

The

Game of

Proyek Akhir PSD

Kelompok 6

Life

Physics Engine

Kelompok 6

Anggota Kelompok



Djukallita Tafiana
Djoewarsa
2406416573



Muhamad Rifqi
Fadil Itsnain
2406355306



Naziehan Labieb
2406487102



Soraya Azzizah
Pahlevi
2406487001

Background

Perkembangan teknologi digital mendorong kebutuhan akan sistem komputasi yang semakin cepat dan efisien. Salah satu konsep penting dalam komputasi modern adalah paralelisme, yaitu kemampuan menjalankan banyak proses secara bersamaan. Untuk mempelajari prinsip ini, cellular automata digunakan sebagai model sederhana namun sangat efektif, dengan Conway's Game of Life menjadi contoh paling populer dalam pendidikan dan penelitian. Melalui model tersebut, konsep interaksi lokal yang menghasilkan perilaku kompleks dapat dipahami dengan lebih intuitif dan praktis.

Project Description

Proyek ini adalah implementasi Hardware Accelerator untuk simulasi Cellular Automata (Conway's Game of Life) menggunakan VHDL. Berbeda dengan simulasi software yang memproses data secara berurutan (sekuensial), sistem ini menggunakan arsitektur Massively Parallel Processing, di mana 64 sel logika menghitung nasibnya secara bersamaan dalam satu siklus clock.

Sistem ini juga dilengkapi dengan CPU Mini (Microprogrammed Controller) yang bertugas menggambar pola awal kehidupan ke dalam Grid sebelum simulasi dijalankan.

Objectives

Kelompok 6

Proyek ini bertujuan memberikan pemahaman mengenai penerapan paralelisme dan cellular automata dalam sistem digital, serta melatih untuk bagaimana merancang dan mengimplementasikan rangkaian berbasis VHDL secara nyata

1. Mempelajari dan menerapkan konsep paralelisme melalui simulasi berbasis hardware.

Grid 8×8 memungkinkan 64 sel bekerja secara simultan, sehingga mahasiswa dapat melihat bagaimana proses paralel terjadi dalam setiap siklus clock melalui implementasi langsung menggunakan VHDL.

2. Merealisasikan cellular automata menjadi sistem digital nyata menggunakan VHDL.

Dengan mengimplementasikan aturan Game of Life pada modul sel dan menyusun grid menggunakan generate loop, mahasiswa dapat memahami bagaimana interaksi antar sel dapat dibangun secara fisik dalam rangkaian digital.

3. Mengembangkan mekanisme pembaruan state sel berbasis topologi toroidal.

Mekanisme ini memastikan setiap sel dapat menghitung delapan tetangganya menggunakan aritmetika modulo, sehingga pembaruan state berlangsung akurat dan konsisten pada setiap siklus clock.

the game of life physics engine

Kelompok 6

IMPLEMENTATION

The Cell Core | The Grid Architecture | Microprogrammed Controller

The Cell Core

- Fungsi Utama:
 - Unit terkecil dalam sistem yang bertindak sebagai "otak" independen.
 - Setiap sel memiliki memori dan logika sendiri untuk menentukan nasibnya.
- Implementasi Modul (2, 3, 7):
 - Modul 2 (Aritmetika): Menjumlahkan sinyal masuk dari 8 tetangga secara real-time (Adder).
 - Modul 3 (Sequential): Menggunakan Register untuk menyimpan status (Hidup/Mati) dan menyinkronkan perubahan dengan Clock.
 - Modul 7 (Function): Logika aturan Game of Life dibungkus dalam fungsi `get_next_state()` agar modular.
- Aturan Logika (The Rules):
 - < 2 Tetangga maka akan Mati (Kesepian).
 - > 3 Tetangga maka akan Mati (Kepadatan).
 - = 3 Tetangga maka akan Lahir (Hidup).
 - = 2 Tetangga maka akan Bertahan (Status Tetap).
- Logika Prioritas (Priority Logic):
 - Sel memiliki Multiplexer logika untuk memilih mode:
 - Mode Edit: Jika CPU mengirim sinyal Write, sel memaksa nilainya berubah (Override).
 - Mode Run: Jika tidak, sel mengikuti aturan Game of Life.

