

**PERANCANGAN DAN SIMULASI PERLINDUNGAN PROPERTI  
INTELEKTUAL MENGGUNAKAN ALGORITME *OBfuscation*  
FILTER DIGITAL**

**PROPOSAL TUGAS AKHIR**

Kelompok Kompetensi : Arsitektur Komputer

oleh

**HANJARA CAHYA ADHYATMA  
1104131113**



**FAKULTAS TEKNIK ELEKTRO  
UNIVERSITAS TELKOM  
BANDUNG  
2017**

**PERANCANGAN DAN SIMULASI PERLINDUNGAN PROPERTI  
INTELEKTUAL MENGGUNAKAN ALGORITME *OBfuscation*  
FILTER DIGITAL**

**PROPOSAL TUGAS AKHIR**

Kelompok Kompetensi : Arsitektur Komputer

oleh

**HANJARA CAHYA ADHYATMA  
1104131113**



**FAKULTAS TEKNIK ELEKTRO  
UNIVERSITAS TELKOM  
BANDUNG  
2017**

## HALAMAN PENGESAHAN

PROPOSAL TUGAS AKHIR ini diajukan oleh :  
Nama : Hanjara Cahya Adhyatma  
NIM : 1104131113  
Program Studi : S1 Sitemns Komputers  
Judul PROPOSAL TUGAS AKHIR : PERANCANGAN DAN SIMULASI  
PERLINDUNGAN PROPERTI INTELEKTUAL  
MENGUNAKAN ALGORITME FILTER DIGITAL

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi S1 Sitemns Komputers, Fakultas Tekniks Elektos, Universitas Indonesia.**

## DEWAN PENGUJI

Pembimbing : Prof. XXXX ( )

Penguji : Prof. XXX ( )

Penguji : Prof. XXXX ( )

Penguji : Prof. XXXXXX ( )

**@todo**

Jangan lupa mengisi nama para penguji.

Ditetapkan di : Depok

Tanggal : XX September 2017

## **HALAMAN PERNYATAAN ORISINALITAS**

NAMA : Hanjara Cahya Adhyatma  
NIM : 1104131113  
ALAMAT : Komplek BPI Blok E1 No. 16 RT 05 RW 06 Kabupaten  
Pandeglang, Banten  
No. TLP/HP : +6285201740588  
E-MAIL : adhyatma.han@gmail.com

Menyatakan bahwa Tugas Akhir II ini merupakan karya orisinal saya sendiri dengan judul:

### **PERANCANGAN DAN SIMULASI PERLINDUNGAN PROPERTI INTELEKTUAL MENGUNAKAN ALGORITME FILTER DIGITAL**

*DESIGN AND SIMULATION OF INTELLECTUAL PROPERTIES PROTECTION  
USING DIGITAL FILTER ALGORITHM*

Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidak aslian karya ini.

## **KATA PENGANTAR**

Puji syukur terhadap Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah Nya serta nikmat sehat dan nikmat waktu sehingga buku ini dapat diselesaikan. Ucapan terima kasih juga diperuntukkan untuk orang tua dan saudara saudara saya yang telah memberikan semangat, serta teman-teman yang telah membantu dalam pengerjaan buku ini. Ucapan terima kasih juga diperuntukan kepada Dosen-dosen pembimbing Tugas Akhir Telkom University yang memberikan masukan dan saran terhadap buku ini.

Buku penelitian ini bertujuan untuk mengembangkan ilmu teknologi serta keamanan dalam bidang System-on-Chip (SoC) yang masih jarang dikembangkan di Indonesia.

Bandung, 1 Juli 2017

Hanjara Cahya Adhyatma

## ABSTRAK

System on a Chip (SoC) adalah sebuah modul embedded system yang memiliki fungsi tertentu dalam sebuah papan chip silicon yang juga bisa disebut dengan Very Large Scale Integration (VLSI). Pemilik dari desain SoC memiliki hak cipta atas desain sistem yang telah dibuat. Fabless manufacturing merupakan cara pencetakan modul perangkat keras yang desainer Integrated Circuit (IC) adalah Outsourcing dari luar pabrik percetakan.

Fabless manufacturing dari desain IC memiliki celah pencurian desain ketika desain akan dicetak atau ketika proyek membutuhkan multiple module dengan berbagai fungsi dari berbagai desainer. Oleh karena itu setiap modul VLSI dari desainer chip ini membutuhkan bukti ownership dari perancang atau perusahaan produksi.

Dalam penelitian ini berencana membuat rancangan verifikasi ownership dengan 2 kunci khusus verifikasi yaitu Polygate sebagai kunci utama yang akan mengaktifkan kunci kedua, dan kunci kedua akan aktif yang prosesnya menggunakan algoritme filter digital.

**Kata Kunci:** VLSI, Intellectual Property Protection, Digital Signal Processing, Polygate Watermark.

## ABSTRACT

System on a Chip (SoC) is an embedded system module Has a certain functionality in a silicon chip board that can also be called With Veri Large Scale Integration (VLSI). The owner of the SoC design has Copyright over the system design that has been created. Fabless manufacturing is How to mold a hardware module that is designer Integrated Circuit (IC) Is Outsourcing from outside the printing factory.

Fabless manufacturing from IC design has gap design theft When the design will be printed or when the project requires mutiple module With various functions from various designers. Therefore every module is VLSI From this chip designer requires proof of ownership from the designer or Production company.

In this study plans to make a verification of ownership design With 2 dedicated verification keys ie Polygate as the primary key going Activate the second key, and the second key will be active which process Using a digital filter algorithm.

**Keywords:** VLSI, Intellectual Property Protection, Digital Signal Processing, Polygate Watermark.

## DAFTAR ISI

|  |             |
|--|-------------|
| <b>HALAMAN JUDUL</b>                                   | <b>i</b>    |
| <b>LEMBAR PENGESAHAN</b>                               | <b>ii</b>   |
| <b>LEMBAR PERNYATAAN ORISINALITAS</b>                  | <b>iii</b>  |
| <b>KATA PENGANTAR</b>                                  | <b>iv</b>   |
| <b>ABSTRAK</b>   | <b>v</b>    |
| <b>Daftar Isi</b>                                      | <b>vii</b>  |
| <b>Daftar Gambar</b>                                   | <b>viii</b> |
| <b>Daftar Tabel</b>                                    | <b>ix</b>   |
| <b>1 PENDAHULUAN</b>                                   | <b>1</b>    |
| 1.1 Latar Belakang . . . . .                           | 1           |
| 1.2 Permasalahan . . . . .                             | 2           |
| 1.2.1 Rumusan Masalah . . . . .                        | 2           |
| 1.2.2 Batasan Permasalahan . . . . .                   | 2           |
| 1.3 Tujuan . . . . .                                   | 3           |
| 1.4 Metodologi Penelitian . . . . .                    | 3           |
| 1.5 Sistematika Penulisan . . . . .                    | 3           |
| <b>2 TINJAUAN PUSTAKA</b>                              | <b>5</b>    |
| 2.1 Very Large Scale Integration . . . . .             | 5           |
| 2.1.1 Arus Pengembangan LSI . . . . .                  | 5           |
| 2.1.2 Kemungkinan Serangan Desain LSI . . . . .        | 7           |
| 2.1.3 Mengatasi Serangan terhadap Desain LSI . . . . . | 8           |
| 2.2 Teknik Proteksi . . . . .                          | 9           |
| 2.2.1 Digital Signal Processing Filter . . . . .       | 9           |
| 2.2.2 Polimorphisme Gate . . . . .                     | 9           |
| 2.3 Peralatan dan Teknologi . . . . .                  | 10          |
| 2.3.1 Verilog HDL . . . . .                            | 10          |



|          |                                      |           |
|----------|--------------------------------------|-----------|
| 2.3.2    | Yosys Open SYnthesis Suite . . . . . | 10        |
| 2.3.3    | Xilinx ISE Design Suit . . . . .     | 11        |
| 2.3.4    | FPGA Elbert V2 Board . . . . .       | 11        |
| 2.4      | Target IP Core . . . . .             | 12        |
| 2.4.1    | Aritmatic Logic Unit (ALU) . . . . . | 12        |
| <b>3</b> | <b>DESAIN DAN SIMULASI</b>           | <b>14</b> |
| 3.1      | Perancangan Desain . . . . .         | 14        |
| 3.1.1    | Skema Perlindungan . . . . .         | 15        |
| 3.1.2    | Spesifikasi . . . . .                | 18        |
| 3.2      | Alur Proses Pengembangan . . . . .   | 18        |
| 3.3      | Simulasi . . . . .                   | 20        |
| <b>4</b> | <b>PENGUJIAN DAN ANALISIS</b>        | <b>21</b> |
| 4.1      | Pengujian . . . . .                  | 21        |
| 4.1.1    | Sekenario Pengujian . . . . .        | 21        |
| 4.1.2    | Hasil Pengujian . . . . .            | 21        |
| 4.2      | Analisis . . . . .                   | 24        |
| <b>5</b> | <b>KESIMPULAN DAN SARAN</b>          | <b>26</b> |
| 5.1      | Kesimpulan . . . . .                 | 26        |
| 5.2      | Saran . . . . .                      | 26        |
|          | <b>Daftar Referensi</b>              | <b>27</b> |
|          | <b>LAMPIRAN</b>                      | <b>1</b>  |
|          | <b>Lampiran 1 Datasheet</b>          | <b>2</b>  |
|          | <b>Lampiran 2 Test Bench</b>         | <b>9</b>  |
|          | <b>Lampiran 3 RTL Design</b>         | <b>14</b> |

## DAFTAR GAMBAR

|      |   |    |
|------|---|----|
| 2.1  | Produksi Chip Moderen . . . . .                                   | 6  |
| 2.2  | Clonning/Sumber Tidak Terpercaya . . . . .                        | 7  |
| 2.3  | RE (Reverse Engineering) . . . . .                                | 7  |
| 2.4  | Model Bisnis Lama . . . . .                                       | 8  |
| 2.5  | Model Bisnis Baru . . . . .                                       | 8  |
| 2.6  | Polymorph gate . . . . .  | 10 |
| 2.7  | Perbedaan Tinkatan Abstraksi dan Sintesis Yosys . . . . .         | 11 |
| 2.8  | Logo Xilinx ISE Design Suit . . . . .                             | 11 |
| 2.9  | FPGA Board - Elbert V2 . . . . .                                  | 12 |
| 2.10 | ALU . . . . .   | 13 |
| 3.1  | Desain ALU yang akan dilindungi . . . . .                         | 14 |
| 3.2  | Desain rangkaian pelindung . . . . .                              | 14 |
| 3.3  | Desain rangkaian Top modul yang terdapat rangkaian lain . . . . . | 15 |
| 3.4  | Desain rangkaian ALU pada top modul . . . . .                     | 15 |
| 3.5  | Desain rangkaian pelindung pada top modul . . . . .               | 16 |
| 3.6  | Algoritma aktivasi . . . . .                                      | 17 |
| 3.7  | Desain rangkaian top modul yang telah diberi pelindung . . . . .  | 17 |
| 3.8  | Skema Perancangan Umum Proses Desain . . . . .                    | 18 |
| 3.9  | Skema kegiatan A . . . . .  | 19 |
| 3.10 | Skema kegiatan B . . . . .  | 19 |
| 3.11 | Skema kegiatan C . . . . .  | 20 |
| 3.12 | Skema kegiatan D . . . . .  | 20 |

## DAFTAR TABEL

|     |  |    |
|-----|--|----|
| 4.1 | Data Mentah . . . . .                                      | 22 |
| 4.2 | Analisis Data Mentah . . . . .                             | 23 |
| 4.3 | FPGA Speed Analysis . . . . .                              | 24 |
| 4.4 | On-Chip Power Summary . . . . .                            | 24 |
| 4.5 | Power Supply Currents . . . . .                            | 25 |
| 4.6 | Gate yang digunakan setelah kompilasi ke netlist . . . . . | 25 |

# **BAB 1**

## **PENDAHULUAN**

Membuat desain IC membutuhkan sumber daya yang sangat banyak, serta prosedur dan ketelitian yang tinggi. Oleh karena itu dalam prosesnya dibutuhkan pengamanan agar desain tidak mudah dicuri yang akan menimbulkan kerugian bagi produsen IC tersebut.

