

# Implementasi Sistem Hermitian Generalized LED Index Modulation (H-GLIM-OFDM) pada Board FPGA Xilinx Arty Artix-7

Yetti Yuniati, S.T., M.T.<sup>1</sup> dan Vini Meiriska<sup>2</sup>

<sup>1,2</sup>Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung

Jl. Prof. Sumantri Brojonegoro No.1 Bandar Lampung 35145

<sup>1</sup>yetti.yuniati@eng.unila.ac.id

<sup>2</sup>vini.meiriska1501@students.unila.ac.id

**Intisari** — Komunikasi nirkabel menjadi jenis komunikasi yang digunakan secara luas, spektrum radio yang umumnya digunakan dalam komunikasi nirkabel menjadi tidak cukup untuk memenuhi tuntutan yang tinggi. Visible Light Communication (VLC) menjadi solusi untuk mengatasi kapasitas bandwidth yang kurang memadai ini. Orthogonal Frequency Division Multiplexing (OFDM) menjadi teknik yang dikembangkan untuk sistem komunikasi cahaya tampak karena pada teknik OFDM frekuensi yang digunakan saling orthogonal dan memungkinkan overlap antar frekuensi tanpa menimbulkan interferensi satu sama lain sehingga menghasilkan kecepatan transfer data yang tinggi. Jurnal ini membahas tentang Implementasi Sistem Hermitian Generalized LED Index Modulation (H-GLIM-OFDM). Skema dari H-GLIM-OFDM ini dirancang dalam System Generator pada bahasa pemrograman Matlab. Desain di implementasikan dalam FPGA dan diterapkan dengan spesifikasi Arty Board Xilinx Artix-7. Field Programmable Gate Array (FPGA) merupakan perangkat semikonduktor yang dapat diprogram secara fleksibel dan dapat melakukan kinerja yang tinggi untuk implementasi VLC. Hasil yang didapat pada simulasi ini yaitu menggunakan pemanfaatan sumber daya desain 5% BRAM, 11% dari DSP, 8% dari LUT, 18% dari IO, dan 3% BUFG.

**Kata kunci** — H-GLIM-OFDM, FPGA Arty Artix-7, Visible Light Communication, Vivado, System Generator

**Abstract** — Wireless communication is becoming the type of communication that is widely used, the radio spectrum which is generally used in wireless communication is not enough to meet high demands. Visible Light Communication (VLC) is a solution to overcome this inadequate bandwidth capacity. Orthogonal Frequency Division Multiplexing (OFDM) is a technique developed for visible light communication systems because OFDM techniques use orthogonal frequencies and allow overlap between frequencies without causing interference with each other so as to produce high data transfer speeds. This journal discusses the Implementation of the Hermitian Generalized LED Index Modulation (H-GLIM-OFDM) System. The scheme of the H-GLIM-OFDM was designed in the System Generator in the Matlab programming language. The design is implemented in the FPGA and implemented with the Xilinx Artix-7 Arty Board specifications. Field Programmable Gate Array (FPGA) is a semiconductor device that can be programmed flexibly and can perform high performance for VLC implementations. The results obtained in this simulation are using the utilization of 5% BRAM design resources, 11% from DSP, 8% from LUT, 18% from IO, and 3% BUFG

**Keywords** — H-GLIM-OFDM, FPGA Arty Artix-7, Visible Light Communication, Vivado, System Generator

## I. PENDAHULUAN

Komunikasi wireless dewasa ini dijadikan sebagai jenis komunikasi yang digunakan secara luas sehingga tuntutan komunikasi wireless semakin meningkat. Spektrum radio memiliki panjang frekuensi hanya sebesar 3Hz-3THz yang menjadikannya tidak cukup

untuk memenuhi permintaan para pengguna yang sangat tinggi. Salah satu komunikasi nirkabel yang dipakai yaitu Wireless Fidelity (Wi-Fi). Wi-Fi adalah satu dari beberapa jenis komunikasi wireless yang memakai gelombang radio yang menggunakan gelombang elektromagnetik sebagai pengirim data pada spektrum Radio Frequency (RF).

Karena pemakaian RF telah banyak digunakan, maka akan terjadi polusi gelombang elektromagnetik RF di udara yang sangat berbahaya khususnya bagi manusia. Salah satu solusi yang digunakan untuk mengurangi polusi pada udara adalah menggunakan spektrum cahaya yang umumnya terlihat dan digunakan sebagai pencahayaan serta penerangan ruangan seperti Visible Light Communication (VLC) yang dapat mengirimkan data pada kecepatan tinggi. Komunikasi cahaya tampak adalah solusi untuk mengatasi kapasitas bandwidth yang tidak memadai dalam spektrum radio dan salah satu contoh sebagai pendukung komunikasi generasi selanjutnya dan khususnya dalam smart city.

Meskipun memiliki kelebihan kapasitas tinggi dan tingkat data yang tinggi, penggunaan komunikasi cahaya tampak dapat meningkatkan efisiensi energi karena Light Emitting Diode (LED) yang digunakan sebagai pemancar. VLC dapat mengefisienkan transmisi dan penerimaan dari suatu data. Data ini dapat dikirimkan melalui cahaya lampu LED dan diterima oleh photodiode. Proses transmitting dan receiving data ini memerlukan teknik modulasi sinyal seperti Quadrature Amplitude Modulation (QAM) dan teknik multiplexing menggunakan Orthogonal Frequency Division Multiplexing (OFDM). VLC ini disebut sebagai Light Fidelity (Li-Fi).

## II. TEORIDASAR

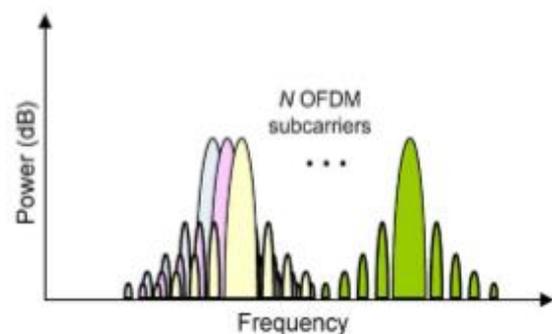
### A. VLC (Visible Light Communication)

VLC menjadi pilihan alternatif untuk teknologi wireless generasi mendatang dikarenakan VLC menawarkan cost yang rendah serta bandwidth yang tidak perlu diatur dan juga dukungan infrastruktur di mana-mana. Teknologi VLC diharapkan dapat digunakan dalam berbagai aplikasi. VLC menggunakan LED (Light Emitting Diode) yang memiliki tugas untuk memancarkan cahaya dan juga transmisi data. Dengan teknologi terdepan ini, data termasuk video, audio, dan internet traffic dapat ditransmisikan pada kecepatan tinggi menggunakan lampu LED.

