

IMPLEMENTASI *DISCRETE COSINE TRANSFORM* (DCT) PADA *FIELD PROGRAMMABLE GATE ARRAY* (FPGA) UNTUK APLIKASI KOMPRESI CITRA

IMPLEMENTATION OF *DISCRETE COSINE TRANSFORM* (DCT) ON *FIELD PROGRAMMABLE GATE ARRAY* (FPGA) FOR IMAGE COMPRESSION APPLICATIONS

Taufik Yumna¹, Rita Purnamasari, S.T., M.T.², Estananto, S.T., M.Sc.,MBA.³

^{1,2,3} Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

¹taufikyumna3004@gmail.com, ²ritapurnamasari@telkomuniversity.ac.id.

³estananto@telkomuniversity.ac.id

Abstrak

DCT adalah transformasi data yang mengubah dari domain waktu ke domain frekuensi, yang mampu memisahkan informasi pada frekuensi rendah hingga frekuensi tinggi. Manfaat dari transformasi ini cukup banyak, namun yang terasa saat era digital saat ini adalah pada kompresi citra. Dengan bantuan transformasi ini dapat mengurangi besar data yang dikandung oleh citra tanpa merusak kualitas citra tersebut.

Pada tugas akhir ini penulis telah merancang sistem DCT dengan $N=8$ dan $N=2$ dan berhasil diimplementasikan pada FPGA Altera Cyclone II – EP2C20F484C7. Penelitian ini bertujuan untuk membuktikan bahwa DCT dapat diimplementasikan pada FPGA dan hasilnya tidak jauh beda dengan perhitungan menggunakan MATLAB. Sistem ini dirancang menggunakan bahasa VERILOG yang merupakan bahasa hardware yang dimengerti oleh FPGA, pemilihan bahasa ini untuk memudahkan dalam mengkodekan perhitungan sistem DCT. Sistem ini dikodekan menggunakan Quartus Lite 16.0.0.2.1.1 dan disimulasikan dengan MODELSIM lalu kemudian disintesa pada FPGA. Hasil keluaran dari FPGA akan dibandingkan dengan hasil keluaran dari MATLAB.

Setelah disimulasikan dan didapatkan hasilnya, sistem disintesa ke *board* FPGA Altera Cyclone II – EP2C20F484C7. Dari hasil simulasi yang dapat diimplementasikan pada *board* FPGA Altera Cyclone II – EP2C20F484C7 hanya DCT dengan $N=2$ karena DCT dengan $N=8$ memakan *resource memory* yang cukup besar yaitu 3285% sedangkan sistem DCT dengan $N=2$ sebesar 92%. Dari hasil simulasi juga didapatkan perbandingan waktu delay dan BER antara perhitungan menggunakan MATLAB dan FPGA. Hasilnya adalah waktu delay yang dibutuhkan oleh MATLAB untuk menghitung DCT dengan $N=8$ adalah 0.10894 s dan dengan $N=2$ adalah 0.12107 s sedangkan waktu yang dibutuhkan oleh FPGA untuk menghitung DCT dengan $N=8$ adalah 0.00000024 s dan dengan $N=2$ adalah 0,00000136 s. BER yang dihasilkan antara MATLAB dengan FPGA adalah 0.2930 untuk DCT dengan $N=2$ dan 0.0503 untuk DCT dengan $N=8$.

Kata kunci : *Discrete Cosine Transform*, Kompresi Citra, *Field Programmable Gate Array*.

Abstract

DCT is a data transform that converts from time domain to frequency domain, it capable to separating informations from low frequency to high frequency. The benefits of this data transform quite a lot, because with the help of this data transform can reduce the amount of data contained by the image without damaging the image's quality.

In this final project the author has designed DCT system with $N = 8$ and $N = 2$ and successfully implemented on FPGA Altera Cyclone II - EP2C20F484C7. This study aims to prove that DCT can be implemented on FPGA and the result is not much different from the calculation using MATLAB. This system is designed using VERILOG language which is a hardware language dimengerti by FPGA, this language selection to facilitate in coding calculation of DCT system. The system is encoded using Quartus Lite 16.0.0.2.1.1 and simulated with MODELSIM and then synthesized on FPGA. The output of the FPGA will be compared with the output of MATLAB.

Once simulated and obtained the result, the system is synthesized to the Altera Cyclone II FPGA board - EP2C20F484C7. From the simulation results that can be implemented on the Altera Cyclone II FPGA board - EP2C20F484C7 only DCT with $N = 2$ because DCT with $N = 8$ consumes a large resource memory of 3285% while the DCT system with $N = 2$ is 92%. From the simulation results also obtained comparison of time delay and BER between calculations using MATLAB and FPGA. The result is the time delay required by MATLAB to calculate DCT with $N = 8$ is 0.10894 s and with $N = 2$ is 0.12107 s whereas the time required by FPGA to calculate DCT with $N = 8$ is 0.00000024 s and with $N = 2$ is 0, 00000136 s. The BER produced between MATLAB and FPGA is 0.2930 for DCT with $N = 2$ and 0.0503 for DCT with $N = 8$.

Keywords: *Discrete Cosine Transform*, Image Compression, *Field Programmable Gate Array*.

Pendahuluan

Perkembangan *Digital Signal Processing* (DSP) saat ini semakin pesat karena DSP banyak digunakan dalam berbagai hal, untuk bidang kesehatan, navigasi, komunikasi, dan lain-lain. Sinyal digital ditransformasikan agar mudah untuk diolah untuk kebutuhan sehari-hari, salah satu manfaat dari transformasi sinyal digital adalah di bidang komunikasi, yaitu kompresi data.

Salah satu teknik DSP adalah *Discrete Cosine Transform* (DCT), biasanya DCT digunakan untuk kompresi citra. Pada dasarnya DCT adalah transformasi data yang mengubah dari domain waktu ke domain frekuensi, yang mampu memisahkan informasi pada frekuensi rendah hingga frekuensi tinggi. Dalam citra informasi utama biasanya terkandung pada frekuensi rendah dan menengah, sedangkan pada frekuensi tinggi merupakan informasi detail. Ini sebagaimana visual mata manusia tidak begitu sensitif pada perubahan informasi detail tapi sangat sensitif pada informasi global. Ini menunjukkan jika bahwa jika menghilangkan informasi detail pada citra tidak menghilangkan kandungan informasi utama yang terkandung pada citra. Untuk menghilangkan informasi detail pada frekuensi tinggi bisa memanfaatkan DCT [1].

Field Programmable Gate Array (FPGA) adalah komponen elektronika dan semikonduktor yang mempunyai komponen gerbang terprogram (*programmable logic*) dan sambungan terprogram. FPGA dapat diprogram menggunakan bahasa *VHSIC Hardware Description Language* (VHDL) atau bahasa Verilog, bahasa pemrograman yang biasa digunakan untuk mendesain perangkat keras atau *Integrated Circuit* (IC), khususnya sistem digital [2].