### **1.1 Latar Belakang**

Integrated Circuit (IC) merupakan modul teknologi dasar dari perangkat elektronika tertanam modern. Dengan berkembangnya teknologi IC yang mengutamakan ukuran kecil, dan performa yang tinggi serta dengan harga yang murah membuat teknologi IC semakin diminati [1].

Dengan ukuran modul yang sangat kecil dan banyaknya komponen pembangun, kerja sama antara desainer dilakukan untuk membangun sebuah modul VLSI sehingga setiap desainer dapat fokus mendesain salah satu fungsi yang terdapat dalam modul tersebut. Kerja sama dilakukan untuk mempermudah pembuatan desain VLSI yang memiliki tingkat kerumitan yang tinggi. Desainer juga dapat mempercepat waktu mendesain dengan menggunakan kode sumber yang sudah ada atau bekerja sama secara paralel membuat masing-masing modul yang nantinya akan digabung menjadi sebuah modul utama VLSI.

Setelah modul selesai dibuat maka modul siap untuk di-produksi. Dalam proses produksi modul perusahaan tempat desainer bekerja tidak perlu memiliki pabrik produksi modul sendiri, perusahaan dapat bekerja sama dengan mitra percetakan yang akan memproduksi modul buatan perusahaan modul tersebut. Cara kerja sama seperti ini disebut dengan Fabless Manufacturing [2]. Ketika akan memproduksi IC, perusahaan harus menyerahkan blueprint modul VLSI ke percetakan, namun blueprint tersebut tidak terjamin kerahasiaannya serta memungkinkan plagiarisme desain oleh oknum perusahaan atau pihak ketiga yang tertarik menggunakan desain VLSI yang telah diserahkan untuk di-produksi.

Dengan memberikan rangkaian watermark sebagai pengamanan pada blueprint VLSI siap cetak yang menandakan kepemilikan dari desainer atau perusahaan produsen modul akan melindungi dari kecurangan pihak lain yang akan mencuri desain. Sehingga kemungkinan pencurian atau plagiarisme yang menyebabkan ke-

rugian pada perusahaan atau desainer karena desain nya dicuri atau di-plagiat berkurang

## **1.2 Permasalahan**

Pada bagian ini akan dijelaskan mengenai definisi permasalahan yang Hanjara Cahya Adhyatma hadapi dan ingin diselesaikan serta asumsi dan batasan yang digunakan dalam menyelesaikannya. Berikut ini dijelaskan rumusan masalah yang dihadapi dalam penelitian Intellectual Property Protection (IPP) menggunakan metode Digital Filter Algorithm :

### **1.2.1 Rumusan Masalah**

Berikut ini dijelaskan rumusan masalah yang dihadapi dalam penelitian Intellectual Property Protection (IPP) menggunakan metode Digital Filter Algorithm :

1. Dengan metode Fabless Manufacturing, desain modul yang siap diproduksi diserahkan kepada perusahaan percetakan mitra sehingga mitra dapat mengetahui desain modul dari desainer yang memungkinkan desain dapat dicuri oleh oknum percetakan atau pihak ketiga yang tertarik dengan desain tersebut.
2. Desain modul rawan terhadap plagiarisme karena desain elektronik sangat mudah ditiru, sehingga pengamanan desain harus dilakukan agar desain tidak mudah untuk dicuri atau di-plagiat.
3. Apabila pihak ketiga mencuri desain, desainer dapat mengklaim modul tersebut dengan bukti dari pengamanan watermark yang telah tertanam dalam IC menggunakan teknik pemanggilan watermark yang hanya diketahui oleh desainer yang mendesain IC tersebut.

### **1.2.2 Batasan Permasalahan**

Dalam penelitian ini rancangan desain VLSI yang disisipkan watermark membatasi masalah serta pembahasan yang akan diteliti sebagai berikut :

1. Tidak membuat modul IC VLSI spesifik, namun menggunakan yang sudah ada dan menyisipkan dengan watermark.
2. Menyisipkan rangkaian dengan data watermark dan tidak membahas detail data dari pemilik cipta.

3. Watermarking yang dilakukan untuk satu chip IC dan tidak mewatermark masing-masing modul yang ter-integrasi dalam chip IC.

### **1.3 Tujuan**

Berikut merupakan tujuan pengamanan desain modul yang siap cetak sehingga aman terhadap pencurian hak cipta :

1. Merancang rangkaian pengamanan dalam sebuah chip design sebagai bukti kepemilikan desain (ownership) atau watermarking.
2. Desain chip yang telah diberi rangkaian watermark akan dianalisis perubahan performa dari desain sebelum dan sesudah watermarking serta kemungkinan watermark di-modifikasi oleh pihak lain atau reverse engineering untuk digunakan kembali oleh pengguna yang tidak sah.
3. Rangkaian ini akan ditanam di dalam chip yang pemanggilan informasi pemilik dari chip hanya diketahui oleh pemilik cipta.

### **1.4 Metodologi Penelitian**

Metode penelitian yang digunakan adalah perancangan dan prototyping dan percobaan untuk membuktikan hipotesis yang ada.

### **1.5 Sistematika Penulisan**

Sistematika penulisan laporan adalah sebagai berikut:

- Bab 1 PENDAHULUAN  
Berisi tentang uraian latar belakang, rumusan serta batasan masalah dan gambaran umum tentang penelitian sebelumnya yang sudah ada.
- Bab 2 TINJAUAN PUSTAKA  
Menjelaskan penjelasan singkat tentang LSI dan proses pembuatannya serta kemungkinan serangan dan cara mengatasinya. Cara mengatasi dari penelitian sebelumnya yang sudah ada dan cara yang diajukan oleh penulis.
- Bab 3 DESAIN DAN SIMULASI  
Penjelasan detail tentang desain yang di ajukan penulis untuk mengatasi pembajakan desain serta simulasi hasil dari perancangan desain yang diajukan.

- **Bab 4 PENGUJIAN DAN ANALISIS**

Hasil pengujian terhadap desain yang diajukan penulis serta analisis terhadap desain yang diajukan.

- **Bab 7 KESIMPULAN DAN SARAN**

Kesimpulan yang didapat dari hasil pengujian desain yang diajukan dan saran untuk pengembangan riset dimasa mendatang.

## **BAB 2**

### **TINJAUAN PUSTAKA**

Membuat desain sebuah perangkat IC membutuhkan proses yang panjang dan sumberdaya manusia yang banyak, serta tingkat ketelitian yang tinggi. Oleh karenanya di butuhkan biaya yang tidak kecil dan waktu yang cukup lama hanya untuk membuat sebuah desain IC. Dengan kerumitan yang tinggi serta waktu yang lama dalam setiap prosesnya kadang pihak yang tak bertanggung jawab melakukan kecurangna dengan mencuri desain untuk memotong waktu dan biaya yang di butuhkan untuk produksi. sehingga menjadi masalah dalam dunia permanufacturan ic. [3]

#### **2.1 Very Large Scale Integration**

*Very Large Scale Integration* atau disingkat LSI merupakan proses pembuatan sebuah IC dengan mengkombinasikan ribuan transistor ke dalam sebuah chip. VLSI ada sejak tahun 1970-an ketika semikonduktor kompleks dan teknologi komunikasi sedang berkembang. Mikroprosesor merupakan salah satu peraangkat VLSI. Sebelum adanya teknologi VLSI kebanyakan IC memiliki set fungsi yang terbatas yang dapat di jalankan. Sebuah perangkat chip elektronik dahulu hanya fokus pada sebuah fungsi seperti CPU, ROM, RAM dan rangkaian logika lainnya. Dengan adanya VLSI memungkinkan disainer IC untuk menambahkan berbagai fungsi kedalam sebuah chip IC. [2]

##### **2.1.1 Arus Pengembangan LSI**

Integrated Circuit (IC) merupakan teknologi sirkuit elektronika yang lebih maju. Sebuah rangkaian elektronika dibuat dari berbagai komponen elektronika yang berbeda beda seperti transistor, resistor, kapasitor dan dioda yang saling tersambung satu sama lain. [2]

Transistor merupakan komponen terpenting pada pengembangan teknologi komputer moderen. Sebelum ditemukannya transistor. Para Engineer harus menggunakan tabung vakum. Tabung vakum dapat bekerja sebagai saklar elektronik. Namun tabung vakum membutuhkan daya dan ruang yang besar, mahal, serta kemampuan eksekusi yang lambat membuat tabung vakum tergantikan oleh transistor.

Dengan ditemukannya transistor yang ukuran dan kebutuhan dayanya yang kecil namun tetap efektif, Para Engineer elektronik di tahun 1950an melihat banyak



sekali kemungkinan untuk implementasinya pada rangkaian elektronik yang lebih maju. Dengan semakin meningkatnya kompleksitas pada rangkaian elektronik munculah masalah-masalah baru.

Salah satunya adalah ukuran rangkaian. Sebuah rangkaian kompleks seperti komputer sangat bergantung pada kecepatan. Apabila jumlah komponen pada komputer terlalu banyak maka sambungan antar komponen juga semakin banyak dan semakin panjang, sehingga menyebabkan kecepatan transfer sinyal listrik menjadi berkurang yang menyebabkan proses pada komputer menjadi lambat.

Tahun 1958 masalah ini dapat dipecahkan oleh ide Jack S Kilby yang idenya adalah merangkai komponen elektronika dalam sebuah blok silikon (Monolithic Idea). Idenya tersebut tidak hanya mengurangi ukuran rangkaian namun juga mengurangi kebutuhan kabel sambungan antar rangkaian serta manufakturingnya dapat diautomasi. Akan tetapi idenya tersebut masih memiliki banyak masalah lain. Walaupun begitu, idenya tersebut mendapatkan penghargaan nobel di tahun 2000.

Setengah tahun setelah Kilby mencetuskan idenya tentang rangkaian Monolithic. Robert Noyce memiliki jawaban untuk beberapa permasalahan pada ide Kilby. Yaitu interkoneksi antar rangkaian. Yaitu menambahkan lapisan metal pada lapisan terakhir dan menghilangkan sebagian lapisannya sehingga sambungan antar komponen dapat terbentuk.

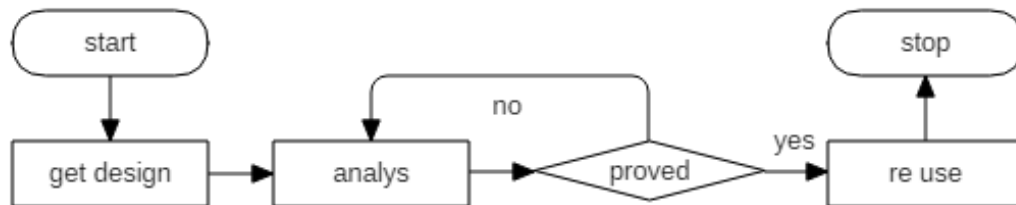


**Gambar 2.1:** Produksi Chip Moderen

Chip pada zaman sekarang berbasis pada photolithography. Pada teknik ini digunakan radiasi sinar Ultra Violet yang melewati sebuah mask menuju lembaran silikon yang di lapiisi filem photosensitive untuk membentuk suatu rangkaian.

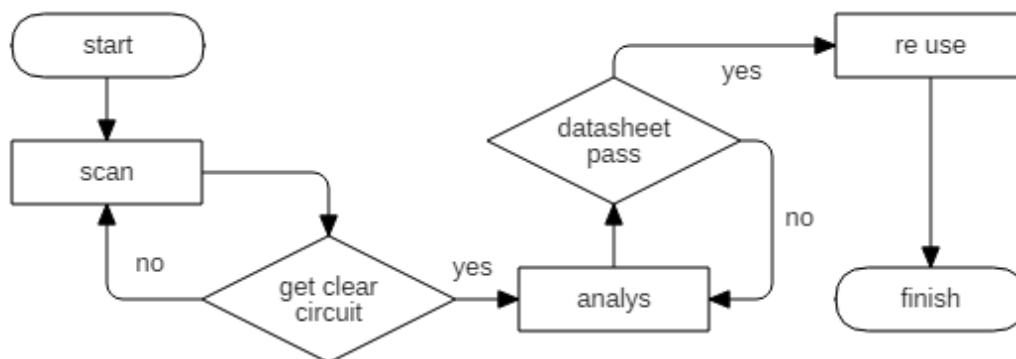
### 2.1.2 Kemungkinan Serangan Desain LSI

Dilihat dari proses developing, terdapat 2 cara untuk mendapatkan sebuah desain untuk di kloning. Pertama dengan mengambil langsung data mentah desain atau "blueprint" dan Reverse Engineering saat barang telah dipublikasi di pasaran.