Li-Fi merupakan teknologi yang diusulkan oleh Prof Harald Hass, seorang ilmuwan Jerman, bersama dengan timnya termasuk Dr. Gordon Povey, Dr. Mostafa Afgani di Universitas Edinburgh. Li-Fi menyediakan transmisi data melalui penerangan dengan mengirimkan data melalui bola lampu LED. Teknologi VLC yang berkaitan dengan transfer data melalui pencahayaan dengan mengambil fiber to fiber dengan mengirim data melalui bola lampu LED. LED adalah dioda sambungan p-n yang memancarkan cahaya ketika diaktifkan dan membantu dalam teknik Li-Fi. Aliran Li-Fi bersifat bidirectional dan menyediakan kecepatan tinggi dengan komunikasi nirkabel murah melalui cahaya. Li-Fi menggunakan cahaya bukan gelombang radio seperti di Wi-Fi untuk transfer data.

### B. OFDM (Orthogonal Frequency Division Multiplexing)

Pengembangan OFDM dimulai dengan menerapkan FFT (Fast Fourier Transform), menambahkan awalan siklik, dan lain sebagainya. OFDM dapat mengirimkan data dengan kecepatan tinggi tanpa menyebabkan ISI (Inter Symbol Interference) dan menghemat lebih banyak bandwidth, sehingga laju data dan bandwidth yang lebih tinggi dapat dicapai. Hal ini disebabkan karena penggunaan frekuensi subcarrier yang orthogonal satu sama lain meskipun subcarrier saling tumpang tindih.



Gbr.1 Bentuk Gelombang OFDM [3]

### 1) O-OFDM (Optical Orthogonal Frequency Division Multiplexing)

OFDM biasanya digunakan dalam komunikasi RF (Radio Frequency). Optical OFDM adalah salah satu pengembangan di OFDM, dimana peneliti mulai

mengembangkan OFDM untuk komunikasi optik nirkabel pada tahun 2001 dan untuk komunikasi optic fiber pada tahun 2005.

## 2) Unipolar Optical OFDM

Seperti yang dijelaskan pada bagian sebelumnya, OFDM optik memiliki kendala yang tak terelakkan bahwa sinyal harus nyata dan bernilai positif [3]. Beberapa metode RF OFDM telah diadopsi ke dalam optical OFDM seperti DC Biased Optical-OFDM (DCO-OFDM) dan Asymmetrical Clip Optical-OFDM (ACO-OFDM). Baik DCO-OFDM dan ACO-OFDM sama-sama digunakan untuk menghilangkan nilai negatif. DCO-OFDM memiliki pendekatan yang sederhana, pertama dibatasi oleh simetri Hermitian untuk memastikan output nyata, kemudian menambahkan daya bias DC ke dalam sinyal dan sinyal negatif yang tersisa akan terpotong menjadi nol. DCO-OFDM mencapai efisiensi spektral yang baik, tetapi karena ditambahkan bias DC yang menyebabkannya mendapatkan PAPR yang tinggi. ACO-OFDM di sisi lain menggunakan properti subcarrier, di mana hanya subcarrier ganjil yang digunakan [4].

Sinyal dibatasi ke dalam simetri Hermitian seperti DCO-OFDM, karena hanya subcarrier ganjil yang dimodulasi, IFFT menghasilkan proporsi anti simetris. ACO-OFDM mencapai PAPR yang lebih baik karena tidak ada bias DC seperti DCO-OFDM. Sinyal dibatasi ke dalam simetri Hermitian, kemudian subcarrier ganjil diproses oleh ACO-OFDM dan DCO-OFDM. Pendekatan lain dengan menggunakan FLIP-OFDM/U-OFDM yang mencapai efisiensi daya yang lebih baik dibandingkan dengan DCO-OFDM dan ACO-OFDM. Pada FLIP-OFDM, sinyal masih dibatasi oleh simetri Hermitian. Dibandingkan dengan ACO-OFDM dan DCO-OFDM yang hanya menggunakan subcarrier ganjil dan memotong sinyal negatif, pendekatan FLIP-OFDM sepenuhnya menggunakan semua subcarrier dan nilai negatif digunakan dengan membalikkan sinyal. Kemudian dua pasang sinyal dikelompokkan menjadi sinyal positif dan sinyal negatif. Pertama ia akan mengambil sampel sinyal positif sementara sinyal negatif diabaikan dan sampel sinyal negatif diambil sementara sinyal positif

diabaikan. Kemudian sinyal sampel digabungkan menjadi satu frame [5].

## 3) Bipolar Optical OFDM

Istilah bipolar di bagian ini tidak berarti sinyal yang dikirimkan berada dalam sinyal bipolar real, seperti yang dijelaskan dibagian sebelumnya bahwa domain optik hanya dapat mengirimkan sinyal dalam bentuk nilai positif. Istilah bipolar yang benar untuk bagian ini adalah dimana beberapa metode OFDM mengeksplorasi properti bipolar dari sinyal kompleks sementara beberapa dari mereka memotong sinyal negatif [5]. Pada GLIM-OFDM (Generalized LED Index Modulation-OFDM), sinyal kompleks diubah menjadi sinyal bipolar real, namun sinyal-sinyal ini kemudian ditransmisikan melalui saluran MIMO dalam modulasi spasial multipleks dan modulasi indeks LED yang berbeda dari skema SISO (single-output single-output). GLIM-OFDM memanfaatkan sinyal kompleks dan indeks LED, jadi tidak perlu menggunakan simetri Hermitian dan bias DC. Sinyal kompleks dipisahkan menjadi sinyal nyata dan kompleks, sedangkan setiap sinyal komponen masih mengandung nilai bipolar, separator polar digunakan untuk memisahkan indeks polar dan bagian negatif dibalik menjadi sinyal positif. [5].

### C. Hermitian Symmetry

Hermitian Symmetry digunakan untuk menghilangkan nilai imajiner dengan mengkonjugasikannya. Karena karakteristik optik tidak dapat mentransmisikan nilai imajiner, maka simetri Hermitian diperlukan untuk memastikan keluaran dari IFFT bernilai nyata atau real. Simetri Hermitian pada OFDM digunakan untuk menentukan kondisi dimana sinyal masukan bernilai nyata yang dapat diperoleh secara langsung [8].