The Grid Architecture

- Dua Komponen Utama:
 - Controller (Otak): Pengendali utama sistem.
 - Game Grid (Arena): Tempat simulasi berlangsung.
- Tugas Controller:
 - Menggambar pola awal (seperti "Glider") ke dalam Grid.
 - Mengirim sinyal untuk memulai simulasi.
- Tugas Game Grid:
 - Menampung 64 sel (8x8).
 - Menghitung nasib setiap sel secara bersamaan (parallel).

Microprogrammed Controller

- Fungsi Utama:
 - Bertindak sebagai "otak" untuk menggambar pola awal ke Grid.
 - Menggunakan memori ROM (Read-Only Memory) sebagai penyimpan instruksi (Control Store).
- Format Instruksi Mikro (8-bit):
 - Instruksi dalam ROM dipecah menjadi sinyal kontrol:
 - 3-bit Row: Koordinat Baris.
 - 3-bit Col: Koordinat Kolom.
 - 1-bit Data: Nilai sel (Hidup/Mati).
 - 1-bit Last Flag: Penanda akhir instruksi.
- Cara Kerja (Fetch-Decode-Execute):
 - Fetch: Controller membaca baris instruksi dari ROM berdasarkan Program Counter.
 - Decode: Memecah 8-bit data menjadi sinyal row, col, dan we (write enable).
 - Execute: Mengirim sinyal tersebut ke Grid untuk menyalakan sel spesifik.
- Keunggulan:
 - Fleksibel: Mengganti pola awal (misal dari Glider ke Blinker) cukup dengan mengubah isi teks di ROM, tanpa merombak rangkaian hardware.

