

PSD FINAL PROJECT - GROUP 16

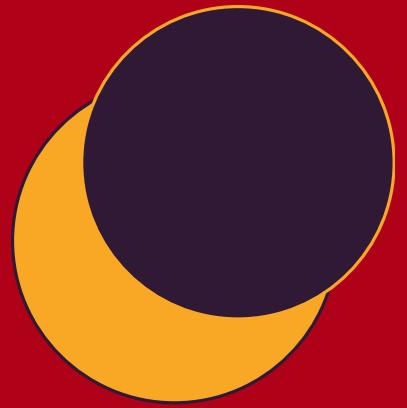
Electronic Slot Machine with Credit Manager

Presented by:

**Muhammad Rafif Batubara
Ayesha Zelene Faeyza
Diandra Pramesti Wicaksono
Radya Gardian Pranoto**

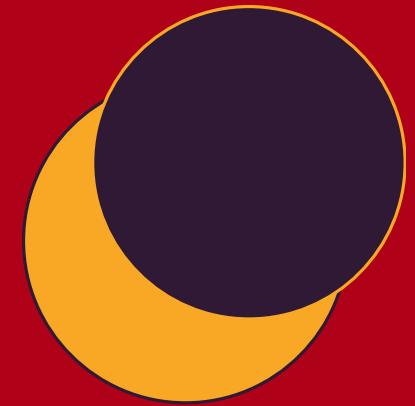


INTRODUCTION



Latar Belakang

Di zaman ini, banyak produk dalam industri gaming yang bergantung pada randomness untuk menciptakan sebuah suasana yang adil dan tidak terprediksi, contohnya dalam permainan gacha, loot yang muncul di tanah, dan sebagainya. Dalam sebuah sistem digital seperti FPGA, menghasilkan angka yang benar-benar acak merupakan tantangan tersendiri karena sifat deterministik dari logika digital. Salah satu solusi yang efisien untuk diimplementasikan adalah dengan Pseudo-Random Number Generator (PRNG).



Deskripsi

Proyek ini berjudul "Electronic Slot Machine with Credit Manager". Sistem ini mensimulasikan sebuah 3-reel slot machine, dimana mesin slot tersebut memiliki tiga item atau gulungan. Pengguna dapat memasukkan sebuah koin virtual untuk menambah kredit. Permainan dimulai ketika pengguna memulai spin, yang akan memutar angka pada reel.

Kami mengimplementasikan modul LFSR (Linear Feedback Shift Register) 16-bit pada proyek ini untuk menentukan hasil dari setiap reel setelah diacak. Fitur unik dari proyek yang kami buat adalah adanya input `admin_cheat`, yang memungkinkan pengguna untuk mendapatkan hasil "Jackpot" (7-7-7) yang dapat digunakan untuk demonstrasi atau debugging. Sistem juga memiliki mekanisme penghitungan hadiah otomatis berdasarkan kombinasi angka yang muncul.

Tujuan Proyek

Mengimplementasikan Finite State Machine (FSM) untuk mengatur alur permainan (Idle, Spinning, Stopping, Evaluate, Payout).

Menerapkan algoritma LFSR untuk menghasilkan angka random pada FPGA.

Menguji coba fitur Admin, dimana digunakan untuk mendapatkan jackpot untuk fungsi demonstrasi dan/atau debugging

IMPLEMENTATION

IMPLEMENTASI MODUL

Modul 2 : Concurrent

- Output seperti credit_saldo & reel_val diperbarui secara concurrent
- Dataflow LSFR : shifting & XOR terjadi secara paralel setiap clock
- UUT, clock process, dan stimulus process berjalan bersama di testbench

Modul 3 : Behavioral Style

- Pengelolaan koin & kredit
- Pergerakan reel saat SPINNING
- Transition state FSM berdasarkan spin_trig dan timer
- Cheat mode pada STOPPING
- Evaluasi kombinasi reel berdasarkan function

Modul 6 : Looping

Looping digunakan di testbench, fungsi pada kode untuk :

- Membaca seluruh perintah pemain secara berurutan
- Menjalankan aksi sesuai command
- Mencatat hasil ke output file

Modul 4 : Testbench & File I/O

Input File : input_cmds.txt

Perintah :