### D. QAM (Quadrature Amplitude Modulation)

QAM digunakan untuk modulasi adaptif dikarenakan efisiensinya dalam penggunaan power dan bandwidth. Untuk lebih memahami sistem QAM, sistem simulasi berbasis Simulink dirancang [6]. Teknik modulasi data QAM dikirim dengan memvariasikan baik amplitudo dan fasa dari

sinyal carrier. Umumnya dua gelombang carrier diambil dari sinyal yang saling orthogonal satu sama lain. Nilai 4 bit dikelompokkan dengan mengambil dua bit dari masing-masing pembawa untuk membentuk simbol. Jumlah simbol yang mungkin adalah  $2^4 = 16$ , salah satu dari 16 kemungkinan sinyal ini ditransmisikan pada setiap periode symbol [7]. Bentuk umum dari 16 sinyal QAM dapat didefinisikan pada Persamaan 1 dan 2 sebagai berikut:

$$S_i(t) = \sqrt{\frac{2E_{min}}{T_s}} a_i \cos 2\pi f_0(t) + \sqrt{\frac{2E_{min}}{T_s}} b_i \sin 2\pi f_0(t) \dots\dots\dots 1$$

$$0 \leq t \leq T_s, I = 1,2,3,4, \dots, 16 \dots\dots\dots 2$$

dimana Emin adalah energi dari sinyal dengan amplitudo terendah, ai dan bi adalah sepasang bilangan bulat independen yang dipilih sesuai dengan lokasi titik sinyal tertentu, f0 adalah frekuensi carrier, Ts adalah periode simbol [7].

**E. FFT (Fast Fourier Transform) dan IFFT (Inverse Fast Fourier Transform)**

FFT (Fast Fourier Transform) yaitu sebuah algoritma yang digunakan untuk menghitung transformasi diskrit Fourier. Transformasi Fourier mengubah informasi sinyal domain dalam waktu menjadi domain dalam frekuensi. Akibatnya, FFT secara luas dipakai dalam teknik DSP dan dalam banyak aplikasi seperti komunikasi telekomunikasi. FFT digambarkan sebagai algoritma numerik yang paling penting dan merupakan salah satu operasi yang belum sempurna di bidang sinyal digital. Menggunakan FFT sangat diperlukan dalam kebanyakan aplikasi pemrosesan sinyal. Hal yang paling penting dan diinginkan dalam banyak aplikasi adalah kecepatan data yang tinggi dan pengandaan sinyal yang efisien [9]. FFT dan IFFT adalah salah satu fungsi utama dalam sistem OFDM. Di sisi pemancar, IFFT digunakan untuk mengkonversi domain frekuensi Xk ke dalam domain waktu xm. Di sisi penerima, FFT digunakan untuk mengubah domain waktu xm menjadi domain frekuensi Xk [9]. Persamaan IFFT ditujukan pada Persamaan 3 sebagai berikut:

$$x_m = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k \exp\left(\frac{j2\pi km}{N}\right) \text{ for } 0 \leq m \leq N - 1 \dots\dots 3$$

Sedangkan untuk FFT ditujukan pada persamaan 4 sebagai berikut :

$$X_k = \frac{1}{\sqrt{N}} \sum_{m=1}^{N-1} x_m \exp\left(\frac{-j2\pi km}{N}\right) \text{ for } 0 \leq k \leq N - 1 \dots\dots 4$$

dimana untuk persamaan IFFT dan FFT, x\_m adalah nilai untuk domain waktu, X\_k adalah nilai untuk domain frekuensi, dan N adalah ukuran atau besar FFT.

**F. MIMO (Multiple Input Multiple Output)**

Komunikasi nirkabel atau wireless memiliki saluran nirkabel yang bermacam-macam sehingga memungkinkan terdapat suara yang berbeda-beda yang disebabkan karena multipath dan fading. Agar kebisingan berkurang dan juga kecepatan data naik, skema MIMO telah diusulkan dalam komunikasi nirkabel optik seperti modulasi spasial. Metode transmisi modulasi MIMO dan indeks modulasi telah mulai menarik perhatian yang signifikan dalam literatur VLC untuk mengirimkan bit tambahan dengan mengurangi konsumsi energi. Kanal MIMO dapat dideskripsikan sebagai berikut :

$$y = Hx + n \dots\dots\dots 5$$

dimana

$$\mathbf{y} = [y_1, y_2, \dots, y_{N_r}]^T \dots\dots\dots 6$$

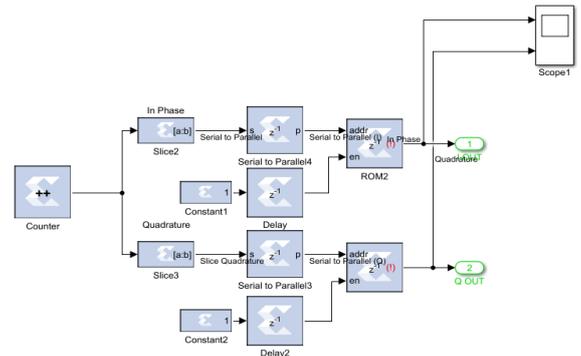
$$\mathbf{x} = [x_1, x_2, \dots, x_{N_t}]^T \dots\dots\dots 7$$

Dimana N<sub>t</sub> sebagai jumlah unit *transmitter* dan N<sub>r</sub> adalah jumlah unit *receiver* dalam skema MIMO. [10].

**G. FPGA (Field Programmable Gate Array)**

FPGA muncul sebagai teknologi logika sederhana yang menyediakan konektivitas terprogram antara komponen utama dimana programabilitas didasarkan pada antifuse yaitu EPROM atau SRAM. FPGA adalah perangkat yang berisi matriks sirkuit logic reconfigurable gate array. Ketika FPGA dikonfigurasi, sirkuit internal terhubung dengan cara menciptakan implementasi

perangkat keras atau hardware dari aplikasi perangkat lunak. Tidak seperti prosesor, FPGA menggunakan perangkat keras khusus untuk memproses logika dan pada perangkat FPGA tidak memiliki sistem operasi. Kinerja satu bagian dari aplikasi FPGA tidak terpengaruh ketika pemrosesan tambahan sedang ditambahkan. Beberapa loop kontrol dapat berjalan pada perangkat FPGA dengan tingkat yang berbeda.



Gbr.2 Blok Simulink 16QAM

### III. PERANCANGAN

#### A. Merancang blok-blok simulasi H-GLIM OFDM

Pada simulasi matlab, blok-blok telah dirancang sebelumnya dan terdapat perubahan dalam blok IFFT. Blok IFFT diubah kedalam versi 9.1 dan nilai bit length diubah dari 1024 menjadi 8 bit.

#### B. Blok Modulasi 16 QAM

QAM (Quadrature Amplitude Modulation) digunakan untuk mapping dari sebuah urutan data ke dalam symbol kompleks. QAM terdiri dari 2 tipe nilai, yaitu In-phase dan Quadrature dimana In-phase mempresentasikan nilai real dan quadrature mempresentasikan nilai imajiner, dan tiap permulaannya dari nilai  $n$  dimana nilai  $n$  bergantung dari ukuran  $M$ ,  $2n = M$ .

Dalam modulator 16 QAM terjadi modulasi data menjadi simbol kompleks yang menunjukkan salah satu dari 16 titik simbol konstelasi QAM. Pada 16 QAM blok, terdiri dari subsistem berupa blok counter, slice, serial to parallel, delay, constant, ROM. Blok Counter adalah blok fungsi yang dapat menghasilkan peningkatan atau pengurangan data dengan berbagai tipe data output.