Kelompok 6

Testing

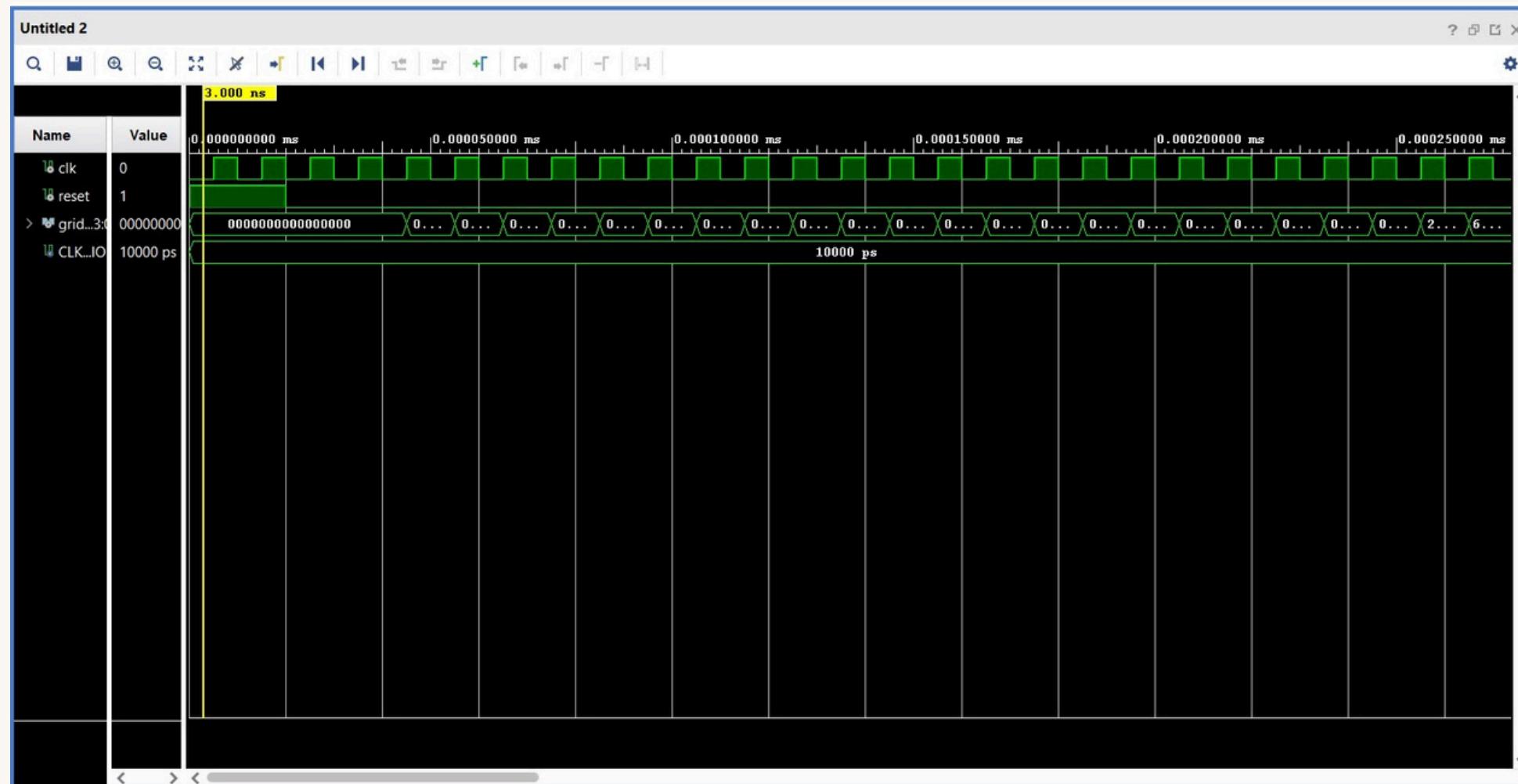
- Pengujian dilakukan untuk memastikan seluruh modul Game of Life bekerja sesuai rancangan.
- Testbench digunakan untuk mensimulasikan berbagai kondisi evolusi sel, dari level sel tunggal hingga sistem penuh.
- Proses uji mengevaluasi respon modul terhadap clock, reset, serta input yang diberikan.
- Pendekatan bertahap memastikan setiap modul sudah berfungsi sebelum digabungkan.

Tahap Testing:

- cell_core: menguji logika aturan Game of Life dengan berbagai kombinasi status tetangga.
- game_grid_8x8: memverifikasi koneksi tetangga toroidal dan update seluruh grid secara serempak.
- game_controller: menguji instruksi SET, CLEAR, STEP, dan RUN untuk memastikan kontrol simulasi berjalan benar.
- top_system: memastikan integrasi penuh; pola awal dibaca dari file dan output grid dicatat kembali dengan benar

Result

Kelompok 6



Top System Waveform

```
FRAME: 1
00000000
00000010
00001100
00000110
00000000
00000000
00000000
00000000
=====
FRAME: 2
00000000
00000100
00001000
00001100
00000000
00000000
00000000
00000000
=====
FRAME: 3
00000000
00000000
00001010
00001100
00000100
00000000
00000000
00000000
=====
FRAME: 4
00000000
00000000
00001000
00001010
00001100
00000000
00000000
00000000
=====
```

