

LAPORAN TUGAS BESAR APLIKASI SISTEM DIGITAL
SMART CAR PARKING SYSTEM
SUB-MODUL *COUNTER PARKING*



Disusun Oleh:

Dhanendra Nivadirrokhman

1103220113

PROGRAM STUDI S1 TEKNIK KOMPUTER
FAKULTAS TEKNIK ELEKTRO UNIVERSITAS TELKOM
2024

A. Pendahuluan

1.1 Latar Belakang

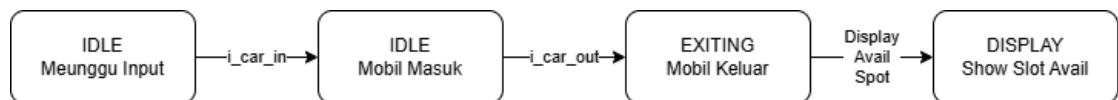
Sistem parkir yang efisien sangat penting dalam mengelola ruang parkir yang terbatas. Dengan meningkatnya jumlah kendaraan, diperlukan sistem yang dapat menghitung dan menampilkan jumlah sisa tempat parkir secara real-time. Laporan ini membahas implementasi modul counter yang digunakan untuk menghitung jumlah sisa parkir. Modul ini dirancang untuk bekerja dengan input berupa sinyal dari sensor parkir yang mendeteksi masuk dan keluarnya kendaraan.

1.2 Tujuan

Sub Modul counter ini berfungsi untuk menghitung jumlah kendaraan yang terparkir dan menampilkan jumlah sisa tempat parkir yang tersedia. Modul ini akan mengatur logika kontrol berdasarkan input dari sensor dan memberikan output yang sesuai untuk menampilkan status parkir.

B. Diagram FSM

2.1 State Utama Dalam Modul Counter



Ada beberapa state utama dalam desain ini, yaitu:

1. **IDLE**: Menunggu input dari sensor.
2. **PARKING**: Mencatat kendaraan yang masuk.
3. **EXITING**: Mencatat kendaraan yang keluar.
4. **DISPLAY**: Menampilkan jumlah sisa tempat parkir.

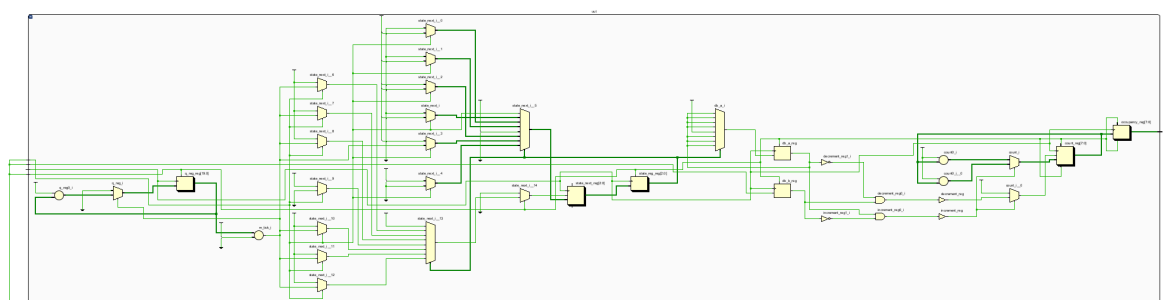
2.2 Flow Diagram



Berdasarkan state yang ada, perubahan flow program berdasarkan pada state dan input. Flow Diagram di bawah menunjukkan bagaimana current state berubah menjadi next state berdasarkan input yang diberikan.