**Gambar 2.2:** Clonning/Sumber Tidak Terpercaya

Dalam segi ini serangan dilakukan dengan cara mencuri langsung desain yang sudah siap di fabrikasi serta uji coba kebenaran. Bila pencuri mendapatkan desain yang telah di uji coba, maka pencuri tinggal langsung memperbanyak desain yang telah di curi.

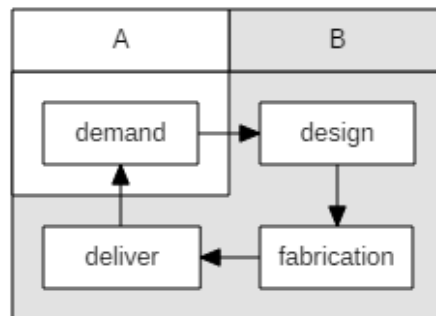


**Gambar 2.3:** RE (Reverse Engineering)

Untuk serangan jenis ini, pencuri sudah mendapatkan produk dari pasar yang telah teruji, pencuri tinggal melakukan scan rangkaian kemudian mengujinya dengan datasheet. Apabila hasil scan desain produk di dapati rangkaian yang konkrit/jelas dan rangkaian tersebut telah teruji sesuai datasheet. Maka pencuri tinggal melakukan fabrikasi.

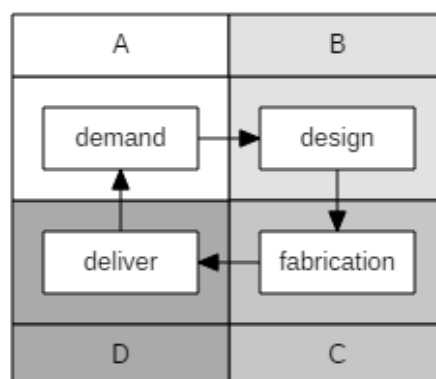
### 2.1.3 Mengatasi Serangan terhadap Desain LSI

Dengan meninjau kemungkinan dari tipe serangan, terdapat berbagai cara untuk mengatasi setiap serangan serangan tersebut. Dari reverse engineering hingga untrust source. untuk reverse engineering digunakan teknik anti reverse engineering dan untuk untrust source digunakan teknik identifier dan dengan enclosure agreement law.



**Gambar 2.4:** Model Bisnis Lama

Pada model gambar diatas, kegiatan desain, fabrikasi dan deliveri di lakukan oleh satu pihak yang sama. Proses pembuatan suatu perangkat IC dimonopoli oleh 1 perusahaan. Sehingga kemungkinan serangan hanya ada di antara pihak A dan Pihak B.



**Gambar 2.5:** Model Bisnis Baru

Namun seiring dengan perkembangnya jaman. Monopoli proses dari desain, fabrikasi hingga deliveri mulai sulit di terapkan. Karena dengan semakin berkembangnya jaman dan deman akan fitur desain semakin tinggi, otomatis biaya semakin

tinggi dan kompleksitas suatu desain semakin rumit serta waktu untuk menyelesaikan suatu desain semakin lama.

Oleh karena itu sekarang mulai diterapkan Fabless manufakturing atau joint venture untuk membuat suatu perangkat elektronika. Setidaknya pada proses bisnis ini terdapat 4 pihak. Pihak A dari keinginan pasar, pihak B yang melakukan perancangan desain, pihak C yang melakukan fabrikasi hasil rancangan pihak B dan Pihak D yang melakukan delivery hasil fabrikasi di pihak C ke A.

## **2.2 Teknik Proteksi**

Dari berbagai teknik yang telah digunakan, penulis melakukan penggabungan 2 teknik pengamanan dalam sebuah desain IC. Dalam penelitian ini dilakukan penggabungan 2 teknik agar cakupan wilayah keamanan sebuah IC semakin luas. Berikut teknik yang digabungkan dalam penelitian kali ini.

### **2.2.1 Digital Signal Processing Filter**

Digital Signal Processing (DSP) merupakan pengolahan sinyal digital, seperti digunakan pada komputer hingga untuk melakukan berbagai operasi proses sinyal. Sinyal yang diproses merupakan kumpulan bilangan sekuensial yang merepresentasikan sampel dari variabel sinyal kontinyu pada suatu domain seperti domain waktu, ruang atau frekuensi.

Pada pengolahan sinyal, sebuah filter adalah sebuah alat atau proses yang menghilangkan beberapa komponen atau fitur yang tidak diinginkan dari suatu sinyal. Filtering merupakan kelas proses sinyal,

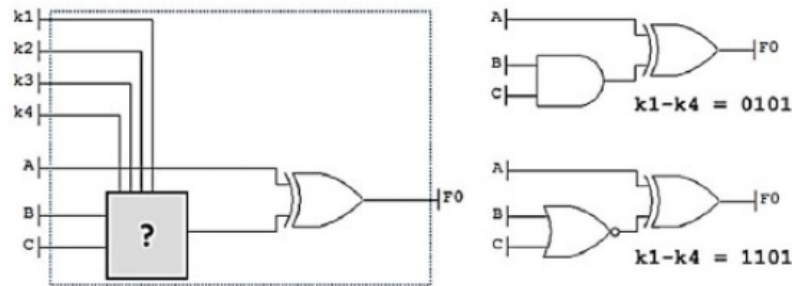
### **2.2.2 Polimorphisme Gate**

Polimorphisme Gate merupakan teknik pengecoh yang digunakan dalam perlindungan desain IC. Sebagai contoh sebuah rangkaian dengan output F dan input A, B dan C akan memiliki hasil yang berbeda jika parameter k yang diberikan berbeda. Misal bila parameter k diisi dengan kombinasi 0101 maka outputnya adalah

$$F = A \text{ XOR } (A \text{ AND } B)$$

Sedangkan bila parameter k diisi dengan kombinasi 1101 maka outputnya menjadi

$$F = A \text{ XOR } (A \text{ NOR } B)$$



Gambar 2.6: Polymorph gate

## 2.3 Peralatan dan Teknologi

Dalam penelitian kali ini dibutuhkan beberapa peralatan dan standard teknologi untuk mengembangkan teknik perlindungan intelektual properti. Sebagai penunjang dalam pembuatan perlindungan, penulis menggunakan tools dan teknologi yang umum digunakan dalam proses pengembangan desain LSI.

### 2.3.1 Verilog HDL

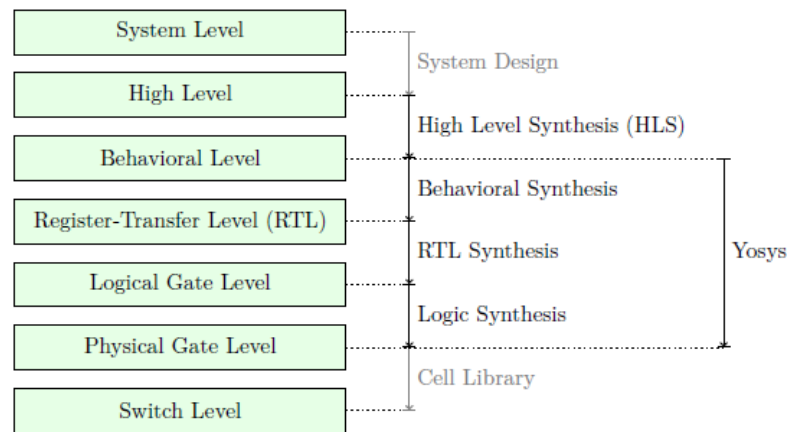
Verilog HDL merupakan bahasa pendeskripsi hardware yang digunakan untuk mendesain dan dokumentasi sistem elektronika. Verilog HDL memungkinkan perancang mendesain pada berbagai tingkatan abstraksi.

Verilog HDL berasal dari Automated Integrated Design System (yang kemudian berubah nama menjadi Gateway Design Automation) pada tahun 1985. Saat itu perusahaan tersebut dipegang oleh Dr. Prabhu Goel, pendiri PODEM test generation algorithm. Verilog HDL di desain oleh Phil Moorby, yang kemudian menjadi chief Designer untuk Verilog-XL dan perusahaan rekan pertama di Cadance Design System.

Awalnya Verilog dibuat sebagai bahasa simulasi. Kemudian setelah berkembang tidak hanya digunakan untuk simulasi namun juga untuk sintesis. (source [www.verilog.com](http://www.verilog.com))

### 2.3.2 Yosys Open SYnthesis Suite

Yosys adalah sebuah framework untuk sintesis Verilog RTL. Sekarang ini memiliki suport yang extensif pada Verilog-2005 dan mendukung berbagai set basik algoritma sintesis untuk berbagai domain aplikasi.



**Gambar 2.7:** Perbedaan Tinkatan Abstraksi dan Sintesis Yosys

### 2.3.3 Xilinx ISE Design Suit

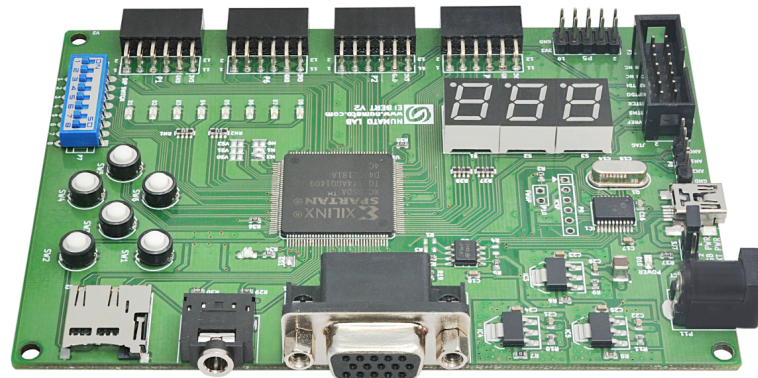
Xilinx ISE Design Suit merupakan Computer Aided Design (CAD) keluaran Xilinx yang digunakan untuk developing IC.



**Gambar 2.8:** Logo Xilinx ISE Design Suit

### 2.3.4 FPGA Elbert V2 Board

FPGA merupakan kepanjangan dari Field Programmable Gate Array adalah perangkat keras yang biasa digunakan dalam proses manufaktur IC. FPGA digunakan untuk mensimulasikan draft rancangan IC yang siap untuk di test yang apabila telah lolos test akan di lanjutkan ke tahap layout. FPGA hanya digunakan apabila rancangan membutuhkan input dari perangkat lain atau program kernel.



**Gambar 2.9:** FPGA Board - Elbert V2

Elbert V2 merupakan Board yang simple namun serbaguna untuk pembelajaran atau pengembangan. Board ini menggunakan Xilinx Spartan 3A FPGA. Pada Development Board ini memiliki fitur FPGA dari Xilinx XC3S50A dengan 144 pin dengan maksimum 108 user IO. Dilengkapi dengan antarmuka USB2 untuk kemudahan konfigurasi ke SPI flash.

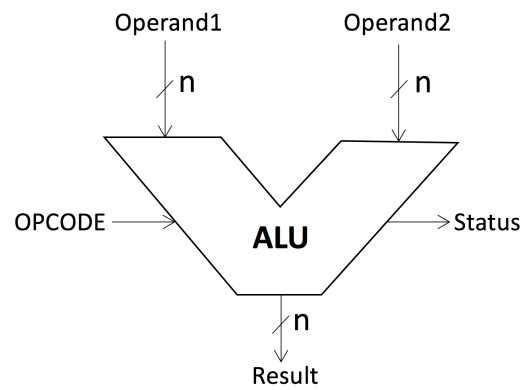
## 2.4 Target IP Core

Watermark adalah rangkaian yang tidak boleh berdiri sendiri pada implementasinya walaupun dalam pengembangannya bisa di lakukan mandiri. Dalam penelitian kali ini Module yang akan di watermark adalah modul ALU.

### 2.4.1 Aritmatic Logic Unit (ALU)

Aritmatik Logic Unit (ALU) adalah kombinasi rangkaian elektronik digital yang melakukan fungsi aritmatika dan operasi bitwise pada bilangan integer binari. Ini sangat kontras dengan Floating Point Unit (FPU), yang melakukan operasi bilangan floating point. Sebuah ALU pada dasarnya bagian dari berbagai macam blok

rangkaian komputasi, termasuk Central Prosesing Unit (CPU). Sebuah CPU, FPU, atau GPU mungkin memiliki banyak ALU di dalamnya.



