

Implementasi Transformasi Wavelet Diskrit Sebagai Pemroses Awal Kompresi Data Sinyal Satu Dimensi Menggunakan FPGA

Ali Husein Alasiry, Hary Oktavianto¹⁾, Bambang Sumantri¹⁾,
Zulfatur Rofiah, Gilang Unthari¹⁾

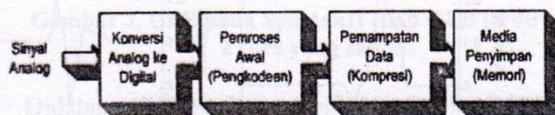
Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya – ITS
Kampus ITS, Keputih-Sukolilo, Surabaya-60111
email : ali@eepis-its.edu

Abstrak - Paper ini merupakan bagian dari penelitian tentang penerapan transformasi wavelet diskrit (DWT) dalam proses kompresi dan dekomposisi data secara hardware menggunakan FPGA. Pada paper ini dijelaskan penerapan DWT sebagai filter dalam pemrosesan awal data sinyal satu dimensi sebelum dilakukannya proses kompresi. Dari bagian penelitian ini diketahui efisiensi dan sumber daya FPGA yang dibutuhkan dalam penerapan blok pemroses awal DWT untuk kompresi dengan tiga tingkatan analisis pada FPGA.

Kata kunci: kompresi, dekomposisi, wavelet, DWT, VHDL, FPGA, SpartanII

1. PENDAHULUAN

Kompresi dan dekomposisi data seringkali dibutuhkan dalam penyimpanan ataupun pengiriman data untuk mengatasi keterbatasan kapasitas memori maupun bandwidth komunikasi dan juga untuk mempercepat prosesnya. Permasalahan yang timbul selanjutnya adalah bagaimana mendapatkan nilai kompresi yang semaksimal mungkin dengan seminimal mungkin informasi hilang pada saat dekomposisi. Untuk meningkatkan efisiensi dan kinerja dari proses kompresi diantaranya adalah dengan melakukan pemrosesan awal berupa pengkodean ataupun transformasi sebelum proses kompresi dilakukan.

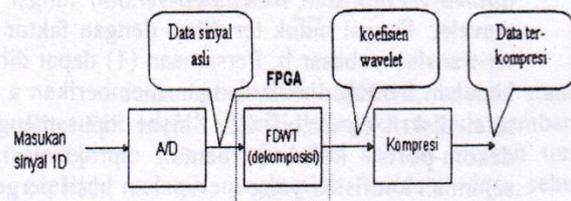


Gambar 1. Proses kompresi dengan pengkodean

Teknik pengkodean yang telah dikenal antara lain dengan pengkodean simbol, pengkodean *fixed length* atau pengkodean Huffman. Sedangkan cara transformasi diantaranya dengan DCT. Dalam penelitian ini digunakan transformasi dengan DWT.

Untuk merealisasikan DWT dapat dilakukan secara perangkat keras ataupun lunak. Untuk realisasi menggunakan perangkat lunak atau program perlu dipertimbangkan kecepatan CPU memproses data. Bila CPU tidak cukup cepat dalam proses kompresi-

nya akan menyebabkan turunnya performa system secara waktu nyata.



Gambar 2. DWT sebagai pemroses awal

Perkembangan teknologi VLSI khususnya FPGA yang sangat cepat telah memberikan kemungkinan yang lebih besar untuk penerapan pengolahan sinyal digital berkecepatan tinggi dalam bentuk hardware dengan cara yang biaya yang cukup murah. Realisasi dengan perangkat keras dipilih dengan alasan ini.

2. TINJAUAN TEORI

2.1. Kompresi Data Sinyal Satu Dimensi

Sinyal satu dimensi (1-D) merupakan sinyal dengan jumlah variabel satu yang berubah dalam skala waktu. Sinyal ini dapat bersifat periodik, misalnya gelombang kosinus, atau aperiodik, misalnya sinyal random atau musik. Sinyal juga dapat mengandung satu atau lebih dari satu komponen frekuensi.

Kompresi data berarti pengecilan jumlah data dalam merepresentasikan hal yang sama. Secara umum ada dua metode kompresi data yaitu *lossless compression* dan *lossy compression*. *Lossless compression* merupakan metoda kompresi yang mempertahankan keutuhan informasi yang dikandung oleh data, sehingga proses rekonstruksi yang eksak dapat dilakukan sehingga diperoleh data yang sama persis dengan data aslinya. Biasanya dipakai pada kompresi file dan komunikasi data. Sedangkan pada *lossy compression*, ada informasi yang hilang pada proses kompresi data sehingga rekonstruksi yang dilakukan tidak sempurna. Dalam batas-batas tertentu ketidaksempurnaan tersebut bisa ditolelir selama informasi-informasi utama tidak hilang, misalnya pada file gambar atau musik.