Blok counter memiliki 4 bit unsigned fixed-point yang diatur kedalam tipe data output dan dihitung dari 0 hingga 15 karena dalam 4 bit ada 16 kemungkinan data dan juga modulasi baseband yang digunakan adalah 16-QAM. Output counter dapat dispesifikasikan dalam bentuk fixed-point yang signed atau unsigned.

#### C. Blok IFFT (inverse fast fourier transform)

Blok IFFT digunakan untuk mengkonversi vektor domain frekuensi  $X$  ke dalam domain waktu vektor  $x$ . Parameter khusus untuk blok IFFT Xilinx adalah transform length yang digunakan untuk memilih ukuran titik yang diinginkan mulai dari 8 hingga 65536. Kemudian scale result by FFT length digunakan bila dipilih maka data akan diskalakan antara tahapan IFFT menggunakan jadwal skala yang ditentukan oleh pengaturan dari transform length. Jika tidak dipilih, data unscaled, dan semua pertambahan bit integer dibawa ke output. Selanjutnya terdapat optimize for digunakan untuk mengarahkan blok agar dioptimalkan untuk kecepatan (kinerja) atau area (resource) di hardware yang dihasilkan.

#### D. Blok MCode

MCode digunakan untuk memisahkan setiap nilai indeks input yang menghasilkan nilai positif dan nilai negatif. Pertama ia akan membandingkan nilai input dengan 0, kemudian ketika pernyataan itu terpenuhi, ia akan mengarah ke port output positif 7 atau negatif, dan karena nilai positif dapat dikirim melalui port output, tetapi tidak dengan nilai negatif sehingga operasi ekstra diperlukan untuk mengkonversi nilai negatif menjadi nilai positif dengan mengatasi port output negatif sebagai nilai mutlak input.

#### E. Generate Netlist System Generator

Setelah Simulink berhasil dilakukan, step selanjutnya adalah generate file Simulink ke Vivado. Langkah yang dilakukan adalah mengklik token system generator pada Simulink, lalu akan muncul option box yang akan dipilih. Terdapat menu part yang digunakan untuk memilih model board

FPGA untuk disesuaikan dengan FPGA yang akan digunakan. Dalam hal ini, FPGA yang digunakan adalah model FPGA artix7 xc7a36ti-1lcs324.

Selanjutnya terdapat menu compilation dimana file Vivado akan disimpan, terdapat 2 pilihan dalam menu compilation yaitu hdl netlist dan IP catalog. Setelah option di setting, langkah selanjutnya adalah generate file ke format Vivado menjadi file .xpr.

#### F. Vivado Design Suite

Vivado Design Suite adalah software diproduksi oleh Xilinx untuk sintesis dan analisis desain-desain HDL untuk sistem pada pengembangan chip dengan sintesis yang memiliki kecepatan tingkat tinggi. Langkah pertama setelah generate pada token system generator berhasil dan file telah berekstensi menjadi .xpr, hal selanjutnya adalah membuka software Vivado untuk menjalankan tahap-tahap berikutnya seperti sintesis, implementasi dan juga generate bitstream agar hasil dari rangkaian H-GLIM-OFDM dapat di implementasikan pada hardware FPGA.

#### G. Digilent Board File

Secara default setelah Vivado terinstall, board yang ada adalah seperti Arty. Data board ini sesuai dengan direktori. Folder tersebut memuat file XML yang berisi informasi Slide Switches, Push Buttons, LEDs, USB-UART, DDR Memory, Ethernet, dan lain-lain.

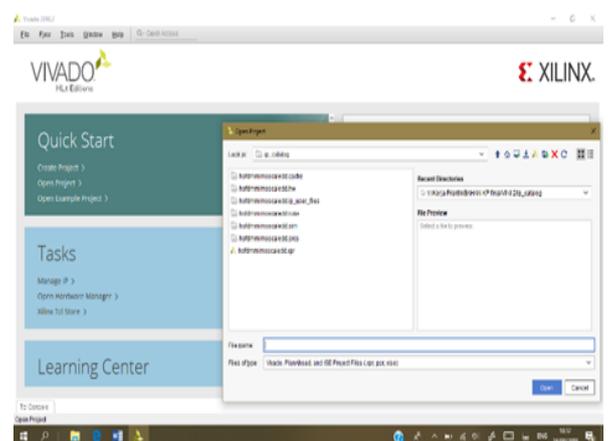
#### H. Tahap Synthesis pada Vivado

Pada tahap ini merupakan tahap synthesis yang berfungsi sebagai simulasi dari block diagram atau file dari hasil generate Simulink yang dicoba dengan menggunakan simulasi Vivado, berbeda dengan tahap implementasi dimana file akan di simulasikan sama seperti board FPGA yang akan digunakan.

Dengan desain pemodelan synthesis, run synthesis digunakan untuk mencoba simulasi dari block-block diagram yang sudah dibuat, kemudian memilih file directory dan number of jobs yang akan di gunakan. Pada tahap ini memiliki set up debug, dimana untuk memastikan wiring block diagram yang

dibuat memiliki clock domain yang sama. Untuk menghasilkan file .ltx yang dapat digunakan pada saat program device. Tahapan selanjutnya setelah generate netlist adalah tahap synthesis. Pada tahapan synthesis, terjadi perubahan deskripsi menjadi netlist yang akan dipetakan. Dengan desain pemodelan synthesis, delay net mungkin akan tidak sangat akurat dan juga pada synthesis netlist harus memenuhi timing atau akan terjadi kegagalan dengan jumlah yang kecil sebelum memulai implementasi.

Terdapat 3 macam constraint pada synthesis diantaranya RTL Attributes yaitu berdasarkan arahan yang ditulis dalam file RTL (MARK\_DEBUG, dll). Selanjutnya terdapat timing constraint (XDC) sebagai create\_clock, create\_generated\_clock, set\_input\_delay, set\_output\_delay, set\_clock\_groups, set\_false\_path, set\_max\_delay, dan set\_multicycle\_path. Kemudian Physical and Configuration Constraints dimana pada pilihan ini constraint menolak dalam hal synthesis algoritma. Constraints digunakan untuk menentukan tujuan kinerja yang diinginkan. Hal pertama setelah mengklik open Project adalah membuka folder yang berekstensi .xpr lalu file akan muncul pada panel Vivado.