Pada tugas akhir ini akan dilakukan penelitian tentang implementasi transformasi DCT pada FPGA. Pada penelitian sebelumnya sudah dilakukan implementasi DCT pada FPGA yaitu oleh Yan Felix Monangin dari Universitas Brawijaya dan Ronald Scrofano dari Los Angeles[3][4], namun pada penelitian kali ini dilakukan implementasi dengan merubah N DCT menjadi 2 yang pada umumnya 8.

1. Dasar Teori

2.1 Citra Digital [1]

Secara visual citra adalah representasi dari informasi yang terkandung di dalamnya sehingga manusia dapat menganalisa informasi yang terkandung di dalam citra tersebut. Namun secara matematis citra didefinisikan sebagai fungsi dua dimensi $f(x,y)$, di mana x dan y adalah koordinat spasial dan f adalah nilai intensitas warna pada koordinat x dan y . Bila nilai-nilai ini bersifat kontinu maka, citra disebut dengan citra analog, contohnya seperti yang ditampilkan pada layar monitor TV. Jika nilai-nilai ini bersifat diskret maka citranya disebut dengan citra digital, contohnya seperti yang tersimpan pada memori atau media penyimpanan data.

Citra digital umumnya merupakan representasi piksel-piksel dalam ruang 2D yang dinyatakan dalam matriks $N \times M$. Setiap elemen matriks disebut piksel (*picture element, image element* atau pel). Nilai setiap piksel f pada posisi koordinat x dan y yang mendeskripsikan intensitas warna yang dikodekan dalam 24 bit dengan komponen warna RGB ($R = red, G = green, B = blue$), dan 8 bit untuk citra *gray-level* atau 1 bit untuk citra biner.

$$f(x, y) = \begin{bmatrix} (0,0) & \dots & (0, N-1) \\ \vdots & \ddots & \vdots \\ (M-1,0) & \dots & (M-1, N-1) \end{bmatrix} \quad (2.1)$$

2.1.1 Jenis-jenis Citra

- Citra berwarna adalah citra yang secara visual memiliki kandungan informasi warna. Warna-warna ini dipresentasikan dalam piksel yang mengandung komponen *luminance, hue dan saturation/chrominance*.
- Citra *gray-level* adalah citra yang nilai pikselnya hanya mengandung komponen *luminance*, yang umumnya dikodekan dalam 8 bit. Artinya citra memiliki skala keabuan yang bervariasi dari 0 sampai 255 dimana 0 mempresentasikan warna hitam dan 255 mempresentasikan warna putih.
- Citra biner merupakan bagian dari citra *gray-level* yang hanya memiliki dua level keabuan, yaitu 0 untuk hitam dan 1 untuk putih, sehingga setiap piksel dalam citra hanya dikodekan menggunakan 1 bit.

2.1.2 Format File Citra

Format file citra standar yang banyak digunakan saat ini antara lain :

- Windows Bitmap* (BMP) adalah format file dari sistem operasi Microsoft Windows yang umumnya file BMP ini tidak dikompresi sehingga ukurannya besar.
- Tagged Image File Format* (TIFF) adalah format file untuk menyimpan data di perangkat Apple Macintosh, industri penerbit, designer grafis, fotografer amatir dan profesional.
- Graphics Interchange Format* (GIF) adalah format file grafik (bitmap) yang didasarkan pada jenis data raster 2D dengan berbagai jenis resolusi. GIF memiliki kelebihan dapat menyimpan beberapa citra dalam satu file sehingga dapat memberikan efek animasi saat ditampilkan pada layar komputer.
- Portable Network Graphics* (PNG) adalah format file citra bitmap yang menggunakan algoritma kompresi lossless. PNG merupakan format file *open source* (dapat digunakan tanpa lisensi hak cipta) dan diciptakan untuk memperbaiki kekurangan format file GIF.
- Joint Photographic Experts Group* (JPEG) adalah format citra yang dirancang agar bisa memampatkan data dengan rasio 1:16. Standar JPEG menentukan codec yang mendefinisikan bagaimana sebuah citra dikompresi ke dalam bitstream dan didekompresi ke dalam citra. Format JPEG menggunakan algoritma

kompresi yang bersifat *lossy* yang mana kualitasnya bergantung pada banyak data atau informasi yang hilang atau berubah. Rasio kompresi JPEG relatif besar namun dengan kualitas citra rekonstruksi yang tetap baik.

2.1.3 Kompresi citra

Kompresi citra adalah ilmu terkait *coding* citra digital secara efisien untuk mengurangi jumlah bilangan bit yang digunakan dalam merepresentasikan sebuah citra. Tujuan pengkompresian ini untuk mengurangi jumlah data dan tetap mempertahankan kualitas citra.

2.2 Discrete Cosine Transform (DCT) [1][4]

Proses transformasi DCT adalah proses transformasi data dari domain waktu ke domain frekuensi, yang mampu memisahkan informasi pada frekuensi rendah hingga frekuensi tinggi. Dalam sebuah citra, informasi utama yang dikandungnya lebih banyak terdapat pada frekuensi rendah dan menengah, sedangkan informasi yang berada pada frekuensi tinggi adalah informasi detail. Ini sebagaimana sifat visual mata manusia yang tidak begitu sensitif pada perubahan informasi detail tetapi sangat sensitif pada informasi global. Ini menunjukkan bahwa menghilangkan informasi detail citra tidak menghilangkan makna informasi yang terkandung pada citra. Untuk menghilangkan informasi detail ini dapat dilakukan melalui proses DCT.

Proses DCT 2-D ini dilakukan secara terpisah, dimulai dari proses DCT arah-x dan kemudian hasilnya dilakukan proses DCT arah-y. Secara matematis, proses DCT sebuah citra adalah jumlah perkalian antara fungsi cosinus diskret dan fungsi citra numerik 2-D $f(x,y)$ yang diberikan pada persamaan dibawah

$$F(u, y) = \sqrt{\frac{2}{N}} \cdot C_u \sum_{x=0}^{N-1} \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cdot f(x, y) \quad (2.2)$$

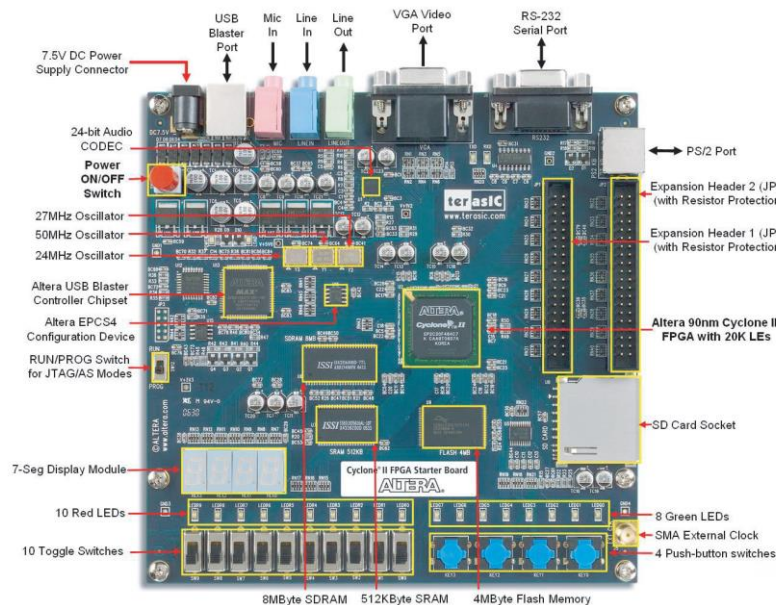
$$F(u, v) = \sqrt{\frac{2}{N}} \cdot C_v \sum_{x=0}^{N-1} F(u, y) \cdot \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (2.3)$$

$$C(i) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{bila } u, v = 0 \\ 1, & \text{bila } u, v > 0 \end{cases}$$