**Gambar 2.10:** ALU



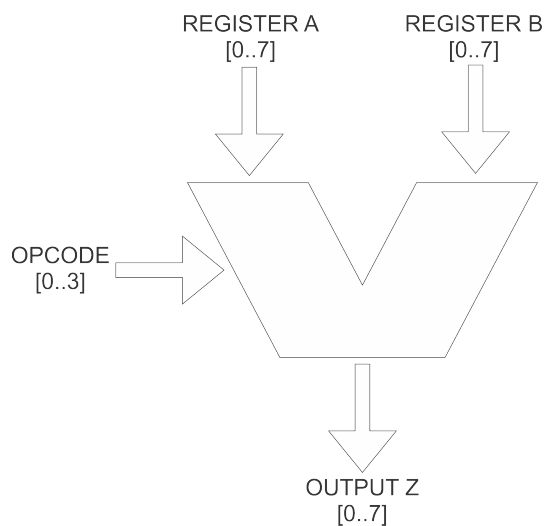
## BAB 3

### DESAIN DAN SIMULASI

Perancangan serta langkah-langkah di perlukan untuk menyelesaikan penelitian ini. Berikut ini akan di jelaskan gambaran serta tahapan dari perancangan system yang di teliti serta skenario simulasi dari hasil desain yang telah dirancang.

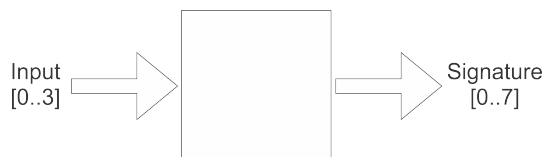
#### 3.1 Perancangan Desain

Desain utama adalah desain modul alu. Pada penelitian kali ini digunakan modul ALU dengan 2 register masing-masing 8 bit input, 4 bit opcode dan 8 bit output seperti ditunjukkan pada diagram dibawah ini.



**Gambar 3.1:** Desain ALU yang akan dilindungi

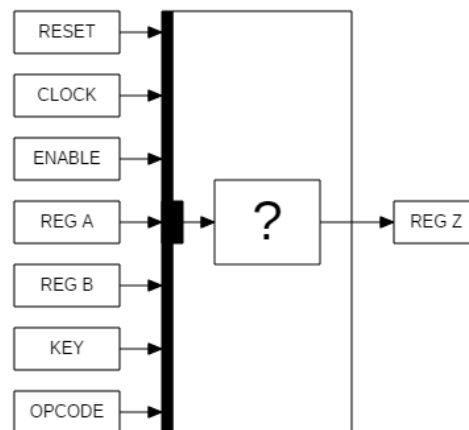
Selain menggunakan desain ALU penulis juga merancang desain perlindungan dengan input 4 bit dan output 8 bit seperti yang ditunjukkan diagram dibawah ini. modul inilah yang nantinya akan diselipkan pada modul ALU.



**Gambar 3.2:** Desain rangkaian pelindung

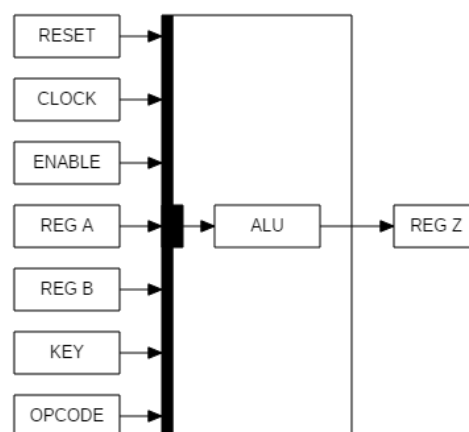
### 3.1.1 Skema Perlindungan

Skema perlindungan ini dilakukan teknik Pengolahan Sinyal Digital untuk ekstraksi data signature dan obfuscation dengan polymorph gate untuk menyembunyikan keberadaan signature. Dibawah berikut merupakan spesifikasi I/O pada modul yang akan dilindungi.



**Gambar 3.3:** Desain rangkaian Top modul yang terdapat rangkaian lain

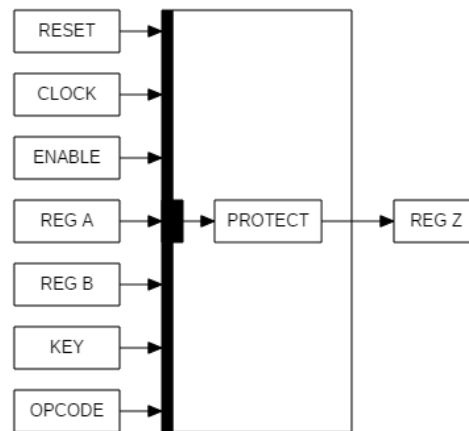
Untuk teknik obfuscation dengan polymorph dilakukan desain yang bertujuan menyelipkan suatu modul lain pada modul utama tanpa diketahui pengguna. Teknik ini seperti memberikan program Trojan kedalam suatu program utama. Namun sub-program ini bukan untuk merusak namun untuk melindungi. Seperti ilustrasi di atas menunjukkan suatu modul besar namun ada sesuatu lain di dalam modul tersebut.



**Gambar 3.4:** Desain rangkaian ALU pada top modul

Pertama penulis menggunakan rangkaian ALU sebagai modul besar (utama)

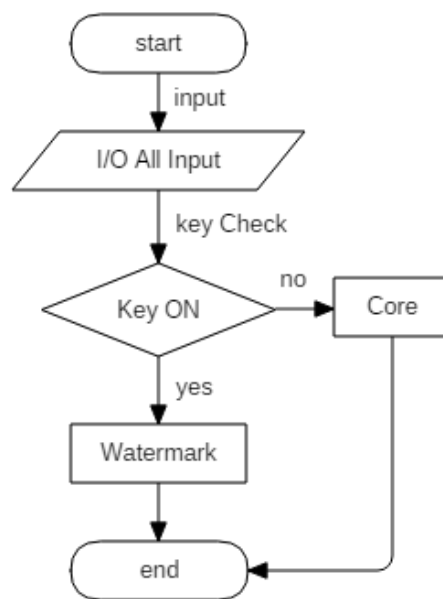
dengan spesifikasi I/O chip yang telah ditentukan sebelumnya. Lalu fungsional ALU dites dengan spesifikasi yang telah dibuat.



**Gambar 3.5:** Desain rangkaian pelindung pada top modul

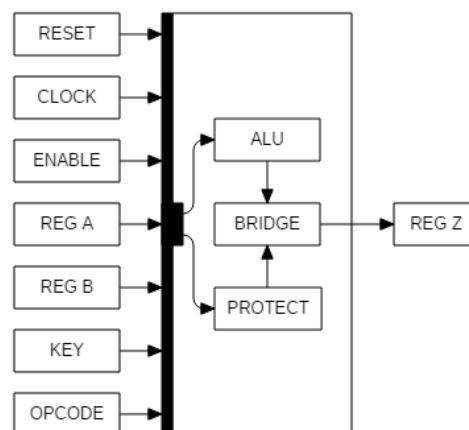
Setelah itu ganti rangkaian modul ALU dengan modul perlindungan sebagai modul utama kemudian dilakukan tes fungsional kembali untuk mengetahui apakah rangkaian perlindungan dapat bekerja diatas arsitektur serta spesifikasi modul utama.

Setelah didapat kedua modul dapat berjalan sesuai dengan semestinya maka langkah selanjutnya menggabungkan kedua modul tersebut bada modul utama. pada penggabungan kali ini digunakan teknik obfuscation polymorph untuk memilih antara modul mana yang harus berjalan pada modul utama.



**Gambar 3.6:** Algoritma aktivasi

Diagram diatas menunjukkan bagaimana cara mengaktifkan modul dengan key sebagai kontroler. Untuk hasil output diperlukan kontrol tambahan penjemabatan antara hasil output modul ALU dan modul perlindungan seperti pada ilustrasi di bawah agar tidak terjadi bentrokan antara kedua output,



**Gambar 3.7:** Desain rangkaian top modul yang telah diberi pelindung

dibawah ini merupakan contoh listing program pada ilustrasi di atas. Sehingga terdapat 1 Top modul dan 3 sub modul pada desain chip yang telah diberikan rangkaian pelindung.

```

module alu( RST, CLK, ENA, RGA, RGB, RGZ, KEY, OPT);
  input  RST, CLK, ENA;
  input  [3:0]OPT;
  input  [7:0]RGA,RGB;
  input  [1:0]KEY;
  output [7:0]RGZ;
  wire   [7:0]A,B,RGZ;
  alu_min aluj(RST, CLK, ENA, RGA, RGB, A, KEY, OPT);
  protection prot(RST, CLK, ENA, RGA, RGB, B, KEY);
  bridge jembatan(A, B, RGZ);
endmodule

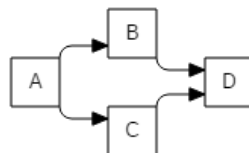
```

### 3.1.2 Spesifikasi

Spesifikasi rangkaian dapat dilihat di lampiran.

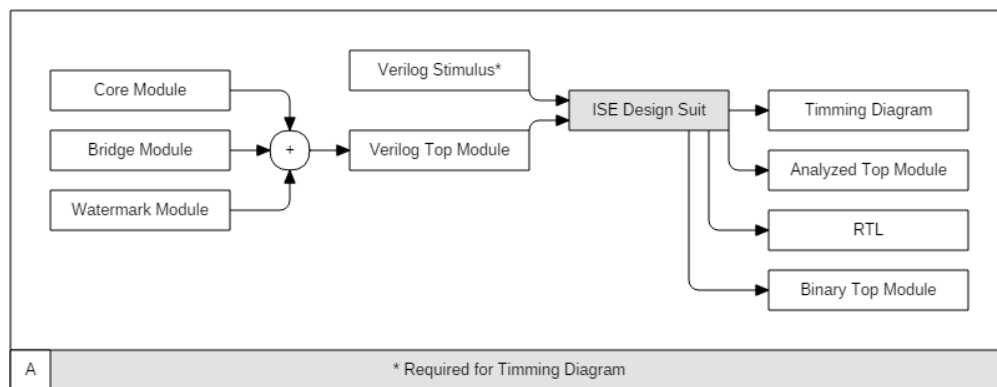
## 3.2 Alur Proses Pengembangan

Secara garis besar pada sisi desainer, terdapat 3 langkah untuk melakukan pengembangan alat dari programming hingga layout siap cetak. Penulis menggunakan cara ini dari hasil studi serta eksperimen saat proses pengembangan alat.



**Gambar 3.8:** Skema Perancangan Umum Proses Desain

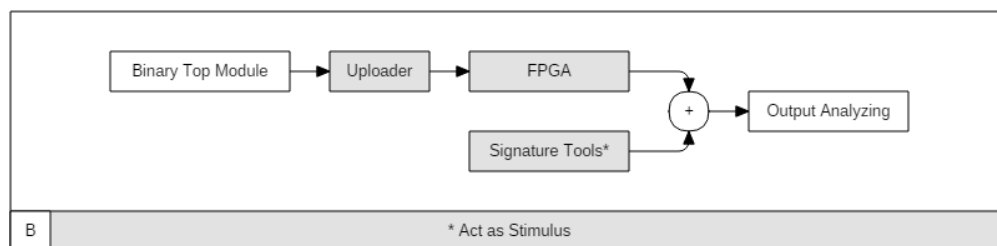
Pada langkah pertama dilakukan proses A, penulis melakukan perancangan desain dari IC yang akan di watermark kemudian di lakukan analisis. Apabila pertama telah selesai, penulis akan melakukan langkah kedua. Pada langkah kedua ini di lakukan kegiatan B yaitu proses ferifikasi dengan FPGA dan kegiatan C yaitu proses syntesys menjadi Raw Layout. Setelah kegiatan B dan C selesai maka kegiatan D yaitu proses finalisasi layout dapat dilakukan yang akhirnya hasil final layout dapat di serahkan ke pabrik untuk di fabrikasi.



**Gambar 3.9:** Skema kegiatan A

Secara umum pada kegiatan A, penulis membuat 3 module verilog untuk digabungkan. Core Module yaitu program rangkaian ALU, Watermark Module adalah program untuk watermarking dan Bridge Module untuk menghubungkan output antara Core Module dan Watermark Module. Setelah selesai dilakukan programming setiap module tersebut maka modul-modul tadi digabungkan menjadi Top Module. Top Module ini lah yang nantinya akan menjadi IC terwatermark.