2.2. Transformasi Wavelet Diskrit (DWT)

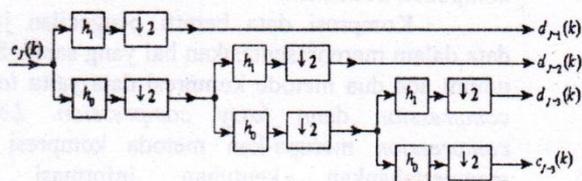
Secara umum transformasi wavelet kontinu untuk sinyal $f(x)$ berdimensi satu didefinisikan pada persamaan (1) [6]:

$$W_a f(b) = \int f(x) \psi_{a,b}(x) dx \quad (1)$$

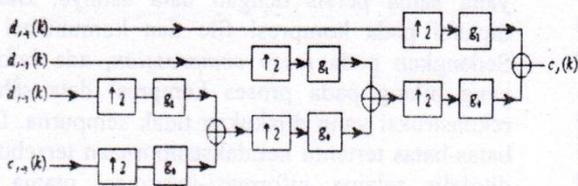
Fungsi Ψ disebut dengan fungsi induk wavelet yang mampu melokalisasi sinyal $f(x)$. Transformasi wavelet men-dekomposisi sinyal $f(x)$ kedalam bentuk varian sinyal induk wavelet yang ter-dilasi dan ter-translasi. Dengan kata lain sinyal $f(x)$ direpresntasikan sebagai jumlah dari kumpulan *dilated-version* dan *translated-version* fungsi induk wavelet. Fungsi induk ter-dilasi dengan faktor a dan ter-translasi sebesar b . Persamaan (1) dapat dibentuk kedalam bentuk diskrit dengan memberikan a dan b nilai diskrit ($a=2^n, b \in Z$). Sinyal input mengalami dekomposisi kedalam bentuk aproksimasi dari sejumlah koefisien yang merupakan hasil pergeseran dan penskalaan dari suatu bentuk fungsi induk wavelet seperti pada persamaan (2).

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right) \quad (2)$$

DWT dapat dihitung baik secara maju (FDWT) maupun terbalik (IDWT) dengan menggunakan piramida Mallat [8] pada gambar 3 dan 4. Proses dekomposisi sinyal menjadi koefisien-koefisien wavelet dapat dilakukan dengan menggunakan transformasi maju (forward DWT, FDWT) seperti pada gambar 2 dimana h_0 dan h_1 koefisien dilasi berkaitan dengan penskalaan dan fungsi wavelet. Sedangkan rekonstruksi ulang sinyal dapat dilakukan dengan menggunakan transformasi terbaliknya (inverse DWT, IDWT) seperti pada gambar 3, dimana g_0 dan g_1 adalah koefisien high-pass dan low-pass filter berhubungan dengan fungsi induk wavelet.



Gambar 3. Pohon Mallat proses analisis



Gambar 4. Pohon Mallat proses sintesis

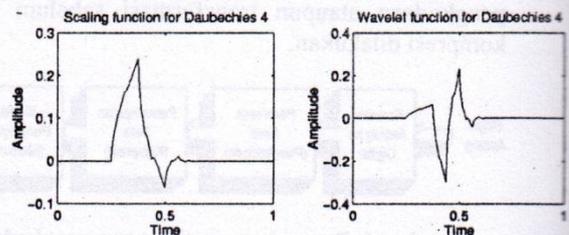
Dengan memasukkan persamaan dasar filter digital sesuai gambar 3 dan gambar 4 diatas selanjutnya diturunkan persamaan untuk DWT maju (FDWT) dan DWT terbalik (IDWT) seperti dinyatakan dalam kumpulan persamaan (3).

$$\text{Forward DWT: } \begin{cases} c_j(k) = x(k) \\ c_{j-1}(k) = \sum h_n(m-2k)c_j(m) \\ d_{j-1}(k) = \sum h_n(m-2k)c_j(m) \\ \text{for } j = J, J-1, \dots, J_0+1 \\ \{d_{j-1}(k), d_{j-2}(k), \dots, d_{j_0+1}(k), d_{j_0}(k), c_{j_0}(k)\} \end{cases} \quad (3)$$

$$\text{Inverse DWT: } \begin{cases} c_{j+1}(k) = \sum c_j(m)g_n(k-2m) + \sum d_j(k)g_1(k-2m) \\ \text{for } j = J_0, J_0+1, \dots, J-1 \\ x(k) = c_j(k) \end{cases}$$