Persamaan (2.3) merupakan hasil dari proses DCT 2-D yang mempresentasikan domain frekuensi citra, di mana x, y adalah koordinat diskret dalam domain spasial citra dan u, v adalah koordinat diskret dalam domain frekuensi. N ukuran segmen matriks dalam sekali proses DCT, dimana dalam proses ini harus berukuran persegi-empat.

2.3 Field Programmable Gate Array [2][5][6]

Field Programmable Gate Arrays (FPGA) adalah komponen elektronika dan semikonduktor yang mempunyai komponen gerbang terprogram (*programmable logic*) dan sambungan terprogram. Komponen gerbang terprogram yang dimiliki meliputi jenis gerbang logika dasar (AND, OR, XOR, NOT) maupun jenis fungsi matematis dan kombinatorik yang lebih kompleks (*decoder, adder, subtractor, multiplier, dll*) yang dilingkupi oleh I/O dimana dapat dihubungkan satu sama lain melalui pemrograman menggunakan bahasa Verilog maupun VHDL. Blok-blok komponen di dalam FPGA bisa juga mengandung elemen memori (*register*) mulai dari flip-flop sampai pada RAM (*Random Acces Memory*).



Gambar 2.1 FPGA [5]

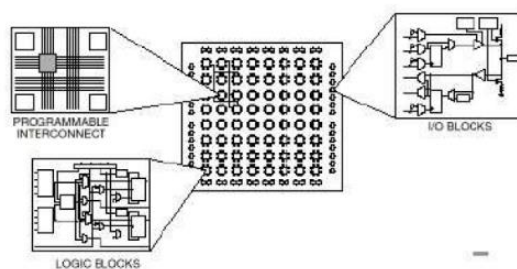
FPGA mempunyai kelebihan sebagai berikut :

- a. Dikonfigurasi oleh *End User*
- b. Tidak memerlukan proses Fabrikasi
- c. Tersedia solusi yang mendukung *chip customized VLSI*

Secara umum arsitektur bagian dalam dari IC FPGA terdiri atas tiga elemen utama yaitu *Input/Output Block (IOB)*, *Configurable Logic Block (CLB)* dan Interkoneksi [4].

1. *Configurable Logic Blocks*, yang memiliki fungsi :
 - a) Memiliki *Look up table* berdasarkan struktur kompleks komponen dalam FPGA.
 - b) Mengimplementasikan rangkaian sekuensial.
2. *Programmable Interconnect*, yang berfungsi:
 - a) Berisi *wire segments* (kabel penyambungan antar segmen) dan *programmable switches* (port switch yang biasa diprogram sebagai input rangkaian).
 - b) Menghubungkan antara *Configurable Logic Blocks* yang berbeda.

Input/output block, yang memiliki fungsi sebagai interface antara external package pin dari perangkat dan internal user logic pada FPGA

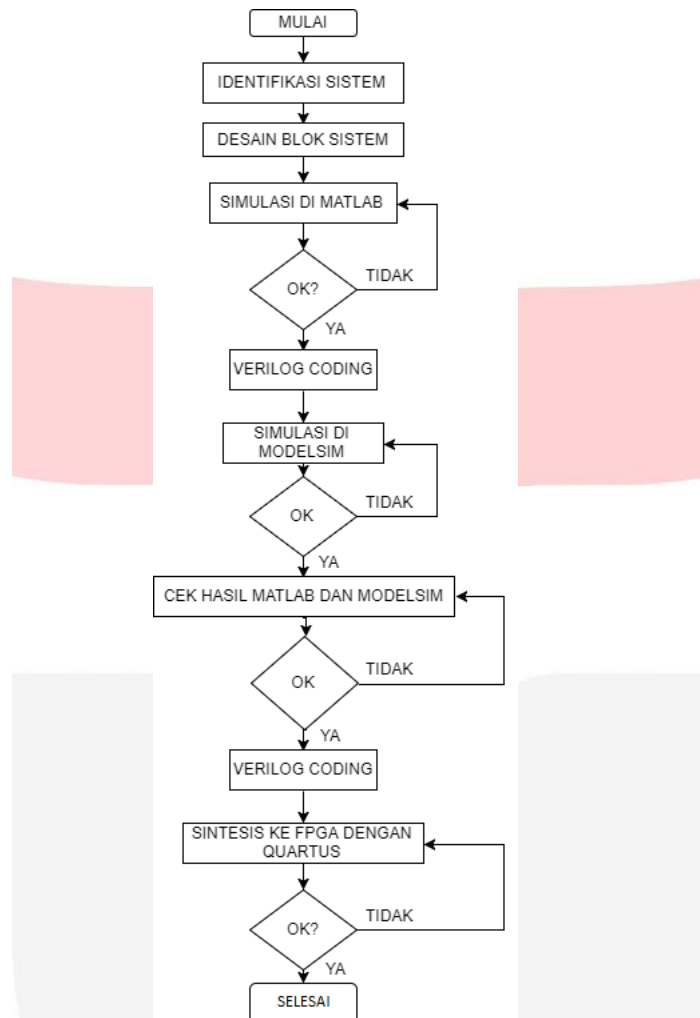


Gambar 2.2 Arsitektur Bagian Dalam FPGA [6]

3. Perancangan Sistem

3.1 Model Perancangan Penelitian

Untuk mengimplementasi sistem DCT pada *board* FPGA diperlukan beberapa tahapan. Berikut ini adalah perancangan sistem untuk mengimplementasikan DCT pada *board* FPGA



Gambar 3.1 Model Perancangan Penelitian

Perancangan sistem pada tugas akhir ini bertujuan untuk merancang algoritma *Discrete Cosine Transform* (DCT) yang akan diimplementasikan kepada FPGA. Perancangan sistem ini dimulai dari identifikasi sistem, mendesain blok sistem, lalu sistem disimulasikan pada software MATLAB. Sistem ini dirancang dengan masukan berupa citra *grayscale* berukuran 16x16. Sebenarnya pada software MATLAB sudah terdapat tool untuk menghitung DCT, namun pada tool MATLAB N DCT bernilai 8. Pada penelitian kali ini akan dicoba merubah N DCT menjadi 2 sehingga harus mendesain sistem di MATLAB untuk menghitung DCT dengan nilai N = 2. Berikut rumus DCT