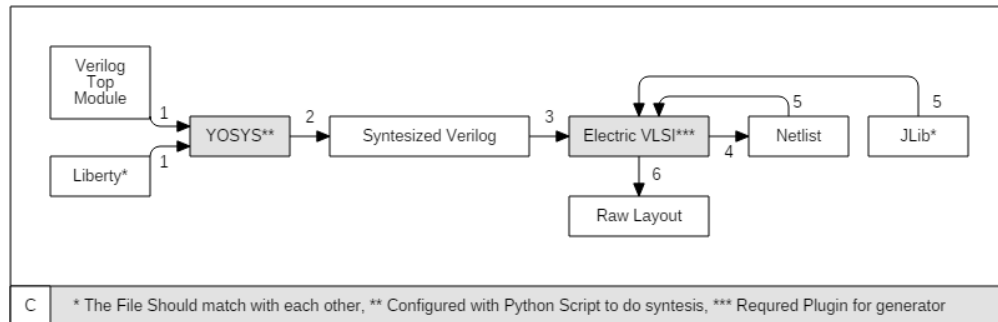
Pada top module ini harus diberikan program tambahan yaitu stimulus untuk dapat mensimulasikan skenario Input dan Output dari Top Module. Bila skenario stimulus telah dibuat, kemudian dilakukan simulasi dengan bantuan software ISE design suit untuk melihat hasil simulasi berupa Timing diagram. Pada Timing Diagram inilah dapat di lihat apakah skenario dari Input dan Output sesuai dengan keinginan. Setelah hasil analisis sesuai dengan yang diinginkan maka dilanjutkan dengan kegiatan selanjutnya yaitu kegiatan B dan C.



**Gambar 3.10:** Skema kegiatan B

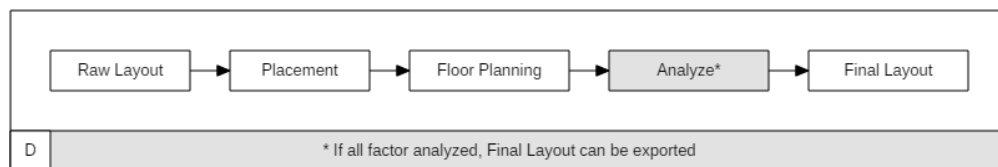
Pada kegiatan B ini dilakukan simulasi verifikasi pada board FPGA dengan alat verifikasi. kegiatan ini dilakukan untuk simulasi verifikasi signature pada IC yang telah diwatermark. IC yang telah diwatermark di tanam pada FPGA dan dengan menggunakan Signature Tools dilakukan verifikasi sehingga didapat data signature

dari IC yang telah di watermark.



**Gambar 3.11:** Skema kegiatan C

Untuk kegiatan C dilakukan developing layout dari Top Module yang telah diferifikasi dengan timming diagram. Defeloping menggunakan software Electric VLSI. Dengan mensynthesis Verilog Top Module dan Liberty file menggunakan YOSYS, maka akan di dapat file verilog tersynthesis. Kemudian File tersintesis tersebut di Load di Electric VLSI untuk di rubah ke NetList. Setelah berhasil di rubah menjadi NetList maka file NetList tersebut di kompilasi bersama file JLib pada electric VLSI untuk di jadikan Raw Layout.



**Gambar 3.12:** Skema kegiatan D

Pada tahap ini di lakukan kegiatan C yaitu memproses Raw Layout menjadi Final Layout yang siap di cetak. Tahapan nya adalah melakukan placement untuk setiap modulnya lalu di lakukan analisis kemudian dilakukan Floor planning. Setelah itu dilakukan analisis kembali hingga didapat hasil yang terbaik. Apabila telah didapat hasil yang terbaik maka File siap untuk difabrikasi.

### 3.3 Simulasi

Simulasi dilakukan pada 2 environment, yaitu simulasi software (Timing diagram) serta simulasi hardware (FPGA).

## **BAB 4**

### **PENGUJIAN DAN ANALISIS**

#### **4.1 Pengujian**

Setelah desain selesai di rancang, kemudian dilakukan pengujian untuk mengetahui apakah desain yang telah dirancang dapat bekerja dengan baik serta mengetahui bagaimana performansi alat yang di rancang.

##### **4.1.1 Skenario Pengujian**

Untuk pengujian yang dilakukan adalah pengujian fungsional algoritma (behavioral) serta pengujian performansi waktu operasi dan daya yang dibutuhkan alat yang telah dirancang.

Pada pengujian fungsional dilakukan pengujian menggunakan simulasi diagram waktu. Pada diagram waktu dapat dilihat proses input output satu demi satu dalam satuan waktu tertentu. Pengujian satu persatu dilakukan untuk mengetahui secara detail fungsional IC yang telah dirancang.

Apabila uji fungsional tidak ditemukan anomali, maka dilanjutkan pengujian selanjutnya yaitu pengujian daya serta performansi. Hal ini untuk mengetahui apakah terjadi perubahan yang signifikan pada perangkat yang diberi perlindungan dengan yang tidak diberi perlindungan

##### **4.1.2 Hasil Pengujian**

Berikut ini adalah hasil pengujian fungsional dari perangkat yang telah dilindungi. pengujian yang pertama adalah pengujian fungsional modul inti yaitu modul ALU.



**Tabel 4.1:** Data Mentah

| REG |   |   |   |   |   |   |   |   |  |   |   |   |   |   |   |   |  |
|-----|---|---|---|---|---|---|---|---|--|---|---|---|---|---|---|---|--|
| CLK | A |   |   |   |   |   |   |   |  | Z |   |   |   |   |   |   |  |
| 0   | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 1   | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |  | 0 | 1 | 0 | 0 | 0 | 0 | 0 |  |
| 2   | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |  | 1 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 3   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |  | 1 | 1 | 0 | 0 | 1 | 0 | 0 |  |
| 4   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |  | 0 | 0 | 0 | 0 | 1 | 0 | 1 |  |
| 5   | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |  | 0 | 1 | 0 | 0 | 1 | 0 | 1 |  |
| 6   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |  | 1 | 0 | 0 | 0 | 1 | 0 | 1 |  |
| 7   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 1 | 1 | 0 | 1 | 0 | 0 | 0 |  |
| 8   | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |  | 0 | 0 | 0 | 1 | 0 | 0 | 0 |  |
| 9   | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |  | 0 | 1 | 0 | 1 | 0 | 0 | 0 |  |
| 10  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |  | 1 | 0 | 0 | 1 | 0 | 0 | 0 |  |
| 11  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |  | 1 | 1 | 0 | 1 | 1 | 0 | 1 |  |
| 12  | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |  | 0 | 0 | 0 | 1 | 1 | 0 | 1 |  |
| 13  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |  | 0 | 1 | 0 | 1 | 1 | 0 | 1 |  |
| 14  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |  | 1 | 0 | 0 | 1 | 1 | 0 | 1 |  |
| 15  | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |  | 1 | 1 | 1 | 0 | 0 | 1 | 0 |  |
| 16  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |  | 0 | 0 | 1 | 0 | 0 | 1 | 0 |  |
| 17  | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |  | 0 | 1 | 1 | 0 | 0 | 1 | 0 |  |
| 18  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |  | 1 | 0 | 1 | 0 | 0 | 1 | 0 |  |

Data mentah (raw) di atas didapat dari hasil timing diagram (lampiran). Untuk mengetahui hasil kebenaran dilakukan truth prove pada data raw.

**Tabel 4.2:** Analisis Data Mentah

| IO |     | bit |    |    |    |   |   |   |   | STEP |      |       |
|----|-----|-----|----|----|----|---|---|---|---|------|------|-------|
| in | out | 7   | 6  | 5  | 4  | 3 | 2 | 1 | 0 | HCUT | MCUT | CHECK |
| 7  | 0   | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 5  | 64  | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 6  | 128 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 2  | 202 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 10   | 2    | OK    |
| 3  | 10  | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 4  | 74  | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 2  | 138 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 0  | 208 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 16   | 0    | OK    |
| 7  | 16  | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 4  | 80  | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 2  | 144 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 1  | 217 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 25   | 1    | OK    |
| 5  | 25  | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 3  | 89  | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 3  | 153 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 4  | 228 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 36   | 4    | OK    |
| 2  | 36  | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 7  | 100 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |
| 3  | 164 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0    | 0    | -     |

Dari hasil pengujian di atas didapat bahwa fungsional dari masing masing modul yang telah digabung tidak saling bentrok satu sama lain dan dapat bekerja dengan semestinya.

Dengan hasil pengujian fungsional yang tidak terdapat anomali maka dapat dilanjutkan analisis daya dari modul sebelum diberi perlindungan dengan modul yang telah diberi perlindungan.

## 4.2 Analisis

Pertama analisis performansi perangkat sebelum dilindungi, kemudian lindungi alat dengan rangkaian pelindung dan analisis performansinya dan bandingkan hasil analisis sebelum dengan hasil analisis sesudah dilindungi. berikut rekap data hasil analisis sebelum dan sesudah diberi rangkaian pelindung. Berikut merupakan soft-estimasi kecepatan clock pada FPGA arsitektur XILINX.

**Tabel 4.3:** FPGA Speed Analysis

| Unprotected                      | Minimum  | Maximum           |
|----------------------------------|----------|-------------------|
| Period                           | 2.692ns  | 371.471MHz (freq) |
| Input arrival time before clock  | 10.075ns | -                 |
| Output required time after clock | -        | 5.558ns           |
| Protected                        | Minimum  | Maximum           |
| Period                           | 4.023ns  | 248.571MHz (freq) |
| Input arrival time before clock  | 8.667ns  | -                 |
| Output required time after clock | -        | 6.962ns           |

Dari hasil analisis terjadi penurunan kecepatan maksimum proses pada FPGA dari 371.471 Mhz menjadi 248.571 Mhz atau sekitar 33%. Pada hasil soft-simulasi ini mengindikasikan bakal terjadi penurunan speed pada modul yang sedang dikembangkan.

**Tabel 4.4:** On-Chip Power Summary

| Unprotected  | Power (mW) |
|--------------|------------|
| Clock        | 1.12       |
| Static Power | 10.42      |
| Total        | 11.54      |
| Protected    | Power (mW) |
| Clock        | 1.37       |
| Static Power | 10.42      |
| Total        | 11.79      |

**Tabel 4.5:** Power Supply Currents

| Unprotected   |                |                    |                      |                        |
|---------------|----------------|--------------------|----------------------|------------------------|
| Supply Source | Supply Voltage | Total Current (mA) | Dynamic Current (mA) | Quiescent Current (mA) |
| Vccint        | 1.20           | 2.95               | 0.94                 | 2.01                   |
| Vccaux        | 2.50           | 3.00               | 0.00                 | 3.00                   |
| Vcco25        | 2.50           | 0.20               | 0.00                 | 0.20                   |
| Protected     |                |                    |                      |                        |
| Supply Source | Supply Voltage | Total Current (mA) | Dynamic Current (mA) | Quiescent Current (mA) |
| Vccint        | 1.20           | 3.16               | 1.14                 | 2.02                   |
| Vccaux        | 2.50           | 3.00               | 0.00                 | 3.00                   |
| Vcco25        | 2.50           | 2.00               | 0.00                 | 0.20                   |

Dari hasil diatas terlihat kebutuhan arus pada FPGA meningkat dari 2.95 menjadi 3.16 atau sekitar 6.64% pada suplay Vccint di 1.2 Volt.

**Tabel 4.6:** Gate yang digunakan setelah kompilasi ke netlist

| Gate Synthesys | Gates |
|----------------|-------|
| Unprotected    | 5234  |
| Protected      | 5324  |

Kompilasi menggunakan library standard dari mosis didapatkan banyaknya gate yang digunakan sebelum rangkaian dilindungi adalah 5234 dan meningkat menjadi 5324 setelah rangkaian diberi pelindung. Peningkatan jumlah gate yang digunakan meningkat sekitar 1.69%. Artinya bila dilakukan generate dari netlist ke layout peningkatan luas layoutnya tidak terlalu signifikan.

Selanjutnya desain disimulasikan dengan FPGA dan kontroler

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Meninjau hasil analisis yang dilakukan dari pengujian yang telah dilakukan, di dapat kesimpulan sebagai berikut.