1 X = insert coin

2 0 = spin

3 0 = enable cheat

4 0 = disable cheat

Output File : output_log.txt

Log reel hasil spin, status win/lose, sisa credit

Modul 5 : Types & Array

Type Reel_Array : untuk menyimpan 3 reel sekaligus

State_Type : enum untuk FSM

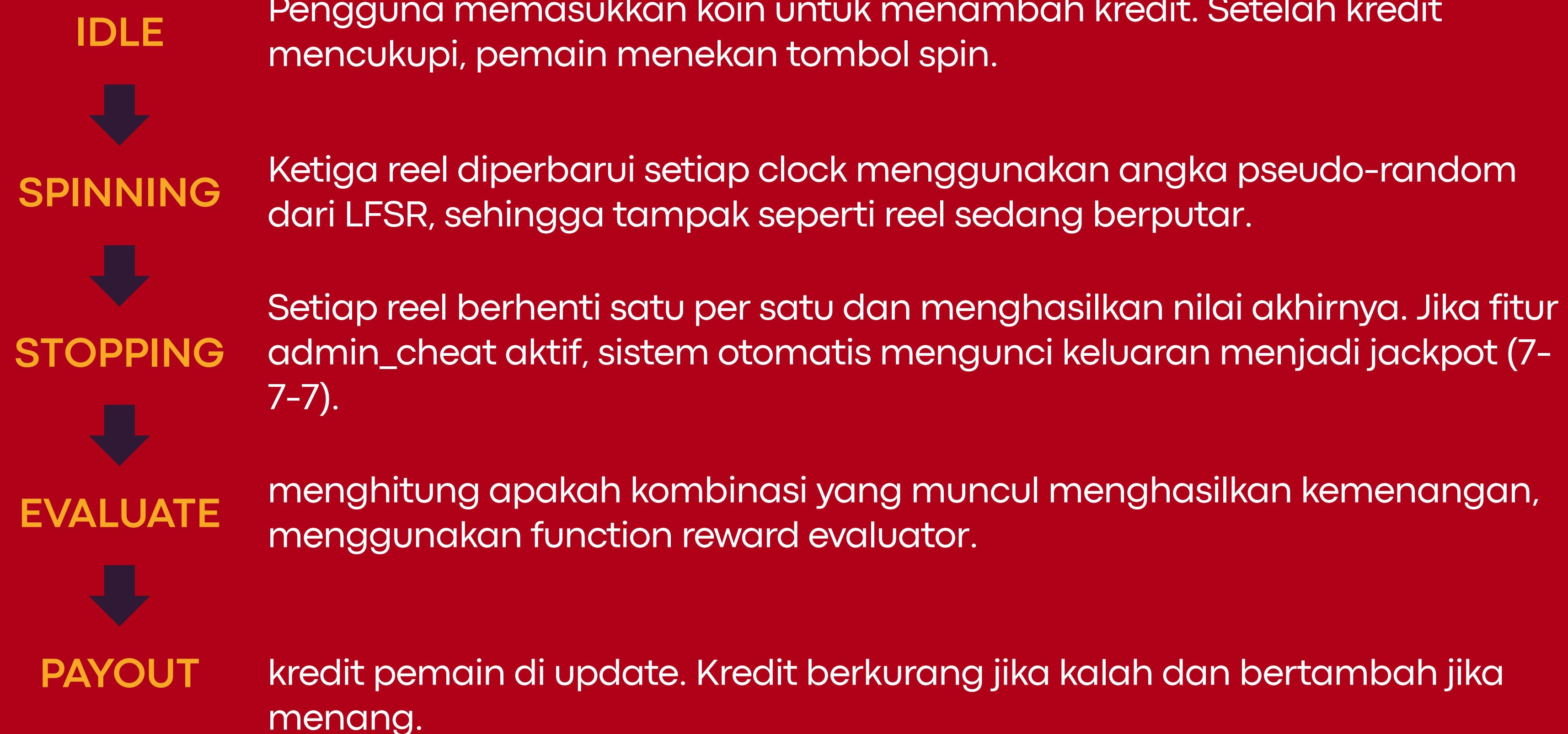
Penggunaannya untuk membuat kode lebih modular dan mudah dibaca

