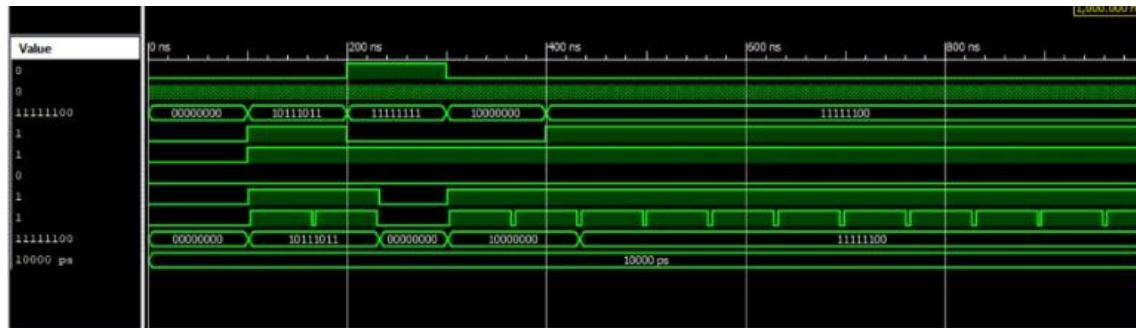


Reporte de comportamiento del controlador LCD

En este reto programamos el controlador de forma de que leyera el archivo y también pudiera escribir.



Para lograr el funcionamiento de la lcd tuve que entender lo que era RW, RS y EN. RW era para leer o escribir, RS para seleccionar los registros de memoria y enable era necesario que se activara antes de cada instrucción y desactivara después de ella y así en todas las instrucciones.

En esta parte del Testbench fue donde le indicaba que tenía que leer:

```
-- Stimulus process
stim_proc: process

    file fin: TEXT open READ_MODE is "input.txt";

    variable current_read_line : line;
    variable current_read_field : bit_vector (10 downto 0);
begin

    while (not endfile(fin)) loop

        wait for 100 ns;
        readline(fin, current_read_line);
        read(current_read_line, current_read_field);

        RESET<= current_read_field(10);
        RS<= current_read_field(9);
        RWDATA<= current_read_field(8);
        DATA_INSTRUCTIONS<= current_read_field(7 downto 0);

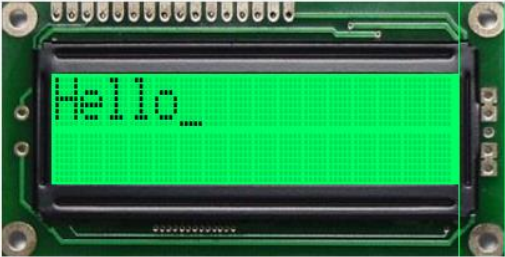
    end loop;
```

Y le daba un archivo para leer llamada “input.txt” que era así.

```
0 1 11100010
0 0 10001110
0 0 01010100
0 1 11111000
1 1 10101010
1 1 11111100
1 0 11001100
```

Los tres primeros números son instrucciones, mientras que los otros es lo que tenía que leer.

Este es el resultado de la simulación.



CGRAM Viewer

ASCII Characters 0-7

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

| | | | |
|----|-----------------------------|--------------------------------|--------------------------|
| LS | Display is ON. | Cursor position is increasing. | Address Counter (AC): 11 |
| CT | Cursor is ON, blinking. | Display scroll is OFF. | AC points to DDRAM. |
| DA | Cursor position is visible. | Line 1 addresses: 0..15 | Line 2 addresses: 64..79 |

D7

D6

D5

D4

D3

D2

D1

D0

E

R/W

RS

0

0

0

0

0

0

0

0

0

0

0

Hex: 0 Dec: 0 Pixel: Char: Disabled, write to IR

www.bobtech.ro

Run script

Clear

```
instr(0);
instr(15);
instr(1);
instr(6);
data(72);
data(101);
data(108);
```