1. Menggunakan lebih dari satu metode perlindungan masih memungkinkan untuk melindungi Properti intelektual perangkat keras.
2. Terjadi penurunan kecepatan proses serta peningkatan kebutuhan daya pada hasil analisis simulasi di fpga.
3. zero-overhead I/O bisa dilakukan dengan tambahan obfuscation-mux.

#### **5.2 Saran**

Agar Teknologi pengamanan Intelektual Properti lebih maju serta memperbaiki permasalahan yang masih ada pada penelitian ini, berikut beberapa saran dari untuk pengembangan dan penelitian selanjutnya:

1. Untuk meningkatkan kecepatan akses dan mengurangi konsumsi daya dari hasil analisis level behavioral pada fpga, dibutuhkan analisis lebih lanjut pada syntesis level gate (netlist) dan level phisical (layout).
2. Saat ini teknologi serta teknik perlindungan properti intelektual perangkat keras masih terbilang baru, bidang ke amanan pada IC masih minim resource serta proses manufakturing IC sendiri begitu kompleks dan luas dan spesifikasi desain setiap produk sangat rahasia. Dibutuhkan pendalaman ilmu khusus pada Level fabrikasi seperti RTL Level, Gate Level dan Layout Level.

## DAFTAR REFERENSI

- [1] Jeff Clark. (n.d). *Introduction to LaTeX*. 26 Januari 2010. <http://frodo.elon.edu/tutorial/tutorial/node3.html>.
- [2] "The History of the Integrated Circuit". Nobelprize.org. Nobel Media AB 2014. Web. 25 Aug 2017. [http://www.nobelprize.org/educational/physics/integrated\\_circuit/history/](http://www.nobelprize.org/educational/physics/integrated_circuit/history/).
- [3] Leonid Azriel, Student Member, Ran Ginosar, Senior Member, and Shay Gueron. Using Scan Side Channel to Detect IP Theft. pages 1–13, 2017.
- [4] Abhishek Basak, Swarup Bhunia, Senior Member, Thomas Tkacik, Sandip Ray, and Senior Member. Security Assurance for System-on-Chip Designs With Untrusted IPs. 12(7):1515–1528, 2017.
- [5] Mohammad-mahdi Bidmeshki, Xiaolong Guo, Raj Gautam Dutta, Yier Jin, and Yiorgos Makris. Tracking in Proof-Carrying Hardware IP Part II :. 12(10):2430–2443, 2017.
- [6] Xi Chen, Gang Qui, Aijiao Cui, and Carson Dunbar. Scan Chain based IP Fingerprint and Identification. 2017.
- [7] Xiaoming Chen, Qiaoyi Liu, Yu Wang, Qiang Xu, and Huazhong Yang. Low-Overhead Implementation of Logic Encryption Using Gate Replacement Techniques. 2017.
- [8] Jeffrey T Delloso. The Impact of the Innovation and Technology Support Offices ( ITSOs ) on Innovation , Intellectual Property ( IP ) Protection and Entrepreneurship in Philippine Engineering Education. (April):762–770, 2017.
- [9] Xiaolong Guo, Student Member, Raj Gautam Dutta, Student Member, and Yier Jin. Eliminating the Hardware-Software Boundary : A Proof-Carrying Approach for Trust Evaluation on Computer Systems. 12(2):405–417, 2017.
- [10] Yier Jin, Xiaolong Guo, Raj Gautam Dutta, Mohammad-mahdi Bidmeshki, and Yiorgos Makris. Tracking in Proof-Carrying Hardware IP Part I :. 12(10):2416–2429, 2017.

- [11] Jian Lin. Analysis of the Key Factors of Intellectual Property Management at Art Institutions. pages 206–208, 2017.
- [12] Hardware Matters. Antipiracy-Aware IP Chip Set Design for CE Devices: A Robust Watermarking Approach. (april):118–124, 2017.
- [13] Hardware Matters. Hardware Security of CE Devices. (January), 2017.
- [14] By Saraju P Mohanty and Rochester Chapters. Information Security and IP Protection Are Increasingly Critical in the Current Global Context. (June):3–5, 2017.
- [15] By Saraju P Mohanty and Rochester Chapters. Information Security and IP Protection Are Increasingly Critical in the Current Global Context. (June):3–5, 2017.
- [16] Xuan Thuy Ngo, Jean-luc Danger, Sylvain Guilley, Tarik Graba, Yves Mathieu, Zakaria Najm, and Shivam Bhasin. Cryptographically Secure Shield for Security IPs Protection Threats on Integrated Circuits. 66(2):354–360, 2017.
- [17] Xuan Thuy Ngo, Jean-luc Danger, Sylvain Guilley, Tarik Graba, Yves Mathieu, Zakaria Najm, and Shivam Bhasin. Cryptographically Secure Shield for Security IPs Protection Threats on Integrated Circuits. 66(2):354–360, 2017.
- [18] Protection Of, Trade Secrets, Under The, T S Directive, and Protection During. The European Union Trade-Secrets Directive: To-Dos for Companies? (april):2016–2017, 2017.
- [19] A Sengupta and D Roy. Protecting IP core during architectural synthesis using HLT-based obfuscation. 53(13):1–2, 2017.
- [20] A Sengupta and D Roy. Protecting IP core during architectural synthesis using HLT-based obfuscation. 53(13):1–2, 2017.
- [21] Anirban Sengupta, Member Ieee, Dipanjan Roy, Student Member Ieee, and Saraju P Mohanty. Triple - Phase Watermarking for Reusable IP Core Protection during Architecture Synthesis. 0070(c), 2017.
- [22] Wei-tek Tsai, Libo Feng, and Hui Zhang. Intellectual-Property Blockchain-based Protection Model for Microfilms. pages 174–178, 2017.
- [23] Nandeeshha Veeranna and Benjamin Carrion Schafer. Efficient Behavioral Intellectual Properties Source Code Obfuscation for High-Level Synthesis. 2017.

- [24] Marc Wehlack and Konrad Spang. Motivations for and Barriers to Offshoring Development Projects to China A Case Study of the Automotive Industry. pages 169–173, 2017.
- [25] Muhammad Yasin, Student Member, Ozgur Sinanoglu, and Senior Member. Testing the Trustworthiness of IC Testing : An Oracle-Less Attack on IC Camouflaging. 12(11):2668–2682, 2017.
- [26] Dongrong Zhang, Miao Tony He, Xiaoxiao Wang, and Mark Tehranipoor. Dynamically Obfuscated Scan for Protecting IPs Against Scan-Based Attacks Throughout Supply Chain. 2017.
- [27] Jiliang Zhang and Lele Liu. Publicly Verifiable Watermarking for Intellectual Property Protection in FPGA Design. 25(4):1520–1527, 2017.
- [28] Jiliang Zhang and Lele Liu. Publicly Verifiable Watermarking for Intellectual Property Protection in FPGA Design. 25(4):1520–1527, 2017.



# LAMPIRAN

## **LAMPIRAN 1 COMERCIAL DATASHEET**

## Pin Map



## Legend

|                                      |        |
|--------------------------------------|--------|
| <span style="color: red;">■</span>   | VCC    |
| <span style="color: black;">■</span> | GROUND |
| <span style="color: blue;">■</span>  | INPUT  |
| <span style="color: gray;">■</span>  | OUTPUT |

## Foot Configuration

| No.           | Pin      | Signal Type | Data Type |
|---------------|----------|-------------|-----------|
| 7-14          | PA 0 - 7 | Digital     | Input A   |
| 27-34         | PB 0 - 7 | Digital     | Input B   |
| 17            | PC 0     | Digital     | Enable    |
| 18            | PC 1     | Digital     | Clock     |
| 19            | PC 2     | Digital     | Reset     |
| 20            | PC 3     | Digital     | Reff      |
| 21-24         | PC 4-7   | Digital     | OPCODE    |
| 37-40 and 1-4 | PD 0 - 7 | Digital     | Output    |

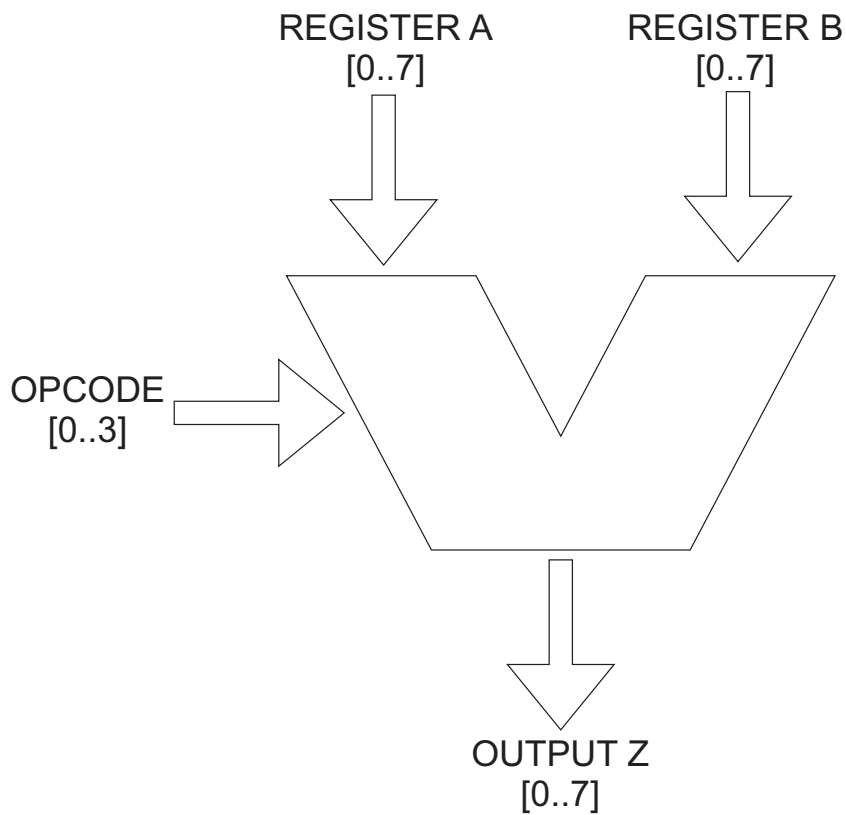
Copyright © 2017 by Hanjara Cahya Adhyatma RnEST Laboratory. Telkom University. All rights reserved. This documentation or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission of the publisher except for the use of brief quotations in a document review.

Production  
**SAMiCALNA Fabric**  
 Type  
**Arithmetic Logic Unit**  
 Family  
**C4B3R4W1T**  
 Package  
**TQFP/MLF**  
 Version  
**1.0**

## Power Information

| Foot | Pin | Data Size | Voltage  | I    | Freq  |
|------|-----|-----------|----------|------|-------|
| 7    | PA0 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 8    | PA1 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 9    | PA2 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 10   | PA3 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 11   | PA4 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 12   | PA5 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 13   | PA6 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 14   | PA7 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 27   | PB0 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 28   | PB1 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 29   | PB2 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 30   | PB3 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 31   | PB4 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 32   | PB5 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 33   | PB6 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 34   | PB7 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 17   | PC0 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 18   | PC1 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 19   | PC2 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 20   | PC3 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 21   | PC4 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 22   | PC5 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 23   | PC6 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 24   | PC7 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 37   | PD0 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 38   | PD1 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 39   | PD2 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 40   | PD3 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 1    | PD4 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 2    | PD5 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 3    | PD6 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 4    | PD7 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |

## ALU Block Diagram



## ALU Function

| opcode | Operation Z | opcode | Operation Z | opcode | Operation Z |
|--------|-------------|--------|-------------|--------|-------------|
| 01     | RGZ = 0     | 11     | RGB + RGZ   | 21     | RGB - RGA   |
| 02     | RGA + RGB   | 12     | RGB - RGZ   | 22     | RGB ^ RGA   |
| 03     | RGA - RGB   | 13     | RGB ^ RGZ   | 23     | RGB & RGA   |
| 04     | RGA ^ RGB   | 14     | RGB & RGZ   | 24     | RGB   RGA   |
| 05     | RGA & RGB   | 15     | RGB   RGZ   | 25     | RGB && RGA  |
| 06     | RGA   RGB   | 16     | RGB && RGZ  | 26     | RGB    RGA  |
| 07     | RGA && RGB  | 17     | RGB    RGZ  | 27     | RGB + 1     |
| 08     | RGA    RGB  | 18     | RGZ + 1     | 28     | RGB - 1     |
| 09     | RGA + 1     | 19     | RGZ - 1     | 29     | RGB << 1    |
| 0A     | RGA - 1     | 1A     | RGZ << 1    | 2A     | RGB >> 1    |
| 0B     | RGA << 1    | 1B     | RGZ >> 1    | 2B     | ! RGB       |
| 0C     | RGA >> 1    | 1C     | ! RGZ       | 2C     | ~ RGB       |
| 0D     | ! RGA       | 1D     | ~ RGZ       | 2D     | RGB + RGB   |
| 0E     | ~ RGA       | 1E     | RGZ + RGZ   | 2E     | RGB - RGB   |
| 0F     | RGA + RGA   | 1F     | RGZ - RGZ   | 2F     | RGA + RGZ   |
| 10     | RGA - RGA   | 20     | RGB + RGA   | 30     | RGA - RGZ   |