2.3 Block Diagram Counter

Diagram di bawah menunjukkan hubungan antara sinyal input dan output dari modul counter:



C. Source Code

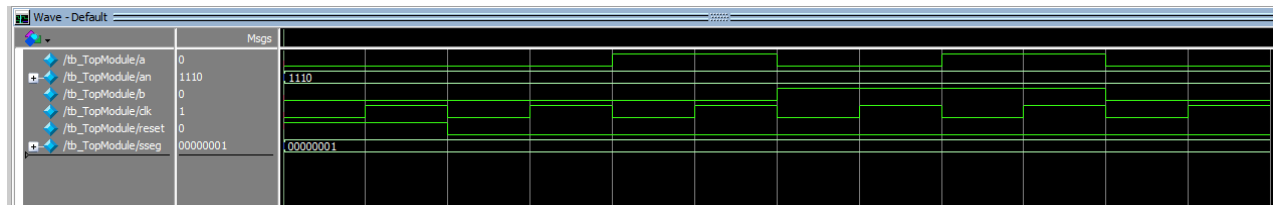
```
module parking_counter(  
    input i_car_in,  
    input i_car_out,  
    input i_reset,  
    input i_clk,  
    output reg [3:0] o_available_spots,  
    output reg o_full,  
    output reg o_empty  
);  
    reg [3:0] total_spots = 4'b1000; // Total 8 spots  
    reg [3:0] current_count;  
  
    always @(posedge i_clk or posedge i_reset) begin  
        if (i_reset) begin  
            current_count <= 0;  
            o_available_spots <= total_spots;  
            o_full <= 0;  
            o_empty <= 1;  
        end else begin  
            if (i_car_in && current_count < total_spots) begin  
                current_count <= current_count + 1;  
            end  
            if (i_car_out && current_count > 0) begin  
                current_count <= current_count - 1;  
            end  
            o_available_spots <= total_spots - current_count;  
            o_full <= (current_count == total_spots);  
            o_empty <= (current_count == 0);  
        end  
    end  
endmodule
```

D. Hardware Implementation

a. Simulation

Proses simulasi dijalankan menggunakan aplikasi Modelsim. Untuk melakukan uji coba pada kode, maka dibuat tes yang dirancang untuk melakukan semua state yang ada secara berurutan:

- Reset -> IDLE -> PARKING (mobil masuk) -> EXITING (mobil keluar) -> DISPLAY (jumlah sisa tempat parkir).

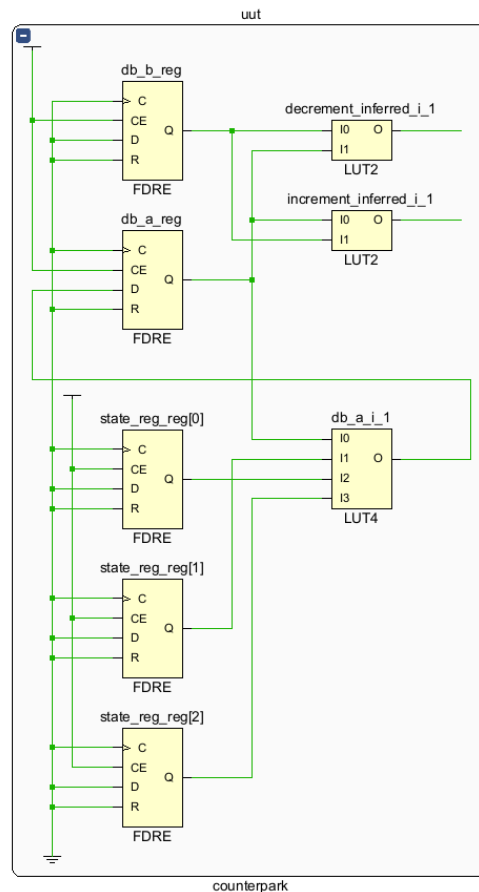


Analisis:

- **Reset ke IDLE:** Ketika reset = 1, jumlah kendaraan terparkir kembali ke 0, dan jumlah sisa tempat parkir ditampilkan sesuai.
- **PARKING (mobil masuk):** Ketika sinyal i_car_in aktif, jumlah kendaraan bertambah, dan jumlah sisa tempat parkir berkurang.
- **EXITING (mobil keluar):** Ketika sinyal i_car_out aktif, jumlah kendaraan berkurang, dan jumlah sisa tempat parkir bertambah.
- **DISPLAY:** Jumlah sisa tempat parkir ditampilkan secara real-time berdasarkan jumlah kendaraan yang terparkir.

