

DISEÑO DE SISTEMAS DIGITALES CON FPGA

GAME OF LIFE

**Luis José Martín Torres,
Francisco Roth,
Mariana Nicole Shimane**

Facultad de Ciencias Exactas y Naturales,
Universidad de Buenos Aires

July 13, 2023

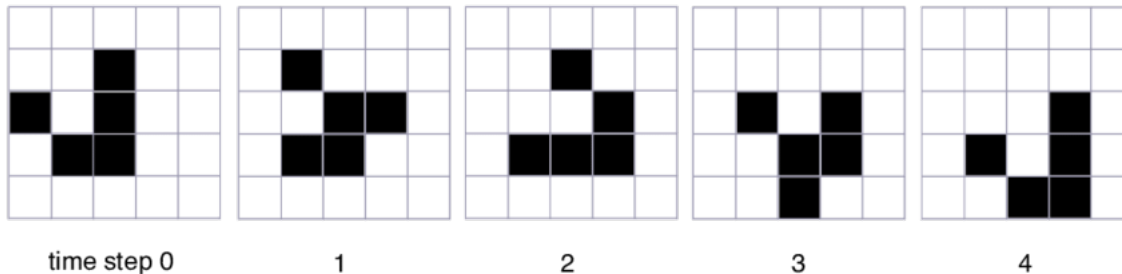
MOTIVACIÓN

- PDES (Parallel Discrete Event Simulation)
 - Grandes simulaciones
 - Cómputo distribuido y en paralelo
 - Memoria compartida o distribuida
 - El sistema solo cambia de estado cuando ocurre un evento
 - Síncrono o asíncrono
 - Cola de eventos
 - Algunas aplicaciones:
 - Redes sociales, transporte, sector militar, logística, telecomunicaciones...

GAME OF LIFE

REGLAS:

- Contamos con una grilla de $n \times m$ células
- Cada célula puede estar "viva" o "muerta" (0 o 1)
- Por cada iteración vamos a actualizar el estado de las células basándonos en las 8 células vecinas
- Si la célula esta viva:
 - < 2 o > 3 células vecinas vivas \Rightarrow Muere
 - 2 o 3 células vecinas vivas \Rightarrow Vive
- Si la célula esta muerta:
 - exactamente 2 células vecinas vivas \Rightarrow Vive
 - Cualquier otro caso, sigue muerta



```

entity cell is
    port ( clk : in std_logic;
           neighbors : in std_logic_vector (7 downto 0);
           value_i : in std_logic;
           enable_set_i : in std_logic;
           enable_run_i : in std_logic;
           current_state_o : out std_logic);
end cell;

entity board is
    generic(N : natural := 16);
    port (
        clk : in STD_LOGIC;
        enable_set : in STD_LOGIC;
        enable_run : in std_logic;

        value : in STD_LOGIC;
        in_x : in integer range 0 to N-1;
        in_y : in integer range 0 to N-1;

        out_o : out STD_LOGIC
    );
end board;

```

```

entity game_of_life is
  generic(N : natural := 16;
          MAX_ITER : natural := 2**32);
  port (
    clk : in std_logic;

    iterations : in integer;
    en_iter_write : in std_logic;

    en_cell_write : in std_logic;
    in_value : in std_logic;
    in_x : in integer range 0 to N-1;
    in_y : in integer range 0 to N-1;

    in_run_simulation : in std_logic;

    out_cell_value : out std_logic;
    out_remaining_iter : out integer
  );
end game_of_life;

```

RETOS

- Hacerlo parametrizable en cuanto a la cantidad de células a simular.
- Asignar los valores iniciales de todo el board, y leerlos al terminar
- ¿Usar multiplexor para configurar la lógica de escritura inicial, o hacer el loop que hicimos? Lo mismo con la lectura.
- Conectar vecinos de manera parametrizable
 - Casos borde: aristas y esquinas
 - El resto
- Testearlo con inputs grandes con la escritura y la lectura por loops.
 - Comparamos con una implementación en Ruby.

MEJORAS

- Optimizar la escritura y la lectura, haciéndolo por batches en vez de célula por célula.
- Pasarle el estado inicial desde un stream de datos.
- Parametrizar la lógica de las células.
 - Lógica contenida dentro de cada célula.
- Poder resetear la simulación.