## **LAMPIRAN 2 INTERNAL/DEVELOPER DATASHEET**

## Pin Map



## Legend

|   |  |
|---|--|
| <span style="color: red;">■</span> VCC      | <span style="color: yellow;">■</span> K0       |
| <span style="color: black;">■</span> GROUND | <span style="color: green;">■</span> K1        |
| <span style="color: blue;">■</span> INPUT   | <span style="color: olive;">■</span> SIGNATURE |
| <span style="color: grey;">■</span> OUTPUT  |  |

## Foot Configuration

| No.           | Pin      | Signal Type | Data Type |
|---------------|----------|-------------|-----------|
| 7-14          | PA 0 - 7 | Digital     | Input A   |
| 27-34         | PB 0 - 7 | Digital     | Input B   |
| 17            | PC 0     | Digital     | Enable    |
| 18            | PC 1     | Digital     | Clock     |
| 19            | PC 2     | Digital     | Reset     |
| 20            | PC 3     | Digital     | Reff      |
| 21-24         | PC 4-7   | Digital     | OPCODE    |
| 37-40 and 1-4 | PD 0 - 7 | Digital     | Output    |

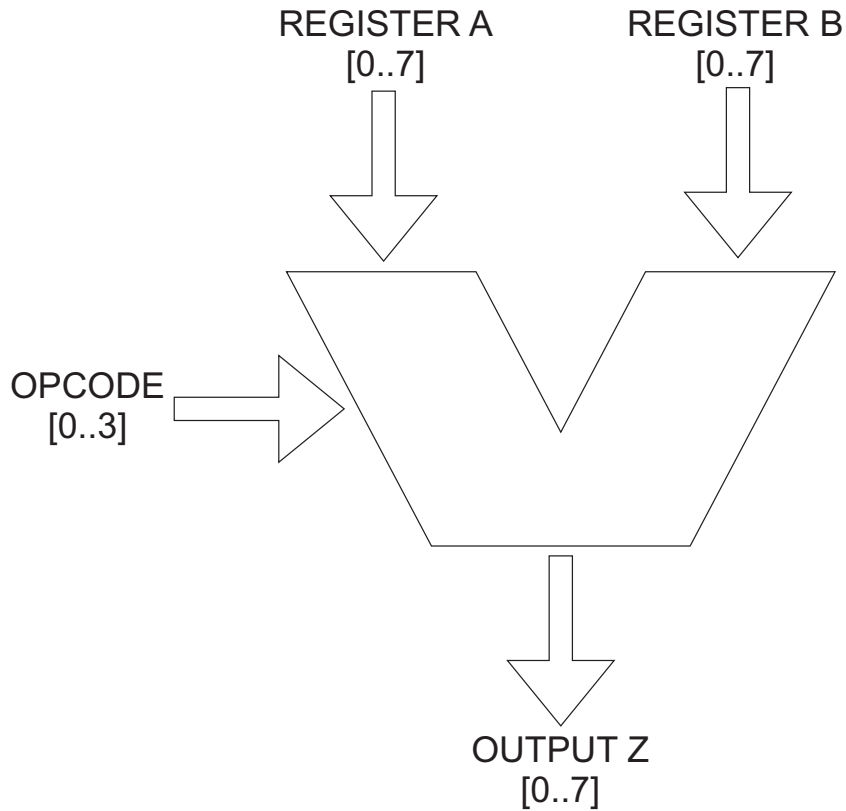
Copyright © 2017 by Hanjara Cahya Adhyatma RnEST Laboratory. Telkom University. All rights reserved. This documentation or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission of the publisher except for the use of brief quotations in a document review.

Production  
**SAMiCALNA Fabric**  
 Type  
**Arithmetic Logic Unit**  
 Family  
**C4B3R4W1T**  
 Package  
**TQFP/MLF**  
 Version  
**1.0**

## Power Information

| Foot | Pin | Data Size | Voltage  | I    | Freq  |
|------|-----|-----------|----------|------|-------|
| 7    | PA0 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 8    | PA1 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 9    | PA2 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 10   | PA3 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 11   | PA4 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 12   | PA5 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 13   | PA6 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 14   | PA7 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 27   | PB0 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 28   | PB1 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 29   | PB2 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 30   | PB3 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 31   | PB4 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 32   | PB5 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 33   | PB6 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 34   | PB7 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 17   | PC0 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 18   | PC1 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 19   | PC2 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 20   | PC3 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 21   | PC4 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 22   | PC5 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 23   | PC6 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 24   | PC7 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 37   | PD0 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 38   | PD1 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 39   | PD2 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 40   | PD3 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 1    | PD4 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 2    | PD5 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 3    | PD6 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |
| 4    | PD7 | 1 bit     | 3.3 Volt | .5 A | 1 MHz |

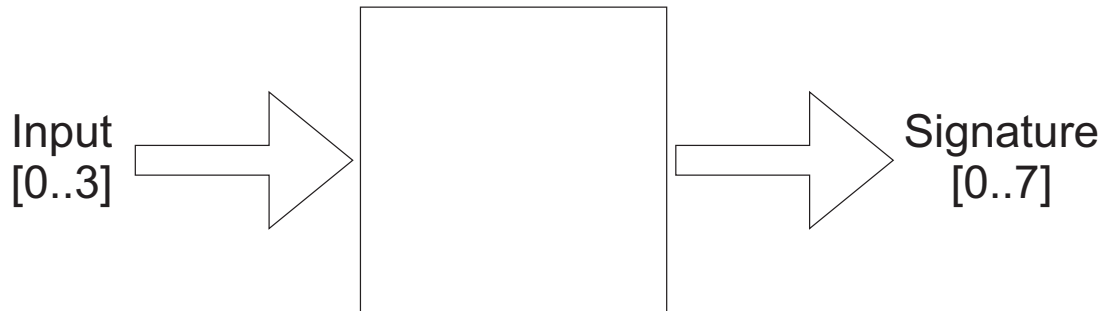
## ALU Block Diagram



## ALU Function

| opcode | Operation Z | opcode | Operation Z | opcode | Operation Z |
|--------|-------------|--------|-------------|--------|-------------|
| 01     | RGZ = 0     | 11     | RGB + RGZ   | 21     | RGB - RGA   |
| 02     | RGA + RGB   | 12     | RGB - RGZ   | 22     | RGB ^ RGA   |
| 03     | RGA - RGB   | 13     | RGB ^ RGZ   | 23     | RGB & RGA   |
| 04     | RGA ^ RGB   | 14     | RGB & RGZ   | 24     | RGB   RGA   |
| 05     | RGA & RGB   | 15     | RGB   RGZ   | 25     | RGB && RGA  |
| 06     | RGA   RGB   | 16     | RGB && RGZ  | 26     | RGB    RGA  |
| 07     | RGA && RGB  | 17     | RGB    RGZ  | 27     | RGB + 1     |
| 08     | RGA    RGB  | 18     | RGZ + 1     | 28     | RGB - 1     |
| 09     | RGA + 1     | 19     | RGZ - 1     | 29     | RGB << 1    |
| 0A     | RGA - 1     | 1A     | RGZ << 1    | 2A     | RGB >> 1    |
| 0B     | RGA << 1    | 1B     | RGZ >> 1    | 2B     | ! RGB       |
| 0C     | RGA >> 1    | 1C     | ! RGZ       | 2C     | ~ RGB       |
| 0D     | ! RGA       | 1D     | ~ RGZ       | 2D     | RGB + RGB   |
| 0E     | ~ RGA       | 1E     | RGZ + RGZ   | 2E     | RGB - RGB   |
| 0F     | RGA + RGA   | 1F     | RGZ - RGZ   | 2F     | RGA + RGZ   |
| 10     | RGA - RGA   | 20     | RGB + RGA   | 30     | RGA - RGZ   |
|        |             |        |             | 00     | NO VALUE    |

