



CONSTRUCTION COST PLANNER

BP-02

TABLE OF CONTENTS

About Our Program

Description about our project and tools that being used.

01



02

Component Explanation

Explanation about each components.

Module Implementation

Explanation about each module implementation in our program.

03



04

Testing and Analysis

Explanation about testing result.



05

Schematic Picture

Schematic picture of our program by Quartus.

Anggota Kelompok



**FARUQ SAMI
RAMADHAN**

2206026675



**KEVIN
RAIHAN**

2206059704



**MARIO MATTHEWS
GUNAWAN**

2206810452



**YASMIN DEVINA
SINURAYA**

2206817244



Construction Cost Planner

01

ABOUT OUR PROGRAM

Proyek ini bernama Construction Cost Planner. Program ini bertujuan untuk mengembangkan dan mengimplementasikan sistem berbasis FPGA untuk estimasi biaya konstruksi yang cepat dan akurat. Sistem akan dirancang untuk membantu user menghitung biaya konstruksi untuk 4 tipe bangunan berbeda, yaitu *commercial building*, *industrial building*, *infrastructure building*, dan *residential building*.

01 EQUIPMENT



VS CODE



ModelSim



Quartus



Github



02 COMPONENT EXPLANATION

A. Decoder

Component Decoder berfungsi untuk me-*decode instruction* yang di-*input*, dan *output* dari component ini yaitu hasil *decode instruction* tersebut. *Instruction* terdiri dari 7 buah bit. Berikut adalah tabel hasil *decoding* yang dilakukan oleh component Decoder:

Bit ke-6 dan 5	Bit ke-4 dan 3	Bit ke-2 dan 1	Bit ke-0
Opcode untuk memilih tipe building	Variabel pilihan pertama	Variable pilihan kedua	Condition bit penggunaan harga tambahan

02 COMPONENT EXPLANATION

B. Calculator

Component Calculator memiliki fungsi utama untuk melakukan perhitungan total biaya untuk tiap jenis bangunan. *Component* ini membutuhkan input berupa **PRG_CNT**(program counter), *opcode*, *operand-operand*, dan *cost* yang dibutuhkan. Sedangkan *output component* ini berupa hasil perhitungan total biaya yang dibutuhkan. Perhitungan biaya ditentukan dengan input *opcode*. Dimana :

- Opcode 00 untuk *industrial building*
- Opcode 01 untuk *residential building*
- Opcode 10 untuk *commercial building*
- Opcode 11 untuk *infrastructure building*

02 COMPONENT EXPLANATION

C. CostPlanner

Component CostPlanner berfungsi sebagai **main program**, karena *component* lain seperti **Calculator** dan **Decoder** digunakan di *component* ini. Di dalam *component* ini terdapat 5 buah *state* untuk menentukan proses apa yang sedang dilakukan oleh program. Berikut penjelasan singkat dari setiap *state*:

- State **IDLE** artinya program akan menunggu hingga enable bernilai 1.
- State **FETCH** artinya program akan menerima *instruction input* dan *program counter* bertambah 1.
- State **DECODE** artinya program memberikan parameter-parameter yang dibutuhkan oleh *component Decode* untuk melakukan *decoding* pada *instruction input*.
- State **EXECUTE** artinya program akan memberikan parameter-parameter yang dibutuhkan oleh *component Calculator* untuk melakukan perhitungan terhadap biaya pembangunan.
- State **COMPLETE** artinya program sudah selesai melakukan perhitungan.