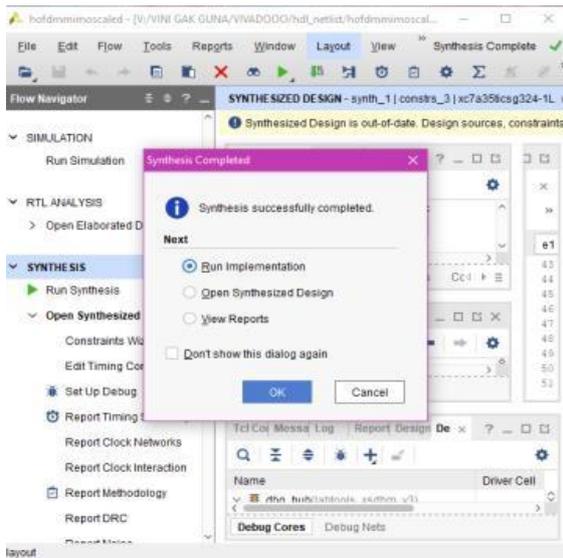


Gbr.3 Open Project pada Vivado

Selanjutnya run systemis dimana dalam tahap sintesis terdapat beberapa options seperti run synthesis dan open synthesized design. Pada option open synthesized design terdapat pilihan constraints wizard, edit timing constraint, set up debug, dan report. Timing constraint digunakan untuk memberitahu tools seperti jika ingin output kita diatur untuk memenuhi dalam waktu

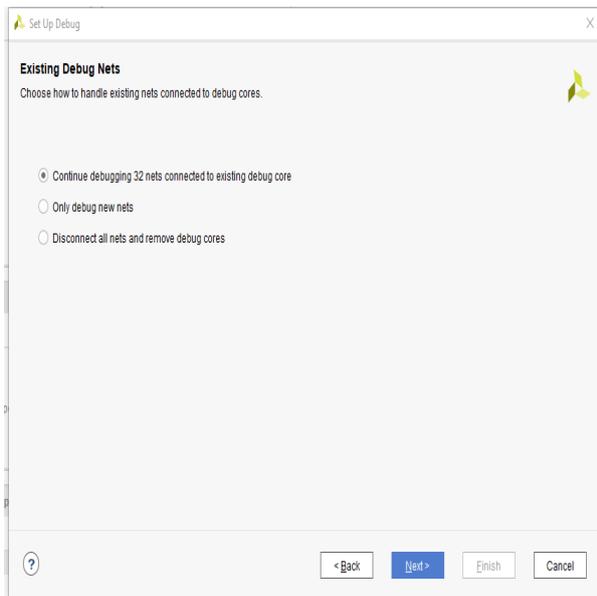
berapa terhadap DAC, jika kita memiliki clock domain W,X, Y, dan Z dimana W dan Z sinkron dan X dan Y tidak sinkron dengan setiap clock lainnya, dan juga jika kita ingin memiliki waktu delay maksimum antara satu dengan yang lainnya.

Untuk melakukan setup debug, hal pertama yang dilakukan adalah mensitisasi desain terlebih dahulu dan setelah selesai mengklik all semua skematik, lalu klik kanan dan tekan mark debug.

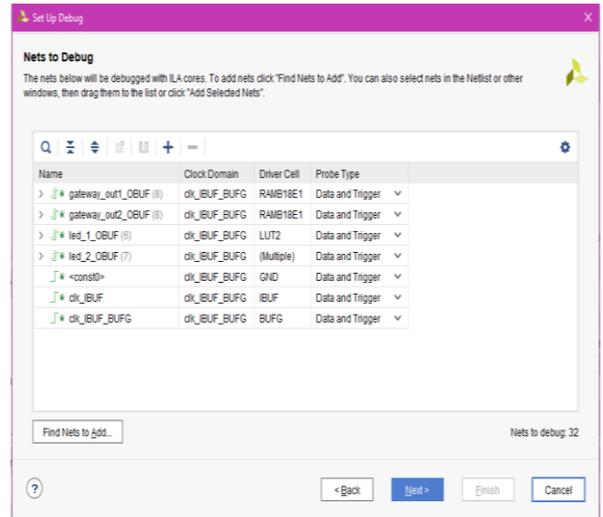


Gbr.4 Synthesized Design

Setelah itu klik opsi setup debug pada sub menu synthesis. Maka core akan bias dilihat pada folder netlist dan akan muncul box seperti dibawah ini.



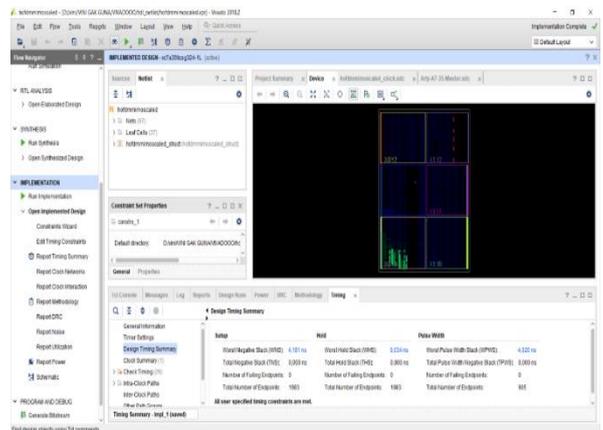
Gbr. 5 Setting Pertama Untuk Setup Debug



Gbr. 6 Setting Kedua untuk Setup Debug

I. Tahap Implementation pada Vivado

Pada tahap implementation, hal yang akan dilakukan meliputi translating, mapping atau fitting dan placing atau routing untuk mengimplementasikan desain ke device Xilinx yang sesuai.



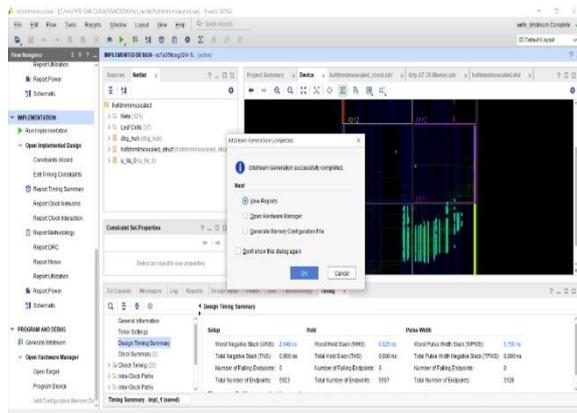
Gbr.7 Design Device pada Tahap Implementation

J. Tahap Program and Debug pada Vivado

Setelah selesai dalam tahap implementasi, tahap selanjutnya adalah menjalankan desain kedalam hardware dengan memogram FPGA dan debugging desain kedalam system. Semua perintah yang diperlukan untuk menjalankan program pada FPGA berada pada tahap program and debug. Terdapat 2 dasar tahap programming hardware ke FPGA yaitu generate file data bitstream dari desain yang telah di implementasikan. Kedua, menyambungkan ke hardware dan mendownload file yang telah terprogram menuju ke target yaitu ke FPGA.

