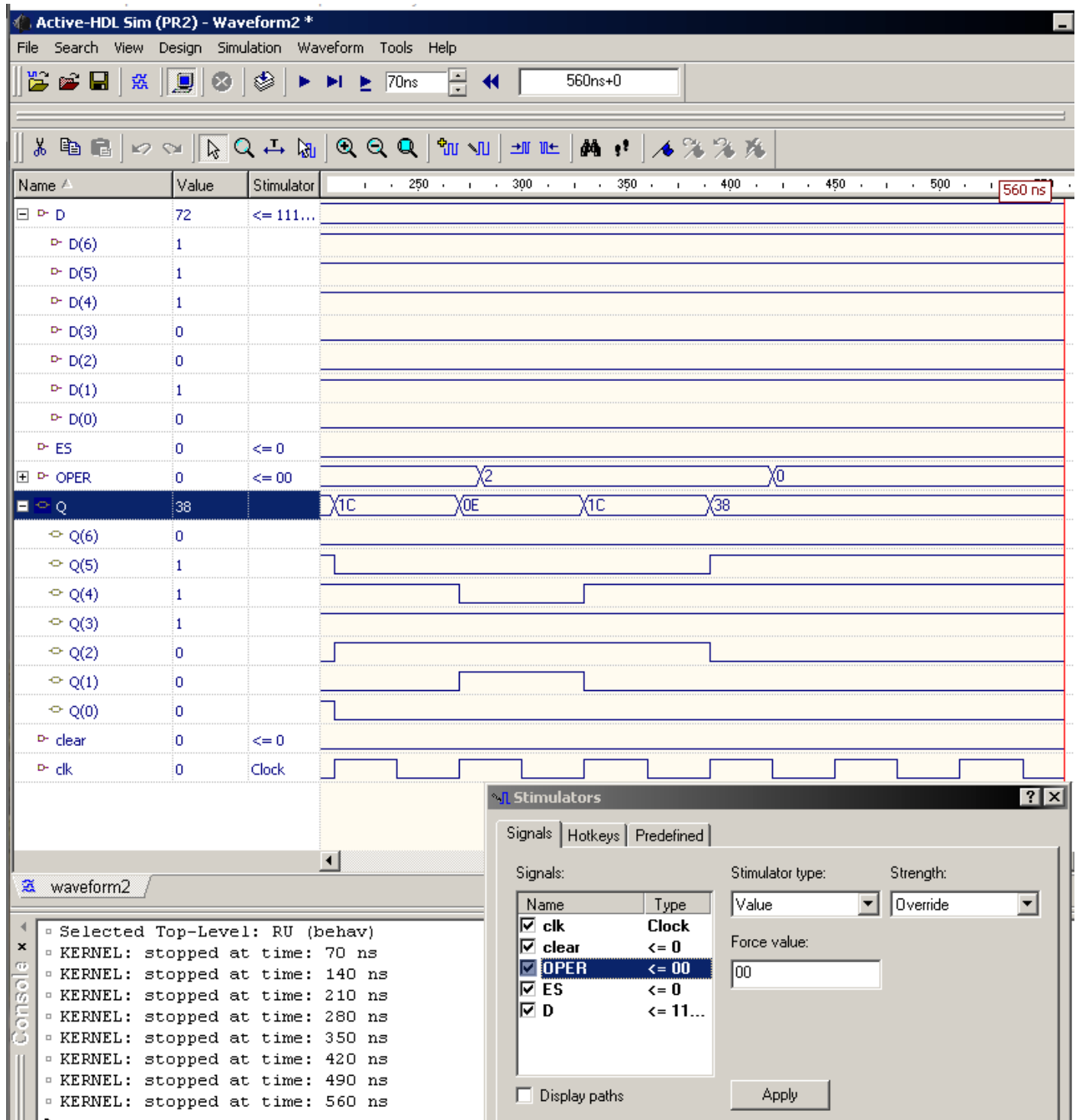
Asignamos a $OPER(0) = 1$ y $OPER(1) = 1$ para tener un desplazamiento hacia la derecha (Abajo en el simulador), note que la Entrada Serial (ES) = 0.



Asignamos a $OPER(0) = 0$ y $OPER(1) = 1$ para tener un desplazamiento hacia la izquierda (Arriba en el simulador), note que la Entrada Serial (ES) = 0.



Asignamos a $OPER(0) = 0$ y $OPER(1) = 0$ para mantener los datos correspondientes a D y Q, es decir el modo de operación Retención.



Conclusiones y observaciones

A través de esta práctica conseguí una retroalimentación sobre lo previamente visto acerca de los Registros, los cuales están conformados por las entradas y salidas, para esta práctica simulamos un registro universal el cual tiene una entrada en paralelo y una salida en paralelo, además están integrados por una señal de reloj, así como por el clear y una entrada serial.

Durante el desarrollo de la práctica ocurrieron algunos problemas puesto que el simulador Galaxy marcaba errores cuando asignabas más de una entrada o salida de 7 bits, sin embargo, este problema no ocurría si le asignábamos menos de 5 bits, así que, fue necesario nuevamente instalar el simulador para que funcionara correctamente, además cuando ejecutaba la simulación del registro en cualquier desplazamiento (derecha/izquierda) pude notar que la salida Q a pesar de ser bidireccional no reflejaba ningún cambio, esto ocurría ya que cuando seleccionaba una entrada para asignarle un estimulador esta no puede modificarse, debido a esto al momento de ejecutar la simulación debe retirarse de la parte de estimuladores la entrada Q para que su valor pueda cambiar, también durante el desarrollo requerí el uso de la instrucción `“clk’EVENT and clk =’1’ ”` la cual utilicé para detectar los flancos de subida.

Anexos y bibliografía

Referencias bibliográficas:

Floyd, T. L. (2021). *Fundamentos De Sistemas Digitales* (9.^a ed.) [Libro electrónico]. PRENTICE HALL/PEARSON.
https://www.academia.edu/34699883/Libro_fundamentos_de_sistemas_digitales_floyd_9ed_PDF

Maxinez, D. (2013). *Programación de sistemas digitales con VHDL* (1.^a ed.). Grupo Editorial Patria. <https://editorialpatria.com.mx/pdf/files/9786074386219.pdf>

Referencias electrónicas.

<https://personales.unican.es/manzanom/planantiguo/edigitali/REGG4.pdf>

<http://www.fdi.ucm.es/profesor/jjruiz/lec/temas/Sesion2.pdf>

http://mixteco.utm.mx/~merg/AC/vhdl/6_Especificacion_del_comportamiento.pdf

<https://www.ele.uva.es/~sduenas/ASIC/VHDL2B.pdf>