03 Module Implementation

Modul 2

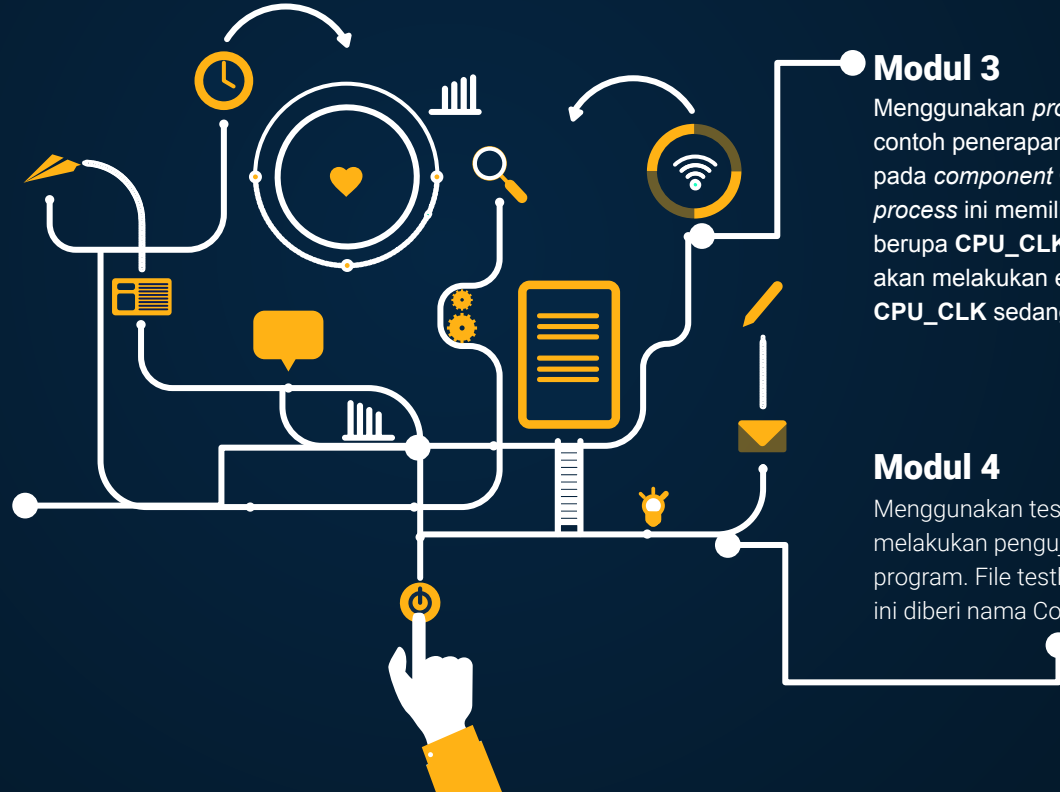
Menggunakan *conditional signal assignment* dan *selected signal assignment* selama proses perhitungan biaya. *Conditional signal assignment (if-else)* digunakan di dalam *function* perhitungan biaya, dan *selected signal assignment (switch-case)* digunakan pada kondisi state program.

Modul 3

Menggunakan *process*. Salah satu contoh penerapan *process* yaitu pada *component CostPlanner*, *process* ini memiliki *sensitivity list* berupa **CPU_CLK**. *Process* ini akan melakukan execution ketika **CPU_CLK** sedang *rising edge*.

Modul 4

Menggunakan testbench untuk melakukan pengujian terhadap program. File testbench program ini diberi nama *CostPlanner_tb*.



03 Module Implementation

Modul 5

Menggunakan *component* **Decoder** dan **Calculator**, dan menggunakan *portmap* pada **CostPlanner** untuk menghubungkan *component* **Decoder** dan **Calculator** agar dapat digunakan.

Modul 7

Menggunakan *function* di dalam *component* **Calculator**. Salah satu contoh penggunaan *function* yaitu *function* bernama **ApplySafetyCost**. *Function* tersebut berfungsi untuk menentukan apakah *user* ingin menggunakan *safety* lebih atau tidak, jika iya maka akan dikenakan biaya tambahan untuk *safety cost*.