### K. Generate Bitstream

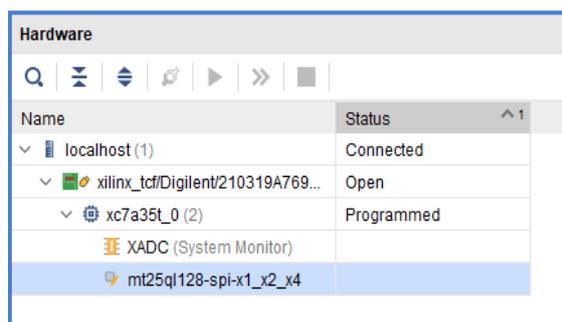
Setelah implementasi berhasil, tahap selanjutnya adalah menjalankan generate bitstream sampai tahap ini berhasil. Setelah generate bitstream berhasil, hal selanjutnya adalah menambahkan configuration memory device dengan cara menghubungkan terlebih dahulu program dengan hardware, lalu klik menu pada add configuration memory device dan akan terlihat macam-macam device yang telah tersedia. Setelah itu akan tersedia beberapa pilihan memory part. Pada pemrograman digunakan tegangan 3,3 volt sehingga dipilih memory part mt25q1128-spi-x1\_x2\_x4 (n25q128-3.3v-x1\_x2\_x4) dengan manufacturing micron type spi dan klik OK. Setelah terpilih device yang akan ditambahkan, maka akan muncul box alert untuk memogram program tersebut. Maka kita akan diminta untuk memanggil file konfigurasi yang telah dibuat.



Gbr. 8 Generate Bitstream Complete

### L. Open Target

Langkah selanjutnya setelah *generate bitstream file* adalah *download* program menuju target FPGA. Hal pertama adalah mengklik *open target* sehingga akan membuka target *hardware* yang baru atau yang sudah ada.



Gbr. 9 Open Target

### M. Program Device

Setelah program telah terhubung dengan FPGA, hal selanjutnya adalah memogram FPGA dengan program yang telah dibuat sebelumnya. File yang akan diminta oleh box alert program device adalah file bitstream dan juga file debug probe dengan ekstensi .ltx. Setelah terhubung ke target perangkat keras, maka selanjutnya menambahkan perangkat memori konfigurasi (Add Configuration Memory Device) pada Flow navigator. Lalu akan tersedia beberapa pilihan memory part. Pada pemrograman digunakan tegangan 3,3 volt sehingga dipilih memory part mt25q1128-spi-x1\_x2\_x4(n25q128-3.3vx1\_x2\_x4) dengan manufacturing micron tipe spi. Setelah itu klik OK.

### N. FPGA (field programmable gate array)

FPGA merupakan sebuah IC digital yang digunakan untuk mengimplementasikan rangkaian digital. FPGA berbentuk komponen elektronika dan semikonduktor yang terdiri dari komponen gerbang terprogram (programmable logic) dan sambungan terprogram (interkoneksi). Versi FPGA yang digunakan adalah versi Arty dimana Arty merupakan platform pengembangan siap pakai yang dirancang untuk Artix-7 Field Programmable Gate Array (FPGA) dari Xilinx.

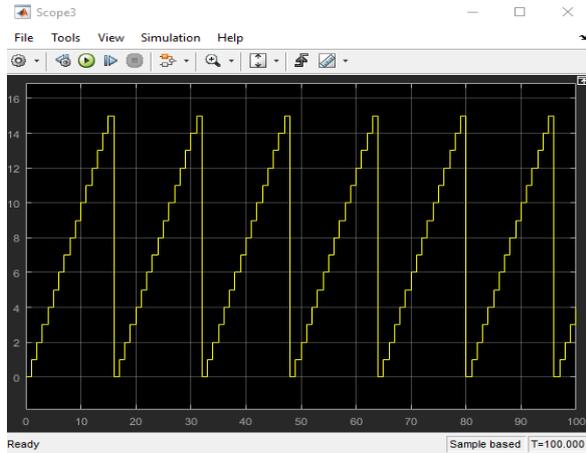
## IV. IMPLEMENTASI DAN HASIL

### A. Output dari QAM

Gambar 10 menunjukkan sinyal data keluaran dari blok counter, dimana axis y adalah nilai sinyal data dan axis x menunjukkan waktu atau periode. Sinyal data output yang dihasilkan dari blok counter perlu dipecah menjadi dua tipe data. Dalam modulasi baseband 16-QAM ada dua buah saluran, yaitu In Phase dengan sinyal data real dan Quadrature dengan sinyal data imajiner.

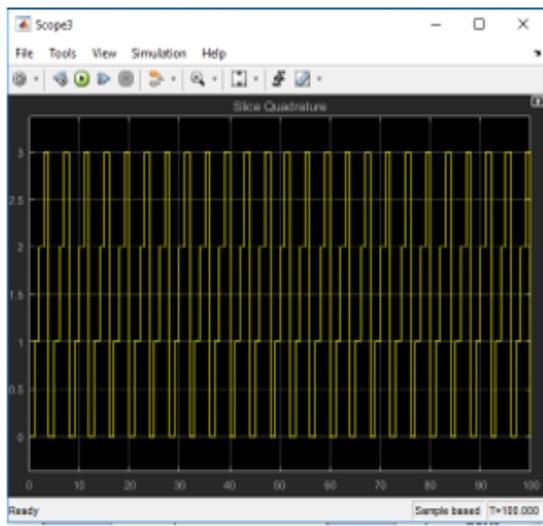
Operasi slice telah diimplementasikan dalam blok slice memecah sinyal input. Slice block adalah operasi untuk mengiris urutan data untuk menciptakan nilai baru dan pada blok ini terdapat 2 jenis blok slice yaitu In Phase dan quadrature. Parameter slice telah

diatur untuk mengiris 2 bit MSB untuk In Phase dan mengiris 2 bit LSB untuk Quadrature. Output dari setiap blok slice menghasilkan lebar 2 bit dan berisi 4 data



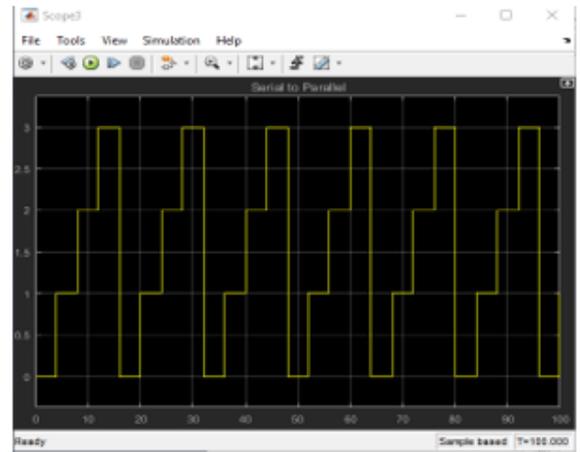
Gbr 10. Output Counter

Pada Gambar 11 menunjukkan keluaran data sinyal pada Inphase, dimana sumbu x menunjukkan waktu atau periode sedangkan sumbu y menunjukkan 4 data yang memungkinkan yaitu (0,1,2,3). Pada Inphase data sinyal berbentuk periodik setelah 16 siklus.