## Signature Block Diagram

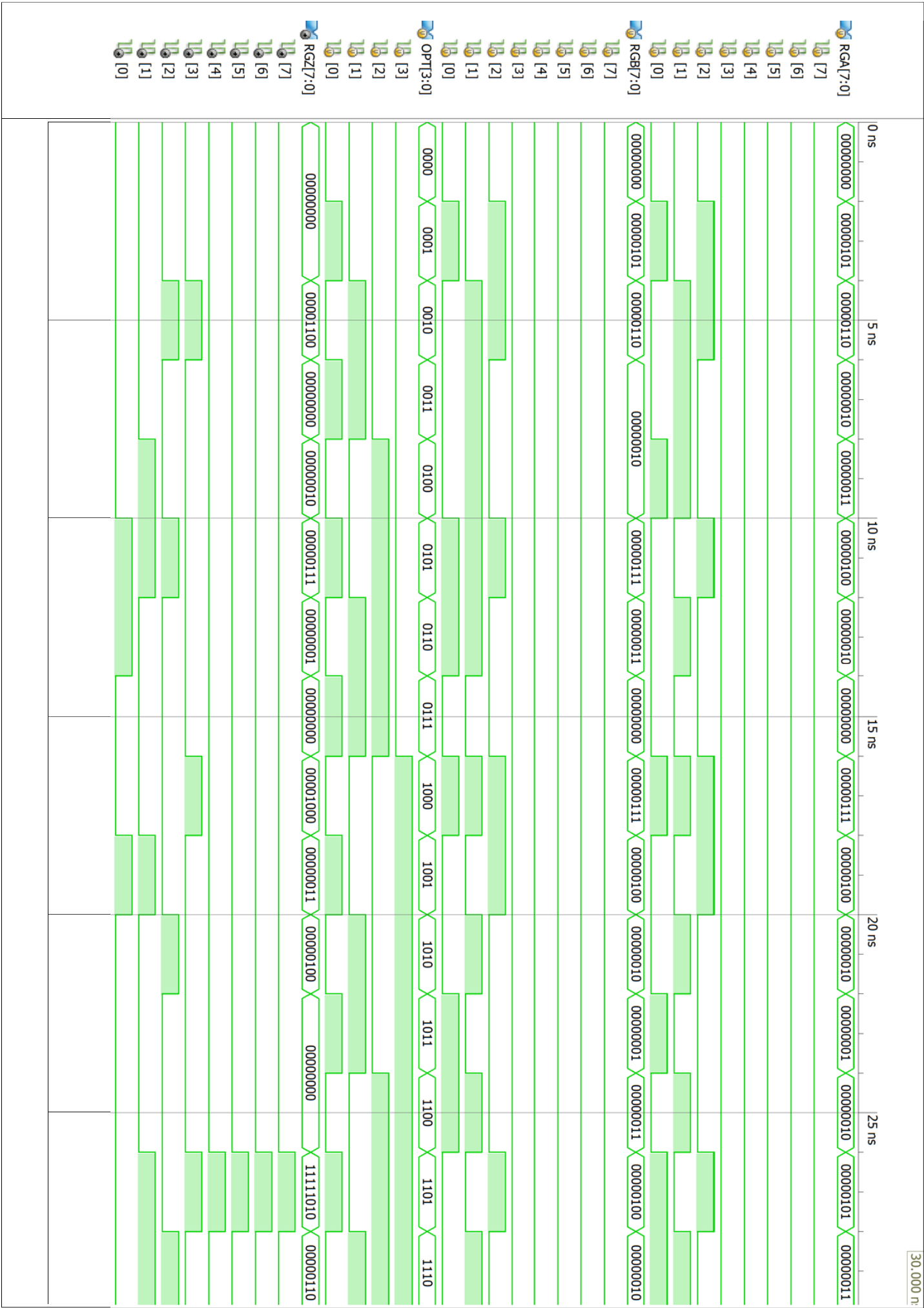


## Signature Activation

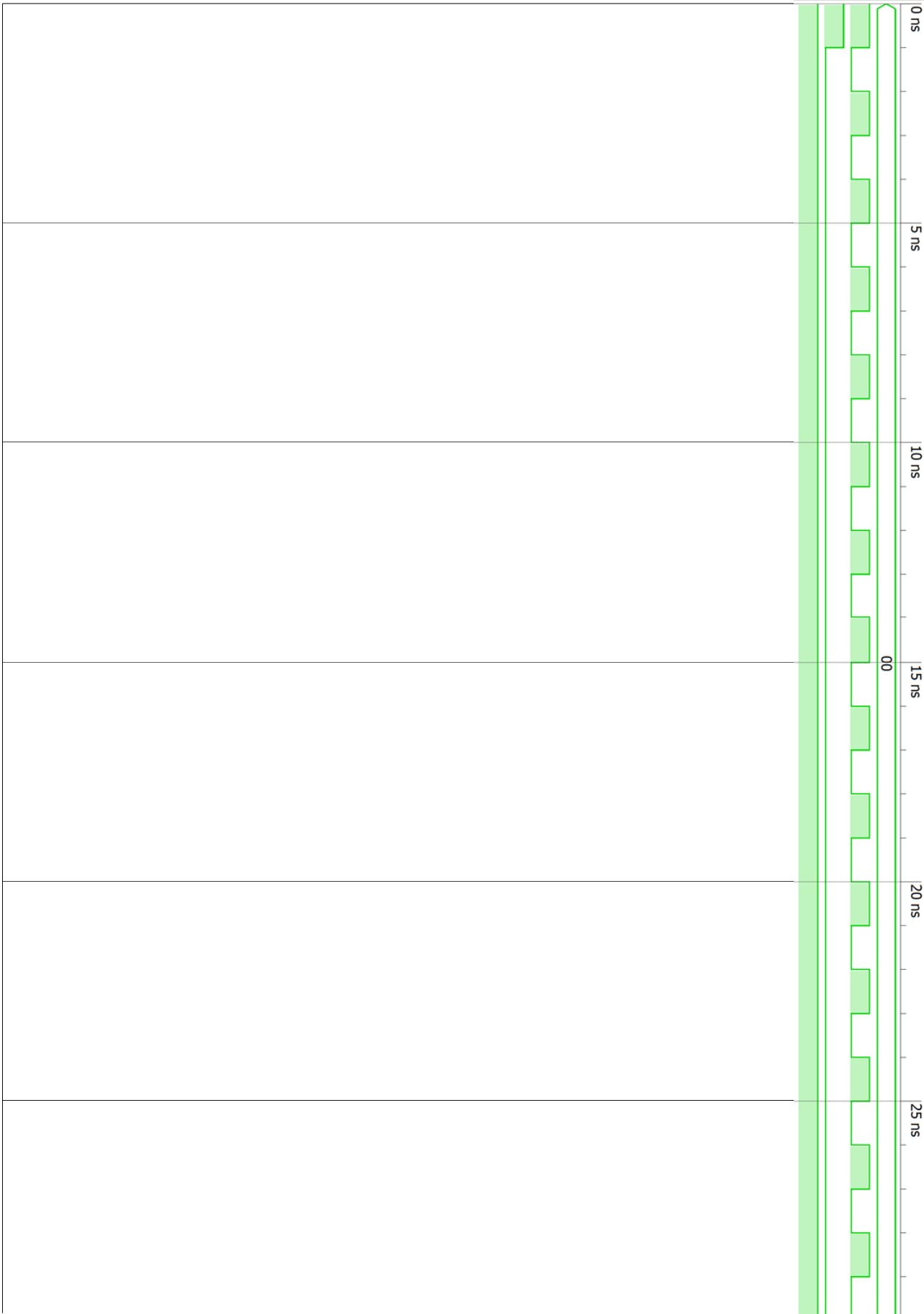
| opcode | Z           | key | Signature |       |       |
|--------|-------------|-----|-----------|-------|-------|
| 00     | Signature   | 01  | 2 bit     | 3 bit | 3 bit |
|        |             |     | Time      | Time  | Data  |
| XX     | Operation Z | 00  | 8 bit     |       |       |
|        |             |     | Data      |       |       |



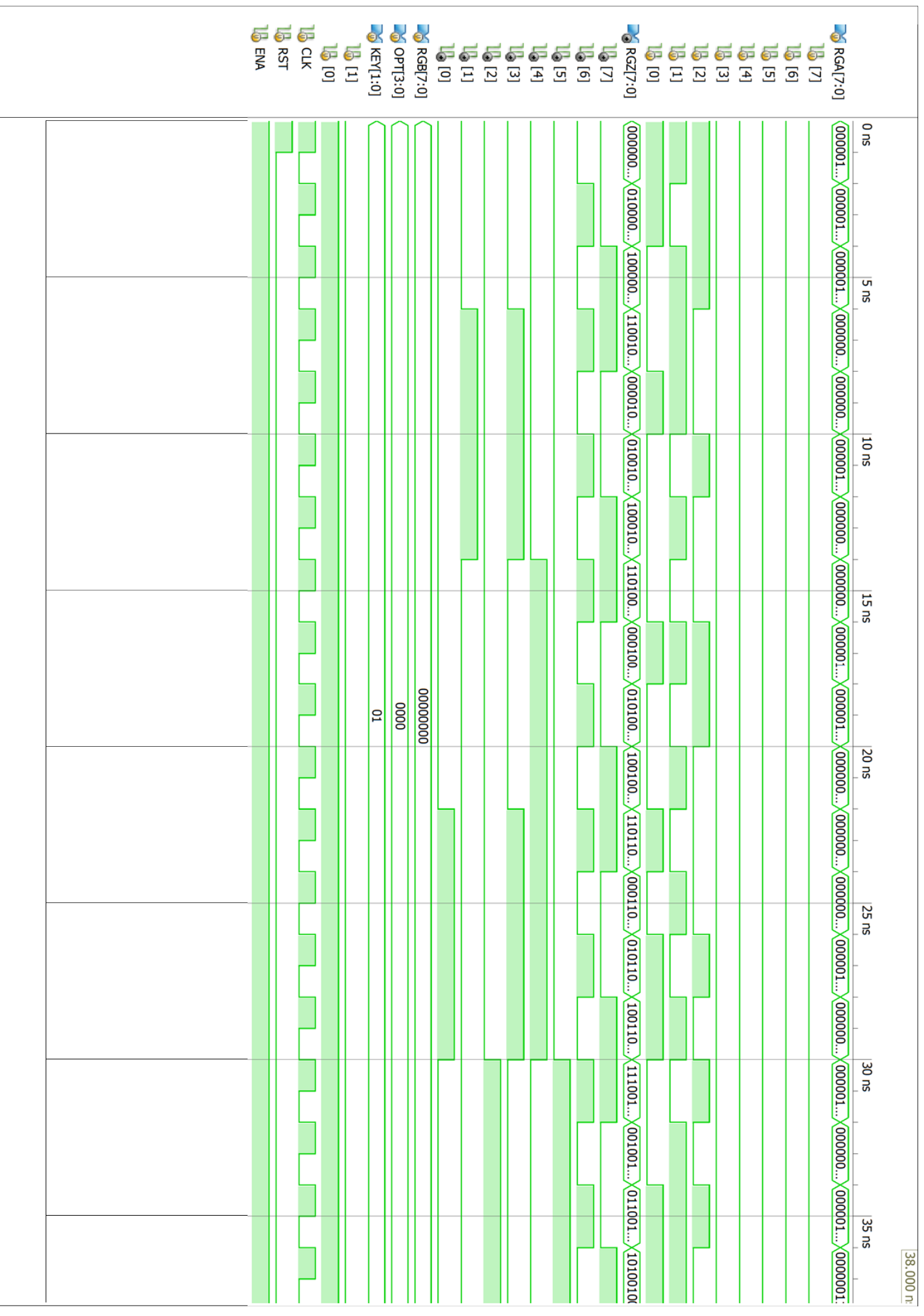
### **LAMPIRAN 3 ALU TEST BENCH - ALU ACTIVE**



KEY[1:0]  
CLK  
RST  
ENA

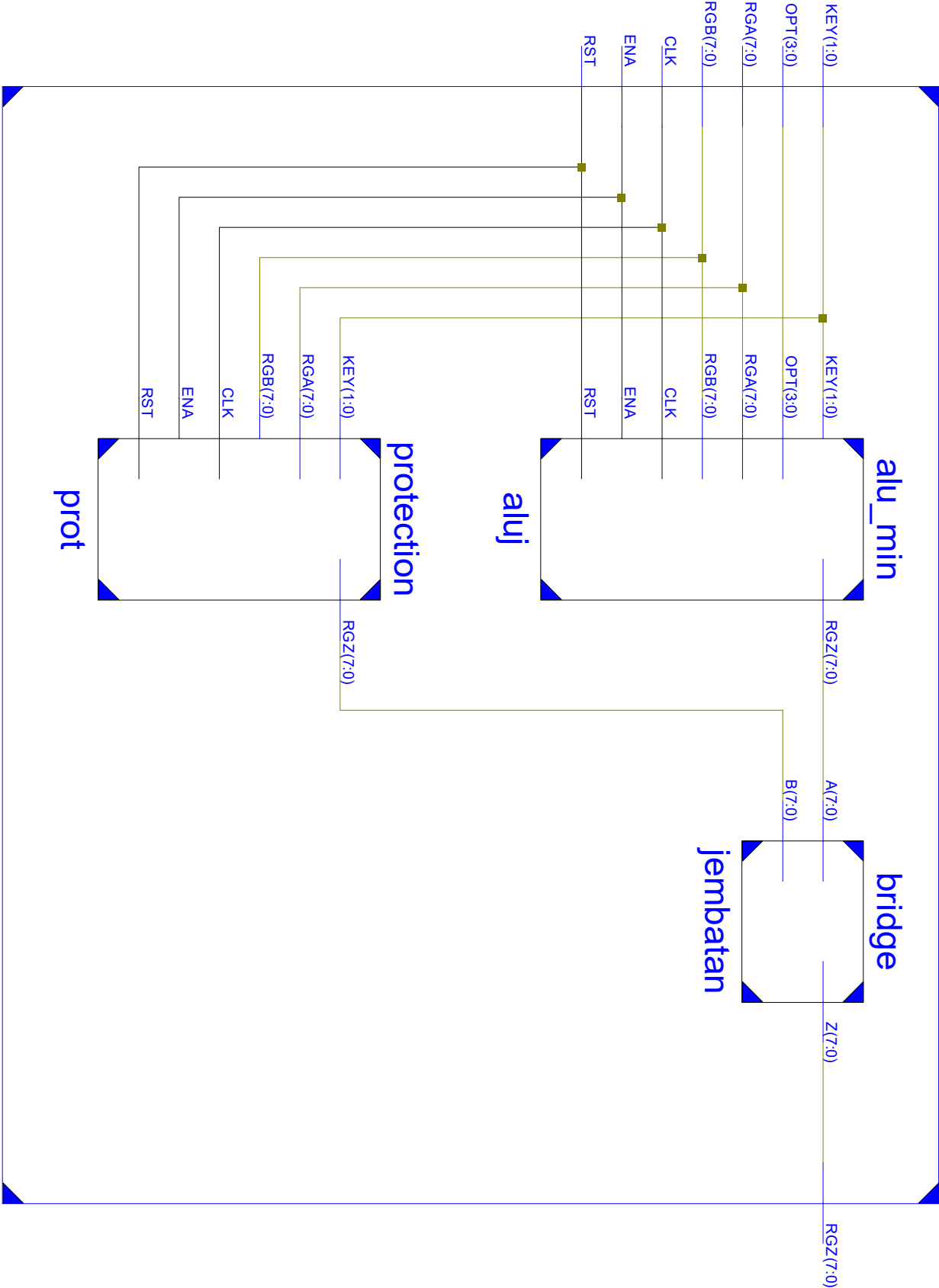


## **LAMPIRAN 4 ALU TEST BENCH - PROTECTION ACTIVE**



## **LAMPIRAN 5 RTL TOP MODULE**

alu:1



alu

## **LAMPIRAN 6 RTL PROTECTION**