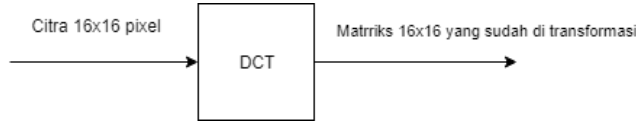
$$F(u, y) = \sqrt{\frac{2}{N}} \cdot C_u \sum_{x=0}^{N-1} \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cdot f(x, y) \quad (3.1)$$

$$F(u, v) = \sqrt{\frac{2}{N}} \cdot C_v \sum_{x=0}^{N-1} F(u, y) \cdot \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (3.2)$$

$$C(i) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{bila } u, v = 0 \\ 1, & \text{bila } u, v > 0 \end{cases}$$

Setelah blok sistem berhasil disimulasikan di MATLAB, kemudian dilakukan pengkodean dengan Bahasa VERILOG. Bahasa ini dipilih untuk memudahkan operasi matematika yang dapat dimengerti oleh *hardware*. Setelah pengkodean berhasil dilakukan simulasi pada MODELSIM. Simulasi ini bertujuan untuk memeriksa hasil perhitungan DCT benar atau tidak, jika benar hasil dibandingkan dengan hasil di MATLAB. Apabila hasil sudah se usai pengkodean VERILOG disintesis ke FPGA menggunakan software QUARTUS.

3.2 Perhitungan DCT



Proses pemodelan sistem DCT dimulai dari memberi masukan berupa citra 16x16 pixel yang sudah dirubah menjadi matriks yang berisi nilai pixel. Kemudian, pixel citra tersebut di transformasi DCT menggunakan persamaan yang sudah dibahas di BAB II. Persamaan tersebut dapat direpresentasikan dalam bentuk perkalian matriks :

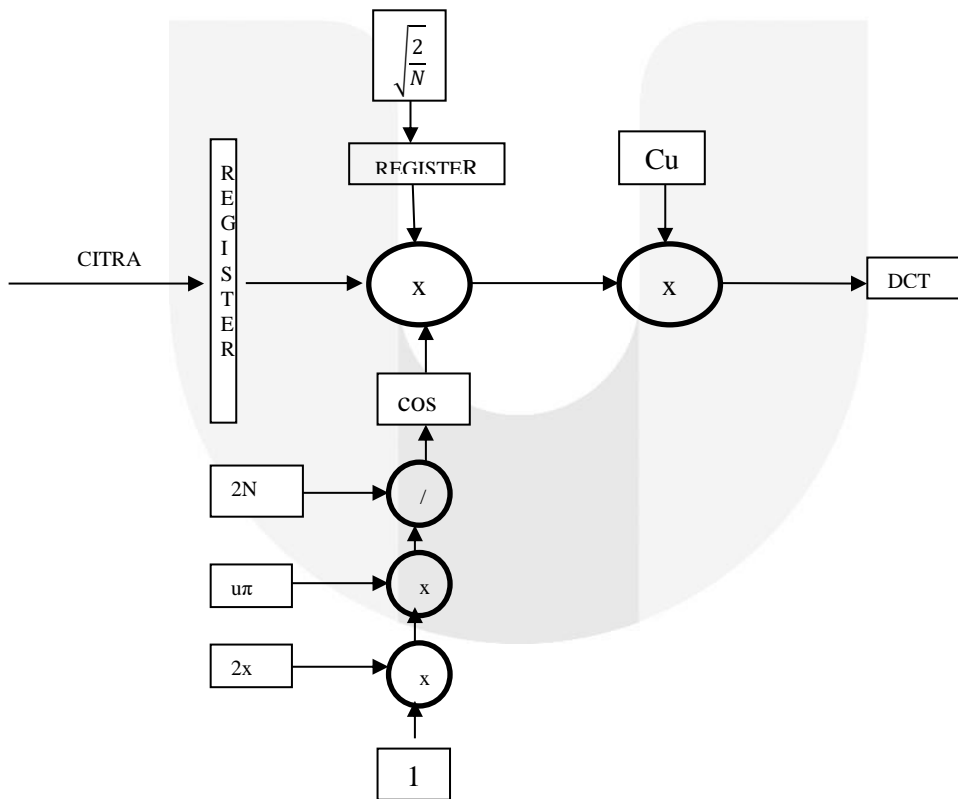
$$[F]u, v = ([DC]u, x \cdot [f]x, y) \cdot [DC]y, v$$

Dimana,

$$DC(u, x) = \sqrt{\frac{2}{N}} Cu \cos \left[\frac{(2x + 1)u\pi}{2N} \right]$$

$$DC(y, v) = \sqrt{\frac{2}{N}}, Cv \cos \left[\frac{(2y + 1)v\pi}{2N} \right]$$

yang berlaku umum untuk ukuran matriks persegi empat dan merupakan perpangkatan dua. Jika digambarkan dalam bentuk gambar blok digram adalah seperti berikut:



Gambar 3. 2 Blok Sistem Diagram

Pada penelitian ini dilakukan perbandingan nilai N pada DCT jika diberi nilai N = 2 dengan N = 8, yang akan dilihat performansinya di FPGA nanti. Proses perhitungan DCT bisa dilihat dibawah ini. Proses perhitungan dimulai dari menentukan DCx atau [DC]u,x, lalu menentukan Dcy atau [DC]y,v yang merupakan transpose dari DCx. Setelah itu DCx akan dikalikan dengan nilai pixel citra yang diuji. Untuk detilnya bisa dilihat pada perhitungan dibawah ini.

$$[DC]u, x = \begin{bmatrix} u0, x0 & u1x0 \\ u0, x1 & u1, x1 \end{bmatrix}$$

$$DC(u_0, x_0) = \sqrt{\frac{2}{2}} \frac{1}{\sqrt{2}} \cos \left[\frac{(2(0) + 1)(0)\pi}{2(2)} \right]$$

$$= 0,7071$$

$$DC(u_1, x_0) = \sqrt{\frac{2}{2}} \frac{1}{\sqrt{2}} \cos \left[\frac{(2(0) + 1)(1)\pi}{2(2)} \right]$$

$$= 0,7071$$

$$DC(u_0, x_1) = \sqrt{\frac{2}{2}} \frac{1}{\sqrt{2}} \cos \left[\frac{(2(1) + 1)(0)\pi}{2(2)} \right]$$

$$= 0,7071$$

$$DC(u_1, x_1) = \sqrt{\frac{2}{2}} \frac{1}{\sqrt{2}} \cos \left[\frac{(2(1) + 1)(1)\pi}{2(2)} \right]$$

$$= -0,7071$$

$$[DC]u, x = \begin{bmatrix} 0,7071 & 0,7071 \\ 0,7071 & -0,7071 \end{bmatrix}$$

Untuk mencari matriks $[DC]y, v$ tinggal mentransposekan hasil dari $[DC]u, x$ sehingga hasil dari DCT citra di atas dengan $N=2$ adalah sebagai berikut:

$$[DC]u, x = \begin{bmatrix} u_0, x_0 & u_1, x_0 \\ u_0, x_1 & u_1, x_1 \end{bmatrix} = \begin{bmatrix} 0,7071 & 0,7071 \\ 0,7071 & -0,7071 \end{bmatrix}$$

$$[DC]y, v = \begin{bmatrix} y_0, v_0 & y_1, v_0 \\ y_0, v_1 & y_1, v_1 \end{bmatrix} = \begin{bmatrix} 0,7071 & 0,7071 \\ 0,7071 & -0,7071 \end{bmatrix}$$

$$[f]x, y = \begin{bmatrix} 0 & 0 \\ 3 & 0 \end{bmatrix}$$

$$[F]u, v = ([DC]u, x \cdot [f]x, y) \cdot [DC]y, v$$

$$[F]u, v = \begin{bmatrix} 1,5 & 1,5 \\ -1,5 & -1,5 \end{bmatrix}$$

Proses perhitungan di atas akan berulang kepada seluruh nilai pixel citra.

Proses yang sama akan dilakukan jika merubah besar N menjadi 8, namun proses pengulangannya akan lebih sedikit karena jika $N=8$ proses perhitungan akan dilakukan sebanyak 4 kali saja.

3.3 Perangkat Lunak yang Digunakan

Dalam penelitian ini, penulis menggunakan beberapa perangkat lunak sebagai berikut.

1. MATLAB R2017a, digunakan untuk melakukan simulasi sistem dari DCT.
2. ModelSim ALTERA STARTER EDITION 10.1b, digunakan untuk melakukan simulasi VERILOG coding.
3. Quartus Lite 160.0.0.1.1.0, yang berfungsi :
 - a. Untuk menampilkan hasil perhitungan yang nantinya akan dikeluarkan oleh FPGA
 - b. Untuk mensintesa hasil rancangan pengkodean VERILOG ke *boardFPGA* menjadi rangkaian ataupun gerbang logika.

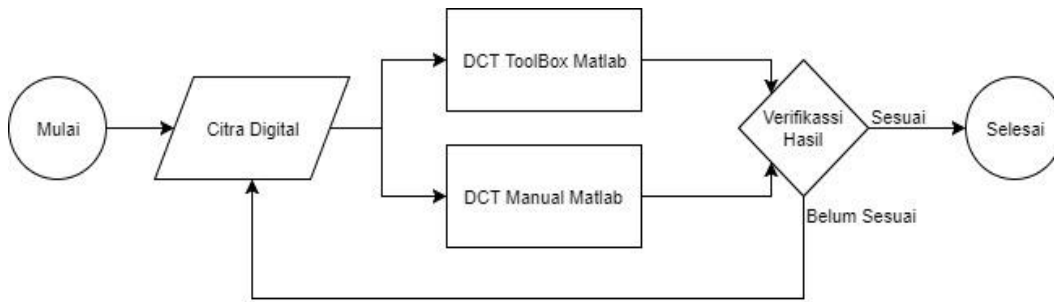
Perangkat Keras yang Digunakan

Dalam penelitian ini, penulis menggunakan beberapa perangkat keras sebagai berikut.

1. Laptop LENOVO S410P RAM 8GB, NVIDIA 840M, MEMORY 500GB
2. FPGA Altera Cyclone II EP2C20F484C7

4. Analisis

4.1 Simulasi pada Matlab

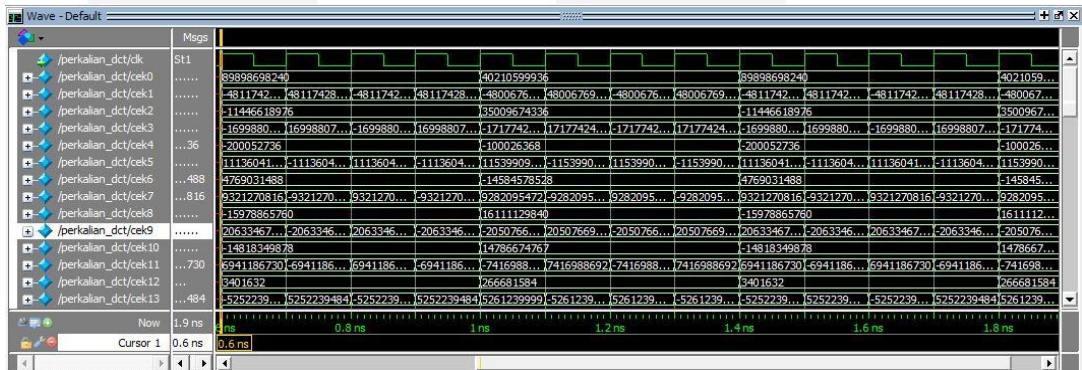


Dalam penelitian kali ini dilakukan simulasi sistem DCT pada Matlab dan hasil simulasi tersebut dijadikan acuan untuk proses sintesa pada FPGA. Seperti gambar diatas, penelitian ini dimuai dari memberikan masukan berupa citra *grayscale* berukuran 16×16 *pixels*. Citra tersebut ditransformasikan menggunakan DCT toolbox dan ditransformasikan juga dengan DCT manual atau dengan *source code* yang berdasarkan rumus DCT. DCT *toolbox* dijadikan sebagai acuan karena *source code* tersebut sudah dijadikan *toolbox* yang sudah lazim digunakan untuk para pengguna MATLAB. Default nilai N pada DCT *toolbox* adalah 8 sehingga pada saat membuat *source code* DCT manual, nilai N ditetapkan 8. Apabila hasil dari DCT manual sudah sama dengan DCT *toolbox* maka *source code* DCT manual yang sudah dibuat bisa dijadikan acuan untuk proses sintesa ke FPGA.



Gambar kiri adalah gambar masukan untuk sistem DCT dengan nilai $N=8$ dan gambar kanan adalah gambar yang telah di transformasi DCT. Simulasi pada MODELSIM

Langkah awal yang diperlukan sebelum sintesa transformasi DCT ke FPGA, diperlukan pengkodean awal dengan disimulasikan pada ModelSim. Agar sistem transformasi DCT yang penulis buat dapat dimengerti oleh FPGA maka dilakukan pengkodean dengan bahasa VERILOG dan *source code* yang dibuat disimulasikan pada ModelSim sebelum nanti akan disintesa pada FPGA.