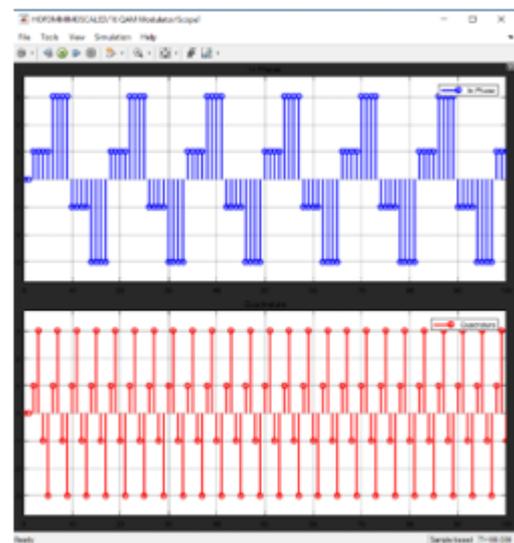
Gbr. 11 output dari In Phase

Pada Gambar 12 menunjukkan keluaran data sinyal pada Quadrature, dimana sumbu x menunjukkan waktu atau periode sedangkan sumbu y menunjukkan 4 data yang memungkinkan yaitu (0,1,2,3) selama counter digunakan dan menghasilkan data yang periodik dan pada quadrature data akan periodik 4 siklus.



Gbr. 12 Output Dari Quadrature

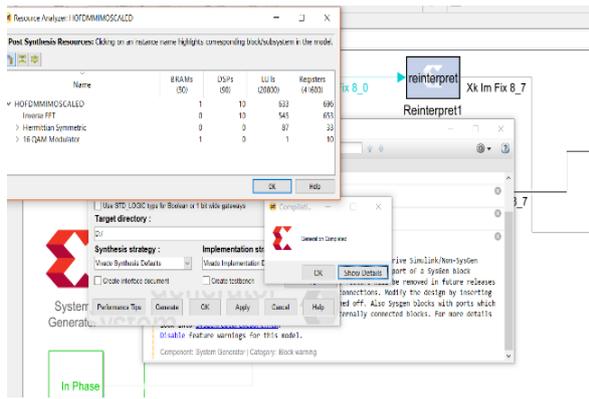
Pada Gambar 13 menunjukkan keluaran data sinyal 16QAM yang merupakan gabungan dari keluaran InPhase dan Quadrature, dimana sumbu x menunjukkan waktu atau periode sedangkan sumbu y menunjukkan 4 data yang memungkinkan yaitu (0,1,2,3) selama counter digunakan dan menghasilkan data yang periodik, pada Inphase data sinyal berbentuk periodik setelah 16 siklus dan pada quadrature data akan periodik 4 siklus.



Gbr. 13 Output Dari 16QAM

**B. Generate Netlist System generator**

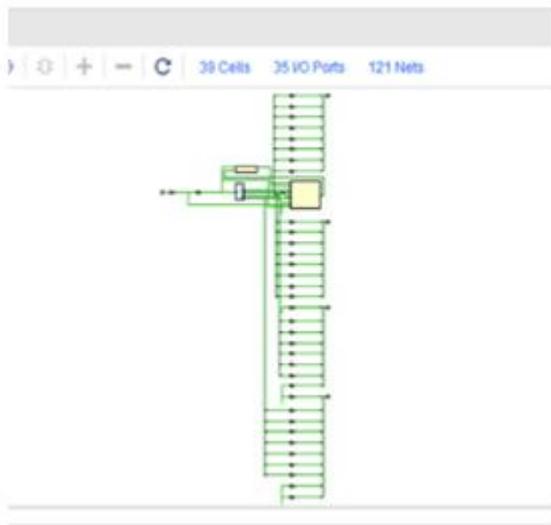
Setelah Simulink berhasil dilakukan, step selanjutnya adalah generate file Simulink ke Vivado. Langkah yang dilakukan adalah mengklik token system generator pada Simulink, lalu akan muncul option box yang akan dipilih. Terdapat menu part yang digunakan untuk memilih model board FPGA untuk disesuaikan dengan FPGA yang akan digunakan.



Gbr. 14 Generate Netlist System Generator Success

C. Tahap Synthesis pada Vivado

Dalam tahap sintesis terdapat beberapa options seperti run synthesis dan open synthesized design. Pada option open synthesized design terdapat pilihan constraints wizard, edit timing constraint, set up debug, dan report.

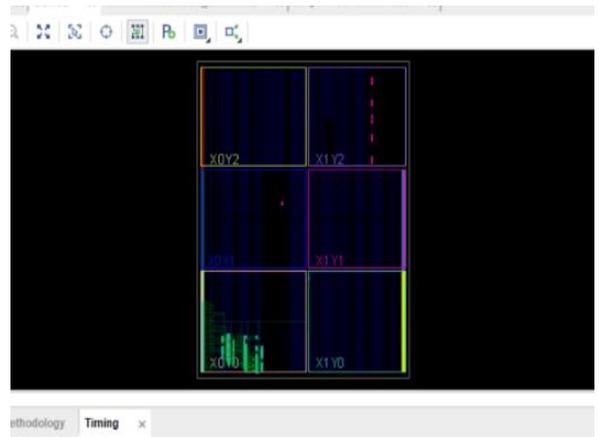


Gbr. 15 Skema Design Netlist Setelah disintesis

Nets yang terpilih terdiri dari gateway out1, gateway out2, LED 1, LED 2, konstanta dan clock sesuai dengan keluaran pada rangkaian desain yang telah dibuat. Setelah diatur setup debug, maka tahap selanjutnya adalah tahap implementasi.

D. Tahap Implementation pada Vivado

Pada tahap implementation, hal yang akan dilakukan meliputi translating, mapping atau fitting dan placing atau routing untuk mengimplementasikan desain ke device Xilinx yang sesuai.

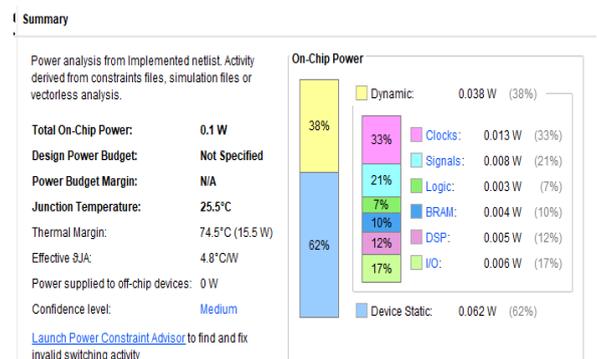


Gbr.16 Implementation Success Tanpa Error

Hasil dari power summary menunjukkan memory yang terpakai sangat besar yaitu sebesar 38% menyebabkan performa pada skema H-GLIM-OFDM ini berkurang. Pada memory menghasilkan sebesar 33% Clocks, 21% Signals, 7% Logic, 10% BRAM, 12% Digital Signal Processing (DSP), dan 17% Input/Output (IO).