Frame Output Text

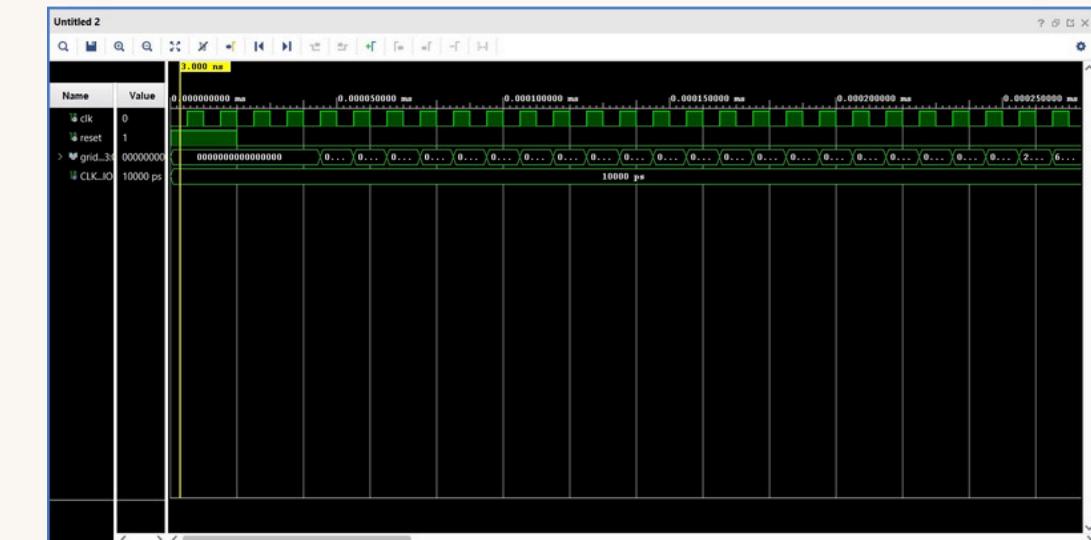
the game of life physics engine

Analysis

Kelompok 6

- Sistem berhasil melakukan sinyal reset dan clock stabil selama simulasi.
- Perubahan nilai grid_out menunjukkan bahwa setiap sel memperbarui state hanya pada rising edge, sesuai aturan Game of Life.
- Koneksi tetangga pada Game_Grid_8x8 bekerja dengan benar, ditunjukkan oleh pola yang berubah serempak di seluruh grid.
- Output file (FRAME 1–4) memperlihatkan evolusi pola yang konsisten dimana saat sel mati/hidup mengikuti aturan underpopulation, overpopulation, survival, dan reproduction.
- Pola berubah secara teratur antar frame → menandakan seluruh sel menerima tetangga yang tepat dan logika cell_core berfungsi.
- Tidak ditemukan glitch atau perubahan nilai di luar clock → sistem berjalan stabil dan sesuai spesifikasi desain.

```
FRAME: 1
00000000
00000010
00001100
00000110
00000000
00000000
00000000
00000000
00000000
=====
FRAME: 2
00000000
00000100
00001000
00001110
00000000
00000000
00000000
00000000
00000000
=====
FRAME: 3
00000000
00000000
00001010
00001100
00000100
00000000
00000000
00000000
00000000
=====
FRAME: 4
00000000
00000000
00001000
00001010
00001100
00000000
00000000
00000000
00000000
=====
```



Conclusion

- Desain berhasil menjalankan cellular automata secara penuh di hardware melalui pendekatan digital paralel, dengan setiap sel bekerja sebagai modul logika independen.
- Sistem mampu memperbarui 64 sel dalam satu siklus clock, menunjukkan efektivitas paralelisme dalam VHDL dan bagaimana logika software dapat diwujudkan dalam rangkaian digital nyata.
- Berbagai konsep VHDL seperti logika kombinasi, sequential logic, generate loop, structural hierarchy, fungsi, prosedur, dan FSM berhasil diterapkan secara terintegrasi dalam satu sistem.
- Penerapan topologi toroidal menjaga konsistensi tetangga setiap sel dan memastikan simulasi berjalan tanpa gangguan batas tepi.
- Pengujian melalui testbench dan file I/O berhasil memverifikasi perubahan state secara detail, memastikan akurasi dan stabilitas keseluruhan sistem.

Kelompok 6

thank you
so much!