898.7500	-481.1252	-114.4654	-169.9604	-2.0000	111.3498	47.6837	93.1685
-159.7724	206.3462	-148.2071	69.4109	0.0342	-52.5252	62.3673	-31.6047
-575.8640	144.8897	318.4486	43.9587	-2.4150	-33.4722	-135.5951	-29.7322
-57.3356	68.0399	-50.4751	25.0959	0.7144	-7.3029	31.7792	-5.5850
1.0000	3.0494	1.2998	-3.2279	3.2500	1.5651	3.7912	4.8433
33.2905	-47.6975	29.9875	-23.4524	-3.0169	6.5587	-3.7551	2.7115
237.7189	-56.0662	-127.0951	-34.1084	-3.8704	7.2827	55.0514	4.4321
32.1640	-35.2542	36.6514	-7.1463	-1.8740	2.6934	-8.7262	3.4993

Gambar di atas merupakan hasil sebagian simulasi pada sistem DCT pada ModelSim. Hasil keluaran tersebut telah dibandingkan dengan hasil keluaran pada MATLAB. Perbandingan lengkap hasil tersebut dapat dilihat di lampiran.

4.2 Sintesa dan Implementasi Sistem DCT ke FPGA

Setelah dilakukan simulasi pada Modelsim dan hasilnya sudah diverifikasi, maka dilanjutkan dengan implementasi dan sintesis sistem DCT ke dalam perangkat FPGA yang telah ditentukan. Sintesis dilakukan menggunakan Aplikasi Quartus yang telah dikoding menggunakan bahasa VERILOG. Kodingan VERILOG untuk sintesis ini diambil dari simulasi sebelumnya yang dilakukan pada ModelSim.

Flow Summary	
Flow Status	Flow Failed - Tue Jan 23 06:01:32 2018
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	project_tugas_akhir
Top-level Entity Name	perkalian_dct
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	615,919 / 18,752 (3285 %)
Total combinational functions	615,481 / 18,752 (3282 %)
Dedicated logic registers	11,395 / 18,752 (61 %)
Total registers	11395
Total pins	3,201 / 315 (1016 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	52 / 52 (100 %)
Total PLLs	0 / 4 (0 %)

Dari gambar di atas dapat dilihat total *logic element* yang digunakan sebanyak 615.919 sedangkan logic element yang tersedia pada FPGA sebanyak 18.752. Jadi sistem DCT dengan nilai $N=8$ tidak bisa disintesa ke dalam FPGA karena FPGA tidak kuat membuat rangkaian yang diinginkan.

Resource	Penggunaan Resource
Total Logic Elements	615919 / 18752 (3285%)
Total Memory Bits	0/ 239616 (0%)
Total Combinational Functions	615481 / 18752 (3282%)
Dedicated Logic Registers	11395/ 18752 (61%)
Total PINS	3201/ 315 (1016%)
Embedded Multiplier 9- bit Elements	52 / 52 (100%)

Tabel 4.4. Penggunaan Resource

4.3 Merubah N DCT Menjadi 2

Setelah melakukan percobaan pada $N DCT = 8$ dilakukan perubahan dengan merubah $N DCT$ menjadi 2. Proses yang terjadi sama seperti $N=8$. Karena saat $N=8$ dinilai membutuhkan memory yang sangat banyak, maka dipilih besar angka perpankangan 2 yang paling kecil yaitu 2.

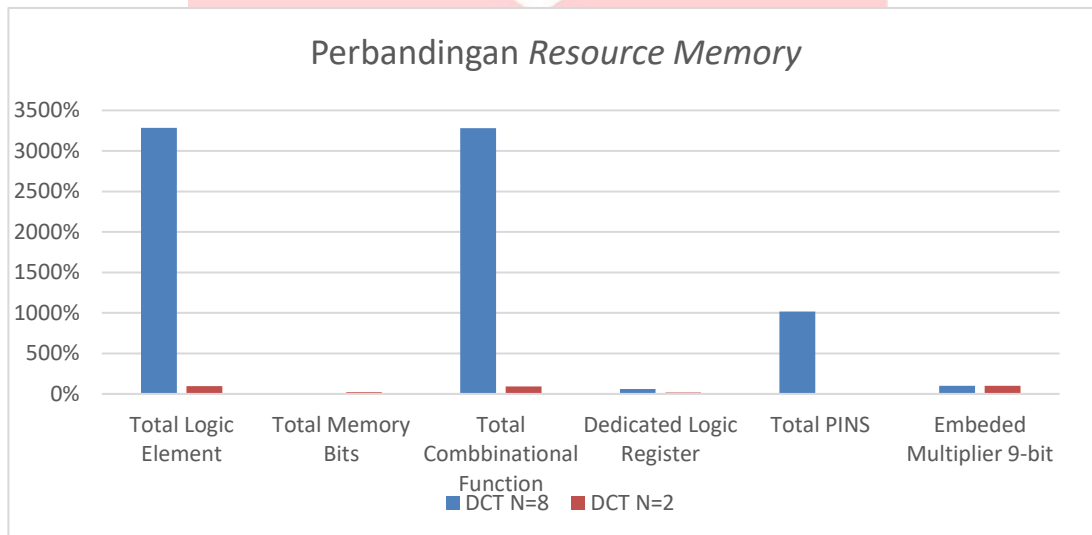
Flow Summary	
Flow Status	Successful - Sun Jan 21 16:01:42 2018
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	project_tugas_akhir
Top-level Entity Name	perkalian_dct
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	18,291 / 18,752 (98 %)
Total combinational functions	17,367 / 18,752 (93 %)
Dedicated logic registers	3,161 / 18,752 (17 %)
Total registers	3161
Total pins	5 / 315 (2 %)
Total virtual pins	0
Total memory bits	48,128 / 239,616 (20 %)
Embedded Multiplier 9-bit elements	52 / 52 (100 %)
Total PLLs	0 / 4 (0 %)

Dapat dilihat dari gambar di atas bahwa total *logic element* yang dibutuhkan untuk $N DCT = 2$ lebih kecil daripada $N DCT = 8$, yaitu sebanyak 18.291 atau 98% dari total logic element yang tersedia pada FPGA. Dengan hasil tersebut diputuskan untuk disintesa ke FPGA.

Resource	Penggunaan Resource
Total Logic Elements	18291 / 18752 (98%)
Total Memory Bits	48128/ 239616 (20%)
Total Combinational Functions	17367 / 18752 (93%)
Dedicated Logic Registers	3161/ 18752 (17%)
Total PINS	5/ 315 (2%)
Embedded Multiplier 9- bit Elements	52/52 (100%)

4.4 Perbandingan Resource Memory pada FPGA

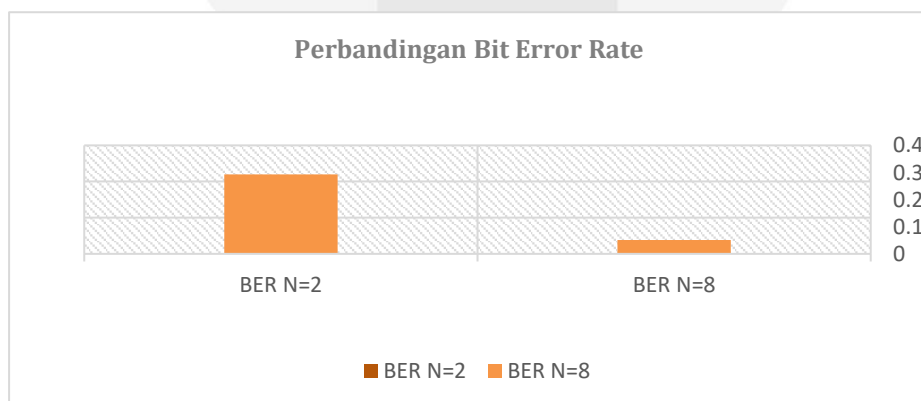
Dapat dilihat dari hasil penelitian bahwa resource memory untuk mengimplementasikan DCT pada FPGA cukup. Untuk meneliti bahwa DCT dapat diimplementasikan.