b. Hardware Implementation (FPGA)

1. Syntesis Schematic



Modul counterpark menggunakan beberapa D flip-flop (FDRE) untuk menyimpan status sinyal yang relevan. Terdapat dua register, `db_a_reg` dan `db_b_reg`, yang berfungsi untuk menyimpan status debounced dari input a dan b, yang penting untuk mendeteksi pergerakan mobil masuk dan keluar dengan akurat.

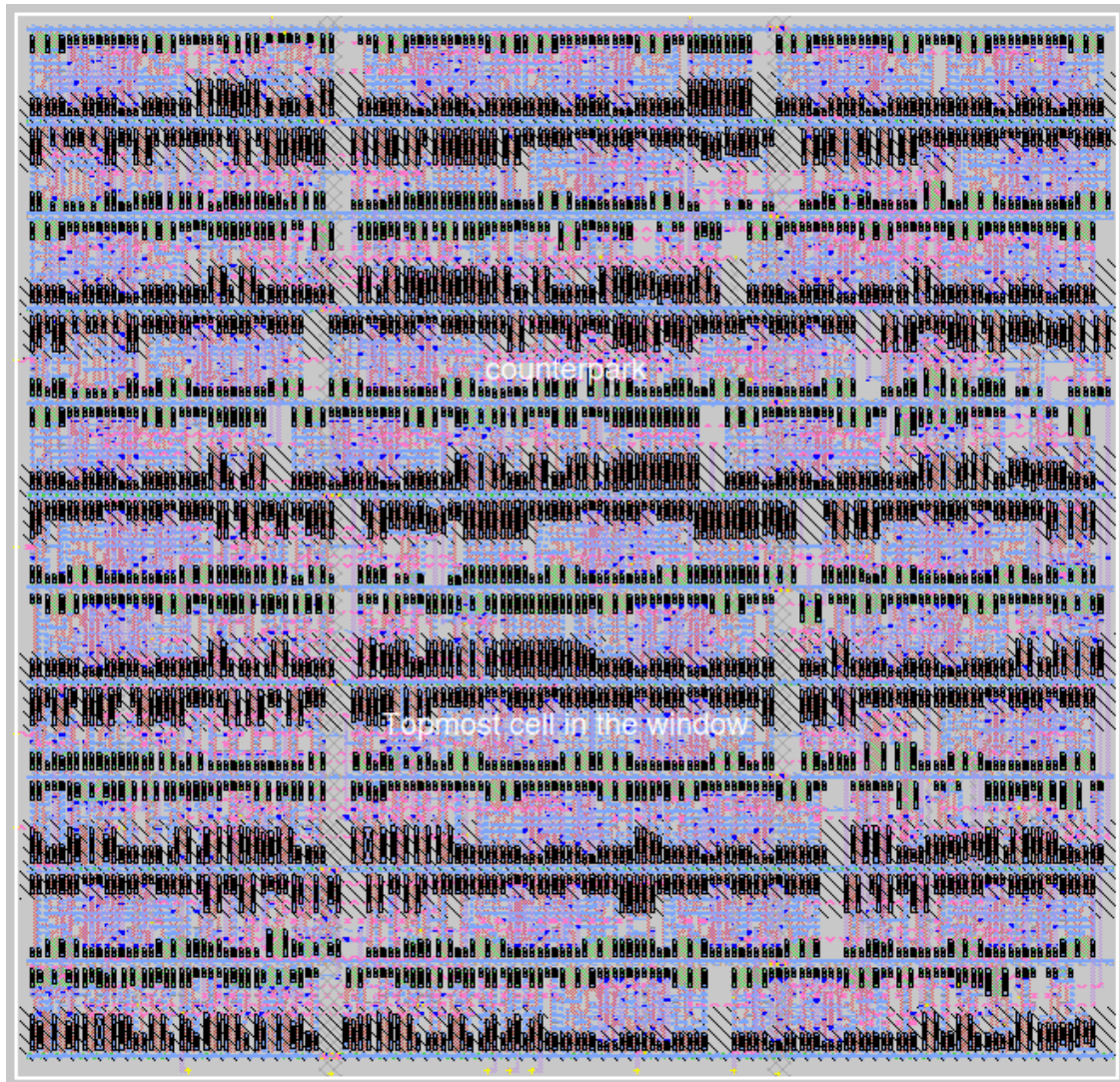
Register `state_reg[0]`, `state_reg[1]`, dan `state_reg[2]` menyimpan status dari state machine yang mengontrol logika counter, memungkinkan sistem untuk beroperasi dalam berbagai kondisi. Logika increment dan decrement terintegrasi untuk mengatur penambahan atau pengurangan nilai counter berdasarkan sinyal input yang diterima.

