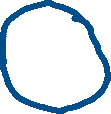
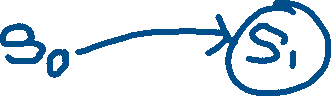
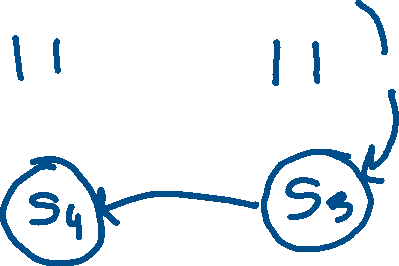
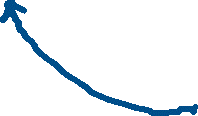
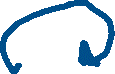
**Reto segundo Parcial: Controlador Display LCD**

**Laboratorio de Sistemas Digitales  
Profesor: Emilio Caballero**

En cuanto al diseño del código de VHDL del controlador de LCD, primero se tuvo que revisar la hoja de datos y los pasos que se tienen que seguir para inicializar la pantalla LCD, que son los siguientes:

1. instr(0x38): Poner la pantalla en modo 8 bits, encendido y 2 líneas.
2. instr(0x0F): Encender el display, el cursor y el parpadeo (opcional).
3. instr(0x01): Hacer un clear al display.
4. instr(0x06): Poner la pantalla en modo entrada y configurar el cursor para que se mueva automáticamente a la derecha después de ingresar los datos.

Para poder realizar estos pasos de inicialización, y posteriormente tener la vía libre para ingresar datos de 8 bits que representen los caracteres, se desarrolló la siguiente máquina de estados:



Adicionalmente, se tenían que implementar comandos de lectura y escritura de texto en archivos .txt para poder leer los comandos con el formato  
RS RW DATA RESET  
0 0 01010101 1  
y crear un archivo de texto, que sería la entrada para escribir mediante el uso del simulador.

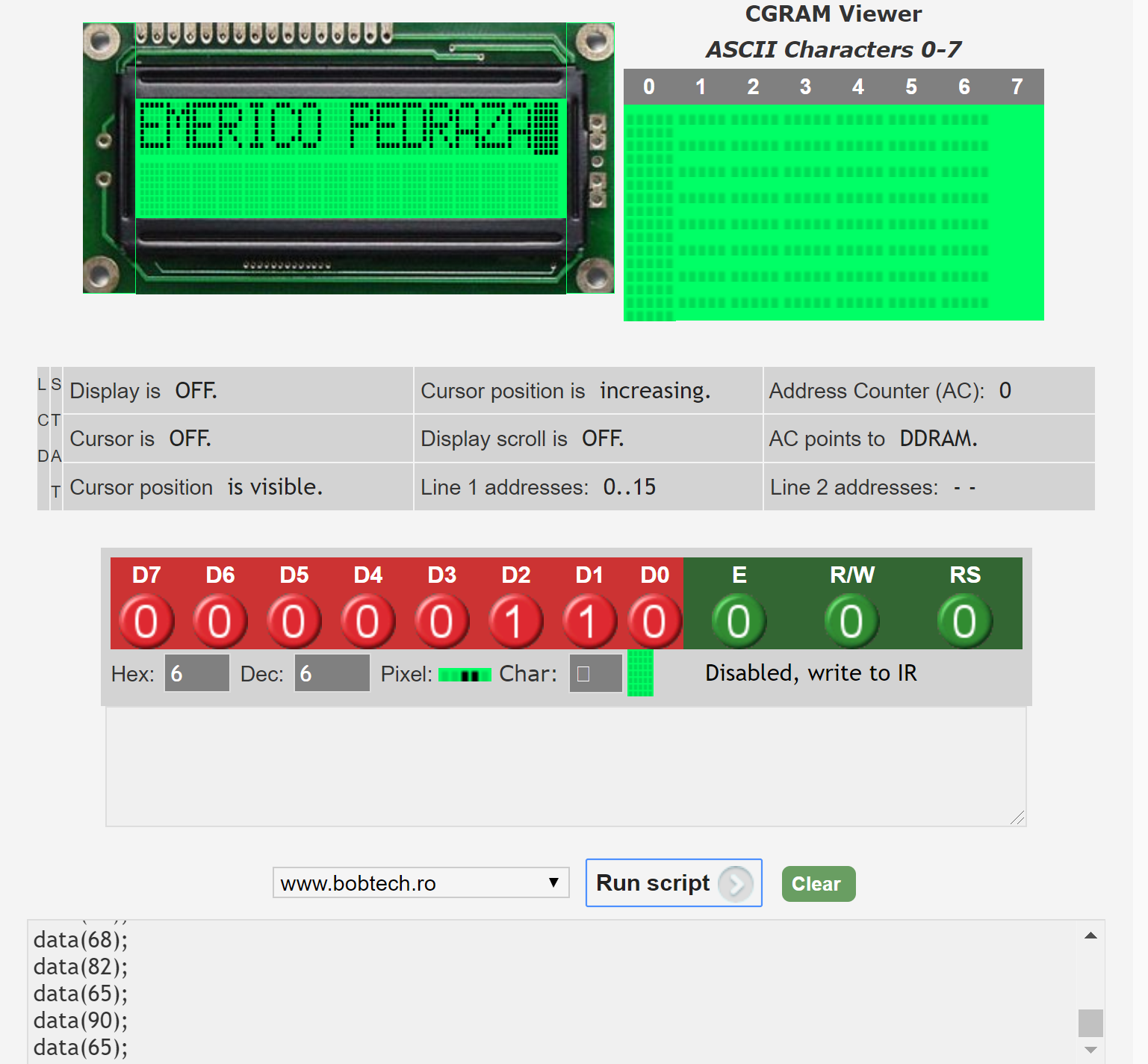
**Este es el texto de entrada que fue utilizado (también se encuentra en el repositorio de GitHub):**

**1 0 01000101 1  
1 0 01001101 1  
1 0 01000101 1  
1 0 01010010 1  
1 0 01001001 1  
1 0 01000011 1  
1 0 01001111 1  
1 0 00010000 1  
1 0 01010000 1  
1 0 01000101 1  
1 0 01000100 1  
1 0 01010010 1  
1 0 01000001 1  
1 0 01011010 1  
1 0 01000001 1**

**Esta fue la salida creada (también se encuentra en el repositorio de GitHub):**

**instr(0x38);  
instr(0x0F);  
instr(0x01);  
instr(0x06);  
data(69);  
data(77);  
data(69);  
data(82);  
data(73);  
data(67);  
data(79);  
data(16);  
data(80);  
data(69);  
data(68);  
data(82);  
data(65);  
data(90);  
data(65);**

**Evidencia del simulador:**



**Diagrama de tiempo:**