Dapat dilihat dari grafik diatas bahwa perbandingannya jauh sekali antara DCT N=8 dan DCT N=2, itu disebabkan karena kapasitas spesifikasi FPGA yang digunakan yaitu Cyclone II – EP2C20F484C7 tidak mumpuni untuk sistem DCT N=8 yang telah dibuat.

4.5 Perbandingan Waktu Delay dan BER yang Dihasilkan

Pada penelitian ini dihasilkan waktu delay untuk menghitung sisitem DCT yang telah dibuat dan BER (*Bit Error Rate*) atau nilai kesamaan yang dibandingkan antara hasil pada FPGA dengan MATLAB. Berikut adalah grafik perbedaan yang dihasilkan.



Dapat dilihat dari grafik di atas besar BER pada sistem DCT N=2 adalah 0.2930 sedangkan pada sistem DCT N=8 adalah 0.0508. Nilai BER diatas didapatkan dari hasil membandingkan hasil perhitungan menggunakan FPGA dengan MATLAB. Nilai BER pada sistem DCT dengan N=2 lebih besar dikarenakan pada sistem DCT N=2 terjadi banyak pengulangan sedangkan pada N=8, pengulangan terjadi hanya 4 kali saja.

Pada perhitungan delay sistem DCT diuji 4 citra, citranya adalah sebagai berikut: Gambar 4.9 merupakan citra masukan sistem citra tersebut berbeda ukuran yaitu 8x8 pixel dan 16x16 pixel, namun haya berbeda latarnya saja. Perbandingan delay yang dibutuhkan dapat dilihat pada tabel di bawah ini.

Tabel 4. 1 Perbandingan Delay



Gambar 4. 1 Citra Inputan Sistem

Input	N	MATLAB	FPGA
Citra 16x16 Latar Hitam	N=8	0,088005	0.00000024
	N=2	0,118108	0.00000136
Citra 16x16 Latar Putih	N=8	0,105505	0.00000024
	N=2	0,11234	0.00000136
Citra 8x8 Latar Hitam	N=8	0,076894	0,00000016
	N=2	0,095376	0,00000004
Citra 8x8 Latar Putih	N=8	0,070354	0,00000016
	N=2	0,163465	0,00000004

Pada perbandingan delay dibutuhkan beberapa citra, dapat dilihat pada tabel di atas delay dipengaruhi pada ukuran citra karena ukuran citra berpengaruh pada proses terjadinya DCT, jika ukuran citra semakin besar maka delay yang diperlukan juga semakin besar. Itu disebabkan karena terjadinya pengulangan untuk proses DCT. Pada FPGA satu clock bernilai 20ns, pada tugas akhir ini diprogram untuk setiap clock menampilkan nilai sebanyak matriks yang dihitung. Misalkan N=2 maka dalam satu clock menampilkan nilai sebanyak 4 buah karena dalam N=2 menghitung matriks sebesar 2x2. Namun jika melihat dari ketelitian MATLAB lebih teliti karena MATLAB mempunyai sebuah fungsi otomatis untuk menghitung DCT, sedangkan FPGA tingkat ketelitiannya tergantung pada spesifikasi dan juga pada saat pemrogramannya.

4.6 Implementasi Sistem DCT pada FPGA

Ada beberapa tahapan yang perlu diperhatikan dalam proses implementasi,

yaitu:

1. *Design Entry*, memasukkan kode VERILOG yang telah dirancang ke *Project* yang dibuat. Dan compile projectnya terlebih dahulu sebelum masuk ke langkah selanjutnya.
2. *Assign Package Pin (PIN Planner)*, memilih pin masukan dan keluaran sesuai dengan rancangan modul yang telah dibuat. Setelah mengisi pin alangkah baiknya di *compile projectnya* terlebih dahulu agar PIN tersebut terbaca di kodingannya.
3. *Programmer*, untuk mensintesis kodingan VERILOG ke *board* FPGA.
4. *Signal Tap II Quartus*, untuk menampilkan hasil perhitungan yang dihasilkan FPGA

4.7 Sintesa Sistem DCT dengan N=2 ke dalam FPGA

Dengan hasil running N=2 yang memakan memory 98 % dapat disimpulkan bahwa kodingan dapat disintesa ke FPGA. Dilakukan Sintesa ke FPGA dan berikut adalah gambar dari hasil Signal II tap pada Quartus. Tabel hasil lengkap dari sintesa dapat dilihat pada lampiran.

Pada tahap sintesa dilakukan dua kali run pada quartus karena untuk menampilkan hasil DCT membutuhkan memori sehingga jika menampilkan seluruh hasil DCT memori FPGA tidak cukup. Berikut setup yang dilakukan

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	94	94	1 Basic
IR		cek0[42..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXXXXXXXXXX
IR		cek1[42..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXXXXXXXXXX
IR		cek2[42..0]	<input type="checkbox"/>	<input type="checkbox"/>	
IR		cek3[42..0]	<input type="checkbox"/>	<input type="checkbox"/>	
IR		counter_clk_pelat	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
IR		vertikal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1

Dapat dilihat dari gambar di atas dilakukan setup untuk menampilkan hasil dari sintesa pada FPGA. untuk running pertama di ceklist untuk menampilkan hasil dari DCT yang berada pada titik 0,0 dan 1,0. Hasilnya seperti

berikut

		cek0[42..0] (cek0)	149997123	849983697	599988492
		cek1[42..0] (cek1)	149997123	549989451	599988492

Setelah dilihat hasil dari signal II tap lalu disamakan lagi dengan hasil pada modelsim. Jika hasil sudah sama maka dilakukan running kedua.

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	94	94	1 Basic
		cek0[42..0]	<input type="checkbox"/>	<input type="checkbox"/>	
		cek1[42..0]	<input type="checkbox"/>	<input type="checkbox"/>	
		cek2[42..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXXXXXXXXXXXX
		cek3[42..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXXXXXXXXXXXX
		counter_clk_pelan	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
		vertikal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1

Running kedua dilakukan lagi setup untuk menampilkan hasil selanjutnya yaitu di posisi 0,1 dan 1,1 yang diberi nama variabel cek2 dan cek3. Ceklist setup dapat dilihat pada gambar di atas. Hasilnya seperti berikut.

		cek2[42..0]	-149997123	549989451	-599988492
		cek3[42..0]	-149997123	849983697	-599988492

Setelah keluar hasil dari signal II tap pada quartus kembali disamakan hasilnya dengan ModelSim jika sama maka dinyatakan sintesa berhasil.