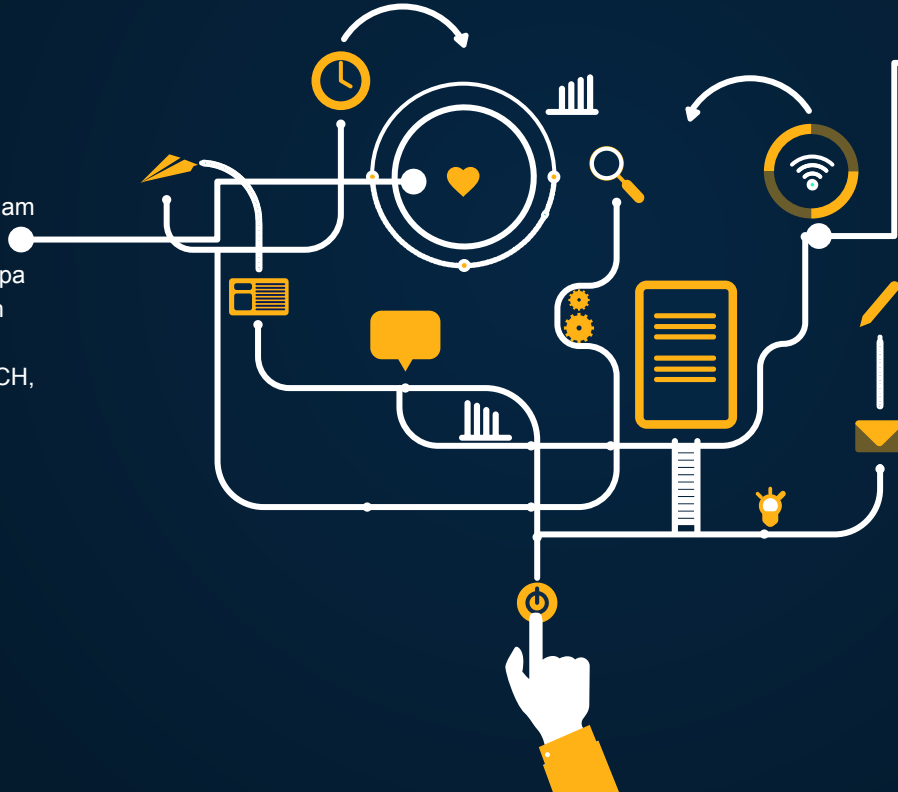
03 Module Implementation

Modul 8

Menggunakan 5 *state* di dalam *component CostPlanner* sebagai penentu instruksi apa yang sedang dilakukan oleh program. 5 *state* yang digunakan yaitu IDLE, FETCH, DECODE, EXECUTE, dan COMPLETE.

Modul 9

Menggunakan *component* dan input dalam bentuk *opcode*. *Opcode* ini yang akan menentukan tipe bangunan apa yang diinginkan, dan *operand* apa saja yang akan digunakan sesuai dengan tipe bangunannya. Penentuan *opcode* ini sesuai dengan hasil *decode* dari *instruction input* yang diberikan.



04 TESTING AND ANALYSIS

Testing dilakukan dengan 4 opcode yang berbeda yaitu 0010011, 0111000, 1011010, 1100111. Masing-masing bertujuan untuk mensimulasikan tiap tipe konstruksi (Industrial, Residential, Commercial, Infrastructure) dengan variabel dan condition bit yang disesuaikan.

```
-- Stimulus process
stimulus : process
constant period: time := 10 ns;
begin
    -- Initialize inputs
    enable_tb <= '1';
    instruction_tb <= "0010011"; -- Set an example instruction
    Var1_tb <= 10; -- production capacity
    Var2_tb <= 200; -- building area
    Var3_tb <= 5; -- equipment quantity
    wait for period;

    enable_tb <= '1';
    instruction_tb <= "0111000"; -- Set an example instruction
    Var1_tb <= 3; -- tingkat bangunan
    Var2_tb <= 200; -- luas tanah
    Var3_tb <= 0; -- dont care
    wait for period;

    enable_tb <= '1';
    instruction_tb <= "1011010"; -- Set an example instruction
    Var1_tb <= 200; -- luas tanah
    Var2_tb <= 10; -- area per customer
    Var3_tb <= 20; -- jumlah customer
    wait for period;

    enable_tb <= '1';
    instruction_tb <= "1100111"; -- Set an example instruction
    Var1_tb <= 500; -- project length
    Var2_tb <= 70; -- unit cost per meter
    Var3_tb <= 100; -- material cost
    wait for period;
    wait; -- End simulation
end process;
```

04 TESTING AND ANALYSIS

	Msgs						
/costplanner_tb/CPU_CLK_tb	0						
/costplanner_tb/enable_tb	1						
/costplanner_tb/instruction_tb	0111000	0010011	0111000	1011010	1100111		
/costplanner_tb/Var1_tb	3	10	3	200	500		
/costplanner_tb/Var2_tb	200	200		10	70		
/costplanner_tb/Var3_tb	0	5	0	20	100		
/costplanner_tb/biaya_tb	82600	207800	82600	127400	25500		

Opcode pertama 0010011, memiliki spesifikasi tipe konstruksi Industrial, unit cost seharga 80\$ per unit, biaya peralatan 1000\$ per alat, dan biaya regulasi keamanan dan kesehatan pabrik (42000\$). Opcode kedua 0111000, memiliki spesifikasi konstruksi Residential, harga material lantai granit (31\$) per meter kuadrat, biaya tambahan halaman (1000\$). Opcode ketiga 1011010, memiliki spesifikasi tipe konstruksi Commercial, biaya material lantai granit (31\$) per meter kuadrat, dan biaya tambahan patung commercial (1200\$). Opcode keempat 1100111, memiliki spesifikasi tipe konstruksi Infrastructure, dengan variabel multiplier kompleksitas (x0), multiplier biaya contingency (x1.5), dan multiplier environmental impact (x2).

Input operasi pertama menghasilkan 207800\$, operasi kedua menghasilkan 82500\$, operasi ketiga menghasilkan 127400\$, dan operasi keempat menghasilkan 25500\$.