Modul 7 : Function

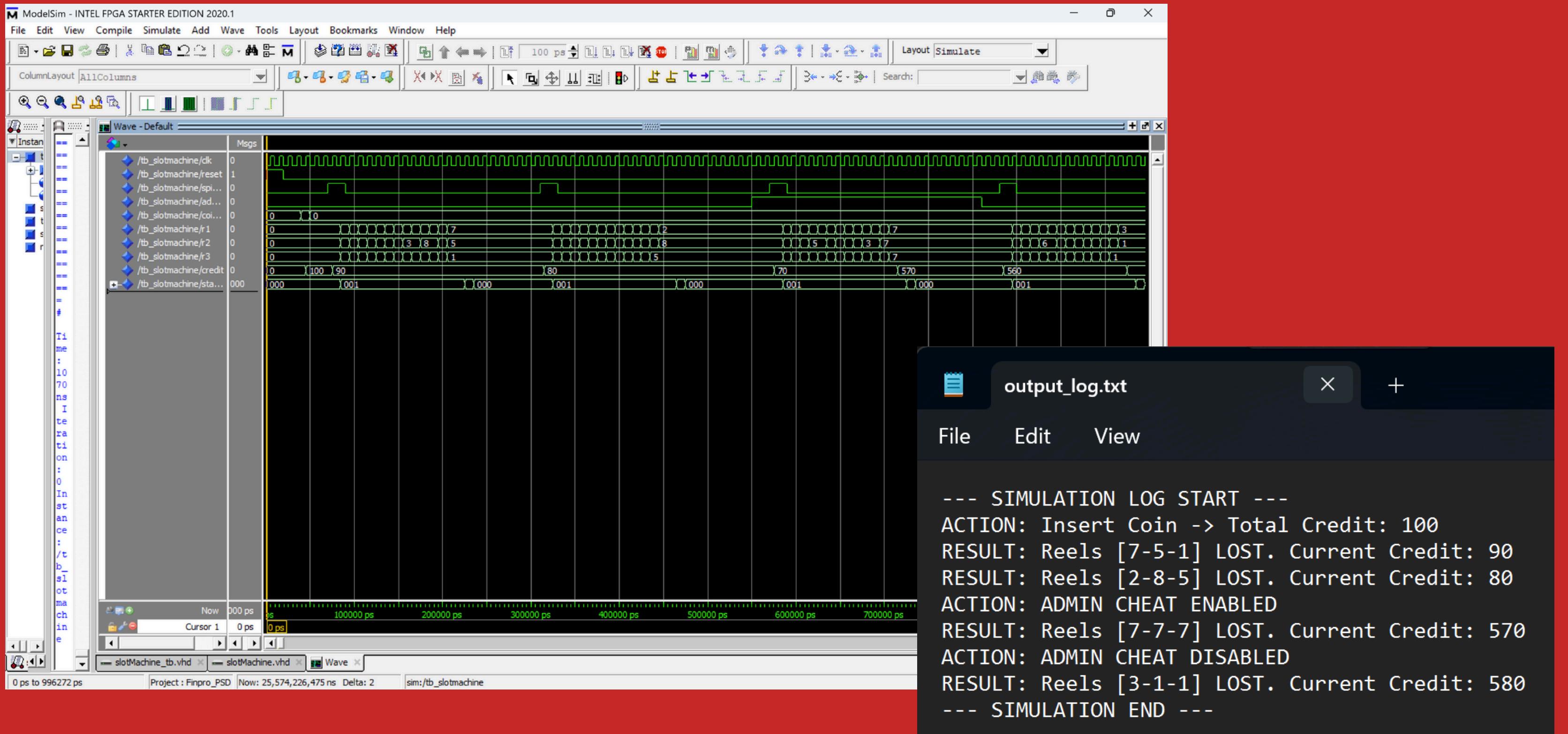
Function yang digunakan :
calculate_reward()

Menentukan reward berdasarkan :

- 777 (jackpot) : 500
- Triple : 100
- Pair : 20
- Lainnya : 0



Result & Testing



TESTBENCH

Inisialisasi

```
signal clk, reset, spin_trig, admin_cheat : STD_LOGIC := '0';
signal coin_in : INTEGER := 0;
signal r1, r2, r3, credit : INTEGER;
signal status : STD_LOGIC_VECTOR(2 downto 0);

constant clk_period : time := 10 ns;
```

Generate Clock

```
clk_process : process
begin
    clk <= '0'; wait for clk_period/2;
    clk <= '1'; wait for clk_period/2;
end process;
```

Reset

```
-- Initial Reset
reset <= '1'; wait for 20 ns;
reset <= '0'; wait for 20 ns;
```

Baca instruksi File

```
file input_file : text open read_mode is "input_cmds.txt";
file output_file : text open write_mode is "output_log.txt";

read(line_in, cmd_code);
read(line_in, val_in);
```