4.8 Perbandingan Hasil Perhitungan MATLAB, MODELSIM, dan FPGA

Berikut ini adalah perbandingan nilai yang dihitung menggunakan MATLAB, diprogram dan disimulasikan pada MODELSIM, dan disintesa pada FPGA.

n	MATLAB		MODELSIM dan FPGA	
	DCT N=8	DCT N =2	DCT N=8	DCT N=2
0	898.7500	1.5	898.9870	1.5000
1	-481.1252	1.5	-481.1743	1.5000
2	-114.4654	8.5	-114.4662	8.4998
3	-169.9604	5.5	-169.9881	5.4999
4	-2.0000	6	-2.0005	59.9999
5	111.3498	6	111.3604	59.9999
6	47.6837	11	47.6903	10.9998
7	93.1685	3	93.2127	2.9999
8	-159.7724	-1.5	-159.7887	-1.5000
9	206.3462	-1.5	206.3347	-1.5000
10	-148.2071	5.5	-148.1835	5.4999
11	69.4109	8.5	69.4119	8.4998
12	0.0342	-6.0	0.034	-59.9999
13	-52.5252	-6.0	-52.5224	-59.9999
14	62.3673	-3.0	62.3663	-2.9999
15	-31.6047	-11.0	-31.6176	-10.9998
16	-159.7724	-3.0	-159.7887	-2.9999
17	-206.3462	11.0	-206.3347	10.9998
18	-148.2071	-6.0	-148.1835	-59.9999
19	-69.4109	6.0	-69.4119	59.9999
20	0.0342	5.5	0.034	5.4999
21	52.5252	-8.5	52.5224	-8.4998
22	62.3673	-1.5	62.3663	-1.5000
23	31.6047	1.5	31.6176	1.5000
24	-575.8640	9.5	-575.8681	9.4998

25	144.8897	-9.5	144.8672	-9.4998
26	318.4486	504.5	318.3693	504.4903
27	43.9587	-0.5	43.9542	-0.5000
28	-2.4150	506.0	-2.4152	505.9903
29	-33.4722	-2.0	-33.4677	-2.0000
30	-135.5951	500.0	-135.5788	499.9904
31	-29.7322	-7.0	-135.5788	-6.9999
32	-575.8640	500.0	-575.8681	499.9904
33	-144.8897	7.0	-144.8672	6.9999
34	318.4486	506.0	318.3693	505.9903
35	-43.9587	2.0	-43.9542	2.0000
36	-2.4150	504.5	-2.4152	504.4903
37	33.4722	0.5	33.4677	0.5000
38	-135.5951	9.5	-135.5788	9.4998
39	29.7322	9.5	29.7384	9.4998

Dapat dilihat dari tabel di atas hasil dari perhitungan MATLAB dengan MODELSIM dan FPGA terdapat perbedaan, itu disebabkan karena tingkat ketelitian yang berbeda.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berikut ini beberapa kesimpulan yang didapatkan penulis dari hasil sintesis dan implementasi sistem IFFT 64 titik radiks-4.

1. Besar *logic element* yang terpakai pada sistem DCT dengan $N=8$ adalah 615.919 sedangkan *logic element* yang tersedia pada FPGA sebanyak 18.752. Jadi sistem DCT dengan nilai $N=8$ tidak bisa disintesa ke dalam FPGA. Sedangkan total *logic element* yang dibutuhkan untuk $N DCT = 2$ lebih kecil daripada $N DCT = 8$, yaitu sebanyak 18.291 atau 98% dari total *logic element* yang tersedia pada FPGA.
2. Delay yang dibutuhkan untuk menampilkan hasil sistem DCT untuk $N=8$ adalah 8 clock atau 160 ns karena 1 clock pada FPGA sebesar 20 ns. Sedangkan pada DCT $N=2$ adalah 4 clock atau 80 ns. Delay tersebut dibutuhkan karena FPGA menghitung dulu sistem yang sudah dikoding barulah ditampilkan.
3. Hasil keluaran dari FPGA dibandingkan dengan hasil keluaran dari MATLAB dan menghasilkan BER sebesar 0.2930 untuk DCT $N=2$ dan 0.0508 untuk DCT $N=8$.
4. Hasil keluaran MATLAB dan hasil simulasi dengan Quartus yang diimplementasikan pada FPGA terdapat perbedaan karena tingkat ketelitian yang terbatas pada FPGA, jika tingkat ketelitian ditingkatkan maka akan membutuhkan *resource memory* yang lebih besar.
5. Sistem DCT $N=2$ dapat diimplementasikan pada board FPGA Altera Cyclone II EP2C20F484C7.

5.2 Saran

1. Untuk penelitian selanjutnya bisa mencoba dengan menggunakan *board* FPGA dengan spek yang lebih tinggi dengan harapan tingkat ketelitian yang lebih tinggi.
2. Untuk pengembangannya dapat dicoba dengan memperbesar nilai N pada DCT menjadi 16 atau lebih dengan harapan proses kompresi citra akan lebih cepat karena akan mereduksi banyaknya pengulangan yang terjadi.

Referensi

- [1] Madenda, Sarifuddin. 2015. Pengolahan Citra & Video Digital. Jakarta: Erlangga.
- [2] Manalu, Manoto J F. 2011. *Perancangan dan Implementasi Prosesor I/FFT 512 titik Radix-8 pada FPGA*. Tugas Akhir pada Institut Teknologi Telkom : Tidak Diterbitkan.
- [3] Monangin, Yan Felix., 2014, Implementasi Discrete Cosine Transform Pada Field Programmable Gate Array. Universitas Brawijaya.
- [4] Scrofano, Ronald, 2003, Energy-Efficient Discrete Cosine Transform on FPGAs. Southern California University.
- [5] *Cyclone II FPGA Starter Development Board Reference Manual*. Altera. [Online] 2006. Available : <http://www.altera.com/>

- [6] Khairuddin, Labib Ahmad., 2011, Perancangan dan Implementasi Prosesor FFT 256 titik-OFDM Baseband berbasis pengkodean VHDL pada FPGA . IT Telkom.
- [7] S. Mazor dan P. Langstraat, A Guide To VHDL, New York: Springer Science+Business Media, 1992.
- [8] Susandi, Rahadian. 2017. *Implementasi FFT 64 Titik menggunakan Radiks-4 pada FPGA*. Tugas Akhir pada Universitas Telkom : Tidak Diterbitkan.
- [9] Basri, I.Y. “Implementasi DCT pada Field Programmable Gate Array”, Thesis S2 Jurusan Teknik Elektro. UGM : Yogyakarta, 2010.
- [10] Anas, Khairul. 2017. *Implementasi IFFT 64 Titik menggunakan Radiks-4 pada FPGA*. Tugas Akhir pada Universitas Telkom : Tidak Diterbitkan.