04 TESTING AND ANALYSIS

Hasil perhitungan menggunakan kalkulator

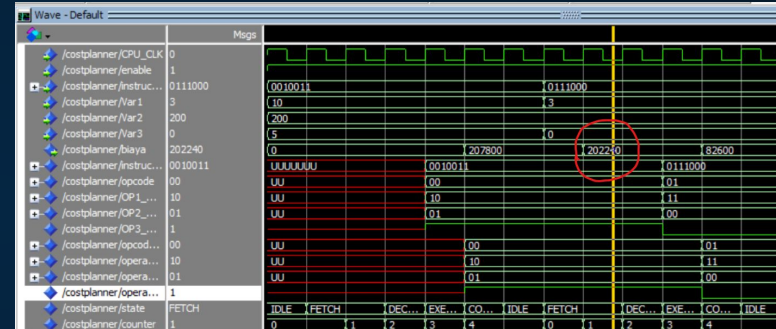
$$(10 \cdot 80) + (200 \cdot 800) + (5 \cdot 1000) + 42000 = 207800$$

$$(3 \cdot 1000) + (200 \cdot 300) + (3 \cdot 200 \cdot 31) + 1000 = 82600$$

$$200 \cdot (300 + 31) + (10 \cdot 20 \cdot 300) + 1200 + 0 = 127400$$

$$500 \cdot \left(\left(70 \cdot \frac{10}{10} \right) + 100 \right) \cdot 2 \cdot \frac{15}{100} = 25500$$

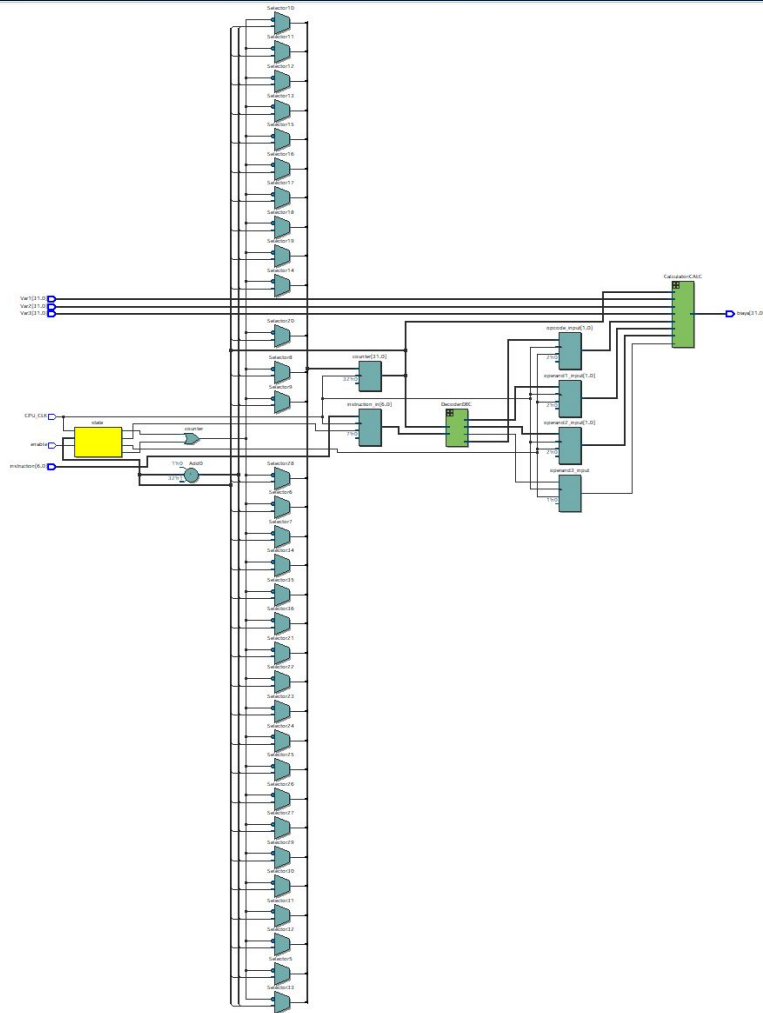
Hasil perhitungan menggunakan program



Dapat kita pastikan bahwa value output program sudah sesuai dari ekspektasi. Tidak terdapat kesalahan logika maupun syntax dalam hasil akhirnya.

Namun terdapat kesalahan dalam output program ketika tengah penjalanan program. Sebelum hasil akhir keluar terdapat value yang tidak diketahui asalnya. Hal ini tidak mempengaruhi tujuan program karena hasil akhir masih valid namun lebih baik jika kita dapat menanggulangi kesalahan ini pada lain waktu.

05 SCHEMATIC PICTURE



Schematic picture is generated using
Quartus

The background is a dark navy blue. In the center is a large, solid orange rectangle. The words "THANK YOU" are written in a bold, dark blue, sans-serif font across the middle of this rectangle. Surrounding the rectangle are various geometric elements: a small yellow gear with a circular outline above it at the top center; a large yellow gear outline at the bottom left; a white gear with a circular outline at the bottom right; and several horizontal white and yellow lines of varying lengths at the top left and top right. A thin yellow line also extends from the right side of the orange rectangle.

THANK YOU