2. Syntesis Schematic

Total On-Chip Power:	0.068 W
Design Power Budget:	Not Specified
Process:	typical
Power Budget Margin:	N/A
Junction Temperature:	25.3°C
Thermal Margin:	59.7°C (11.9 W)
Ambient Temperature:	25.0 °C
Effective θ_{JA}:	5.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Desain pada board Basys3 menunjukkan total konsumsi daya sebesar 0.068 W, yang efisien. Suhu junction berada di 25.3°C, aman untuk operasi, dengan margin termal yang tinggi (59.7°C) yang menunjukkan kapasitas untuk menangani lebih banyak panas. Tidak ada anggaran daya yang ditentukan, dan tidak ada daya yang disuplai ke perangkat off-chip. Tingkat kepercayaan untuk estimasi daya adalah medium, menunjukkan adanya ketidakpastian yang mungkin memerlukan verifikasi lebih lanjut.

c. Hardware Implementation (ASIC)



1. Timing Analysis

Analisis timing menunjukkan bahwa total 73 jalur telah dianalisis, dengan fokus pada delay maksimum dan minimum untuk memastikan desain memenuhi spesifikasi. Jalur dengan delay maksimum mencapai 1281.11 ps, menunjukkan jalur kritis yang dapat mempengaruhi frekuensi operasi sistem, sedangkan jalur dengan delay minimum adalah 56.7708 ps. Clock skew di beberapa jalur menunjukkan perbedaan waktu antara sinyal clock yang diterima oleh flip-flop, yang dapat mempengaruhi stabilitas sistem. Setup dan hold times yang tercantum menunjukkan bahwa desain memenuhi persyaratan timing, dengan beberapa jalur memiliki margin yang baik, menandakan keandalan dalam pengoperasian sistem.

2. Cell Used

Desain modul counterpark menggunakan berbagai jenis gerbang logika dan flip-flop untuk mengelola status dan kontrol sistem. Terdapat D flip-flops (DFFSR dan DFFPOSX1) yang menyimpan status sinyal dan state machine. Buffer (BUFX4 dan BUFX2) digunakan untuk

memperkuat sinyal, termasuk sinyal clock dan output occupancy. Gerbang logika seperti AND, OR, NOR, NAND, XNOR, dan XOR digunakan untuk mengimplementasikan fungsi logika yang diperlukan dalam penghitungan dan kontrol status. Selain itu, inverter (INV) digunakan untuk membalikkan sinyal, memastikan bahwa logika berfungsi dengan benar dalam berbagai kondisi. Desain ini menunjukkan integrasi yang efisien dari elemen-elemen digital untuk mencapai tujuan sistem parkir.

F. Kesimpulan

Modul counter ini berfungsi sebagai pengatur jumlah kendaraan yang terparkir, memastikan bahwa informasi tentang sisa tempat parkir selalu akurat dan terkini. Berdasarkan hasil simulasi, kode ini telah berjalan sesuai dengan logika yang diharapkan, dan dapat diimplementasikan dalam sistem parkir berbasis FPGA.