Design Entry pada implementasi dapat dilakukan dengan cara Schematic capture program, VHDL / Verilog Logic Optimization. Dengan adanya implementasi, kita dapat memodifikasi Boolean Expression untuk mengoptimalkan penggunaan Logic berbasis area dan kecepatan atau kombinasi keduanya.

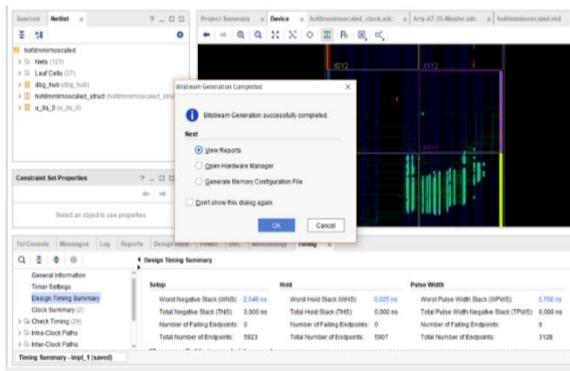
Selanjutnya pada implementasi terdapat technology mapping yaitu berfungsi untuk transformasi dari boolean expression atau VHDL kedalam bentuk circuit pada FPGA logic blocks. Technology mapping pada implementasi juga digunakan untuk area optimization yang berguna untuk meminimalkan penggunaan block dan delay optimization digunakan untuk meminimalkan jumlah stage pada critical path Placement.



Gbr. 17 Power Summary

### E. Tahap Generate Bitstream

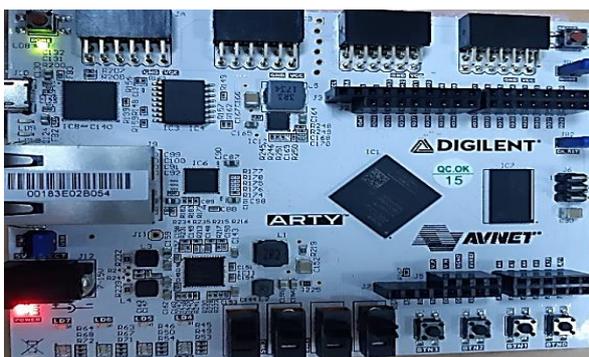
Setelah generate bitstream berhasil, hal selanjutnya adalah menambahkan configuration memory device dengan cara menghubungkan terlebih dahulu program dengan hardware, lalu klik menu pada add configuration memory device dan akan terlihat macam-macam device yang telah tersedia. Setelah itu akan tersedia beberapa pilihan memory part. Pada pemrograman digunakan tegangan 3,3 volt sehingga dipilih memory part mt25q11128-spi-x1\_x2\_x4 (n25q128-3.3v-x1\_x2\_x4) dengan manufacturing micron type spi dan klik OK. Setelah terpilih device yang akan ditambahkan, maka akan muncul box alert untuk memogram program tersebut.



Gbr. 18 Generate Bitstream Complete

### F. Tahap FPGA (Field Programmable Gate Array)

Dalam pengimplementasian dari Vivado ke perangkat keras atau hardware FPGA, pendeklarasian pin pada tahap generate bitstream menunjukkan telah berhasilnya pin-pin sesuai dengan letak LED pada FPGA namun lampu LED yang merupakan indikator belum berhasil hidup pada bagian PMOD JA dan JB FPGA.



Gbr. 19 Hasil Impletasi H-GLIM-OFDM pada FPGA

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Adapun kesimpulan yang diperoleh bahwa penggunaan Hermitian GLIM-OFDM dalam komunikasi wireless adalah untuk mentransmisikan sinyal menggunakan bilangan bipolar dengan cara memproses nilai imajiner dan real kedalam real positif menggunakan separator atau pemisah. Lalu pada proses sintesis dan implementasi desain menggunakan Vivado menghasilkan analisis penggunaan komponen untuk Hermitian GLIM-OFDM sebesar 33% Clocks, 21% Signals, 7% Logic, 10% BRAM, 12% Digital Signal Processing (DSP), dan 17% Input/Output (IO). Hasil implementasi Hermitian GLIM-OFDM pada FPGA dapat dikatakan terjadi kegagalan dikarenakan pada tahapan implementasi belum mampu menghidupkan indikator yaitu berupa LED pada FPGA.

### B. Saran

Hal yang harus diperhatikan adalah bahwa pemakaian Hermitian pada rangkaian GLIM-OFDM sebaiknya tidak perlu, dikarenakan dengan penambahan Hermitian ini berpengaruh terhadap pemakaian *memory* pada rangkaian dan juga fungsi GLIM sudah memiliki fungsi yang sama seperti Hermitian yaitu berfungsi sebagai penyimetri rangkaian OFDM.

## REFERENSI

- [1] R. A. Rajan Sagotra, "Visible Light Communication," *International Journal of Computer Trends and Technology*, vol. 4, no. 4, 2013.
- [2] E. P. M. U. a. H. H. E. Başar, "Generalized LED index modulation optical OFDM for MIMO visible light communications systems," pp. 1-5, 2016.
- [3] J. Armstrong, "FDM for Optical Communications," *ournal of Lightwave Technology*, vol. 27, no. 3, pp. 189-204, 1 february 2009.

- [4] K. P. J. A. Sarangi D. Dissanayake, "A novel technique to simultaneously transmit ACO-OFDM and DCO-OFDM in IM/DD systems," 2011.
- [5] E.B.F.M.E.P.M.U.H.H. Anil Yesilkaya, "Optical MIMO-OFDM with Generalized LED Index," 2016.
- [6] X. Li, "Simulink-based Simulation of Quadrature Amplitude Modulation (QAM) System," 2008.
- [7] A. K. P. Raghunandan Swain, "Design of 16-QAM Transmitter and Receiver: Review of Methods of Implementation in FPGA," *International Journal of Engineering and Science*, vol. 1, no. 9, pp. 23-27, 2012.
- [8] P. S. Hao LIN, "OFDM/OQAM with Hermitian Symmetry: Design," 2008.
- [9] P. V. I. ABHIJIT D. PALEKAR, "Ofdm System Using FFT and LFFT," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 12, 2013.
- [10] E.B.F.M.E.P.M.U.a.HH. Anil Yesilkaya, "Optical MIMO-OFDM With Generalized LED Index Modulation," *IEEE Transactions on Communications*, vol. 65, no. 8, 2017.
- [11] A. M. M. a. G. M. T. A. T. H. Makki, "Implementaion of 16-QAM transmitter and receiver on FPGA," *2nd World Symposium on Web Applications and Networking (WSWAN)*, pp. 1-4, 2015.
- [12] Kharagpur, "Module Design of Embedded Processors," Version 2 EE IIT.
- [13] A. R. & F. H. & N. T. Ndjiongue, "Visible Light Communications (VLC) Technology," pp. 1-15, 2015.