TESTBENCH

Command 1 : Insert Coin

```
when 1 =>
  coin_in <= val_in;
  wait for clk_period;
  coin_in <= 0;
```

Command 2 : Spin Slot

```
spin_trig <= '1';
wait for clk_period * 2;
spin_trig <= '0';

wait for clk_period * 20;
```

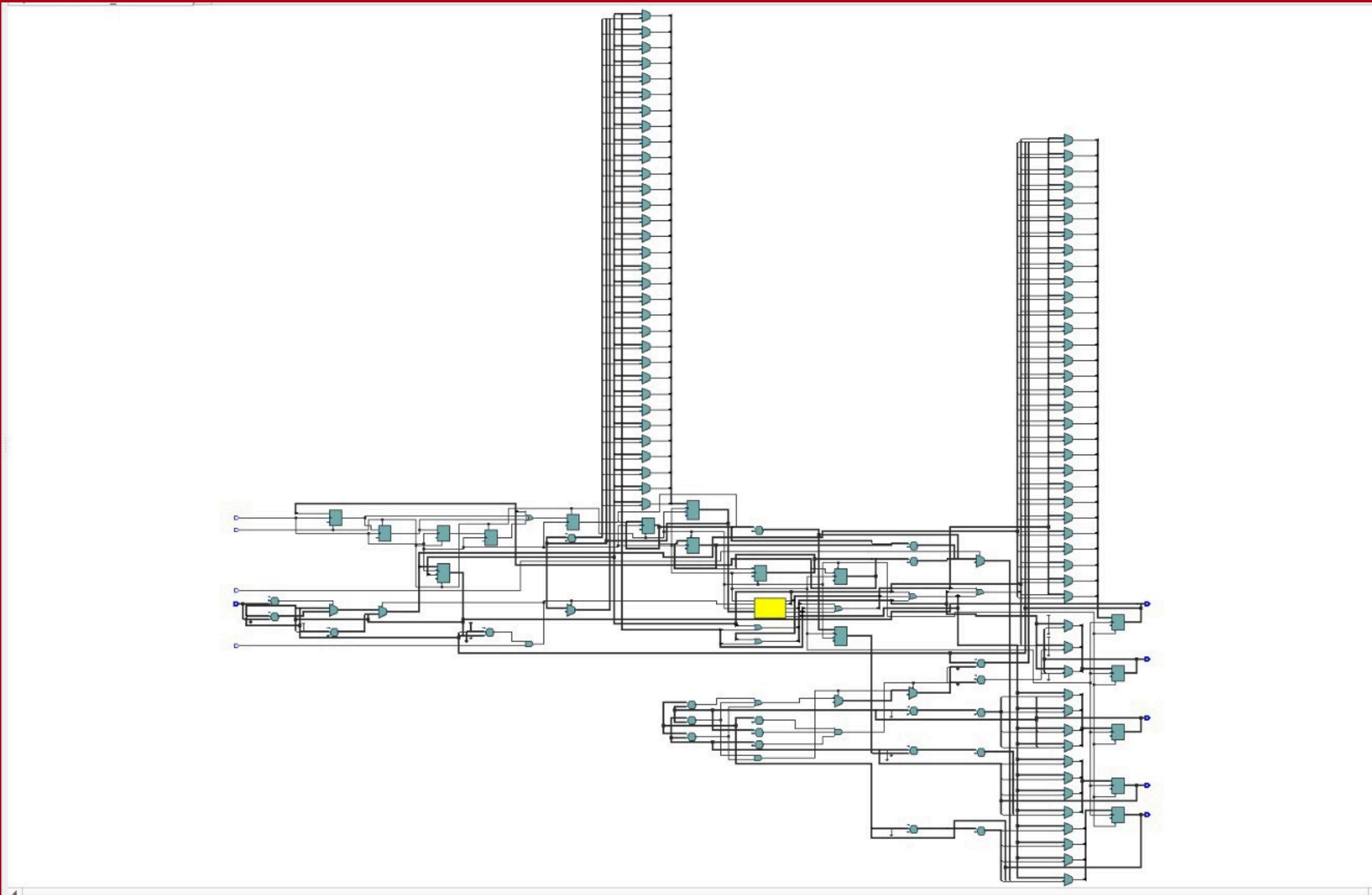
Command 3 : Enable Cheat

```
when 3 =>
  admin_cheat <= '1';
```

Command 4 : Disable Cheat

```
when 4 =>
  admin_cheat <= '0';
```

Results - RTL Design





Thank You