2.3. Filter Wavelet

Masalah utama dalam penggunaan transformasi wavelet adalah memilih filter wavelet yang cocok. Filter wavelet Daubechies, dikembangkan oleh Ingrid Daubechies, mungkin merupakan filter wavelet yang paling populer karena memiliki karakteristik yang diperlukan. Filter wavelet Daubechies mewakili dasar dari pengolahan sinyal wavelet dan digunakan pada banyak aplikasi. Filter wavelet Daubechies kadang juga disebut sebagai filter Maxflat karena mempunyai respon frekuensi puncak yang datar pada frekuensi 0 dan π . Wavelet Daubechies mempunyai sifat memiliki nilai maksimum untuk berkurang hingga titik yang hilang sehingga sesuai untuk aplikasi kompresi. Filter wavelet Daubechies juga dapat dibuat menjadi filter orthogonal dan dapat direkonstruksi kembali dengan sempurna. Dan juga, filter Maxflat mempunyai koefisien dyadic. Oleh karena itu, filter Maxflat dapat dengan mudah diwakili oleh bilangan biner untuk keperluan implementasi secara hardware.

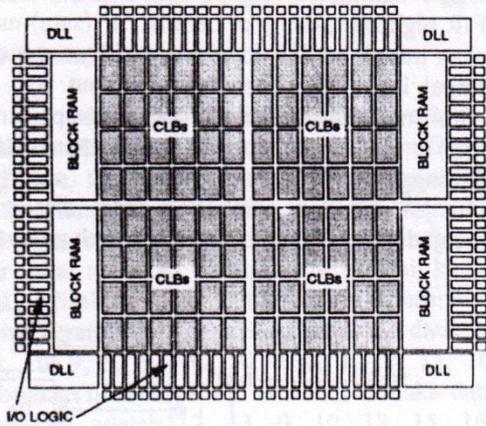


Gambar 5. Fungsi skala dan wavelet Daubechies-4

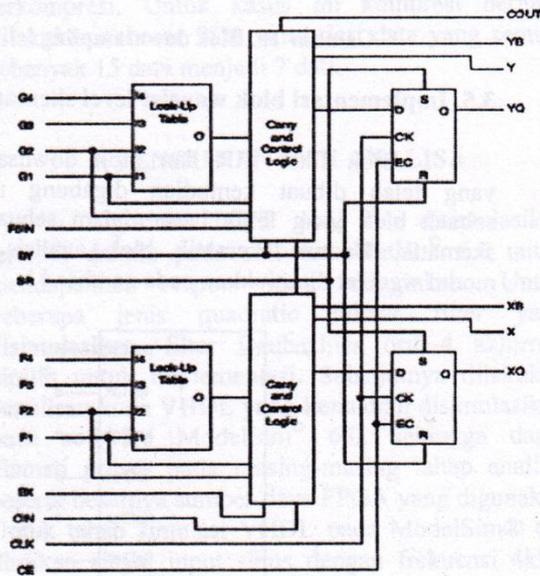
2.4. FPGA seri SpartanII

Bank filter pada wavelet diimplementasikan pada divais SpartanII XC2S100 produksi Xilinx. Divais FPGA tersebut menggunakan teknologi berbasis SRAM sehingga bila catu daya dihilangkan maka program yang telah dimasukkan akan hilang. Blok diagram dari SpartanII ditunjukkan oleh gambar 6.

SpartanII terdiri atas logic array yang disebut CLB berfungsi untuk mengimplementasikan disain yang dibuat. Pada divais XC2S100 terdapat 600 CLB. Tiap CLB terdiri atas dua buah slice seperti yang ditunjukkan oleh gambar 7.



Gambar 6. Blok diagram divais SpartanII



Gambar 7. CLB pada SpartanII (tiap CLB terdiri atas 2 slice yang sama)

Didalam slice terdapat Look-Up Table (LUT) dan Flip-flop. Disain logika yang dibuat akan diubah menjadi tabel persamaan logika yang disimpan didalam LUT. Sedangkan flip-flop berfungsi untuk mengimplementasikan disain logika sekuensial. SpartanII juga dilengkapi dengan BlockRAM sebesar 40 kilo-bit. Empat buah Digital Locked Loop (DLL) disediakan untuk menghasilkan clock dengan delay rendah yang mempunyai jalur khusus ke semua bagian. Input/Output Block (IOB) berada pada tiap tepi divais yang terhubung langsung dengan pin IC FPGA.

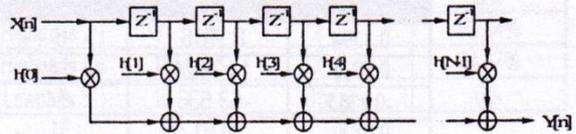
3. Desain Implementasi

3.1. Implementasi filter FIR

Blok dasar FDWT filter bank terdiri dari desimator berupa FIR yang diikuti dengan operator down sampling [10]. Persamaan DWT dapat dihitung secara efisien menggunakan *quadratic mirror filter* (QMF) seperti ditunjukkan pada gambar diatas. Kesemua struktur piramid FDWT dan IDWT dapat disusun menggunakan filter FIR. Secara umum filter FIR dengan panjang M dinyatakan dalam persamaan (4).

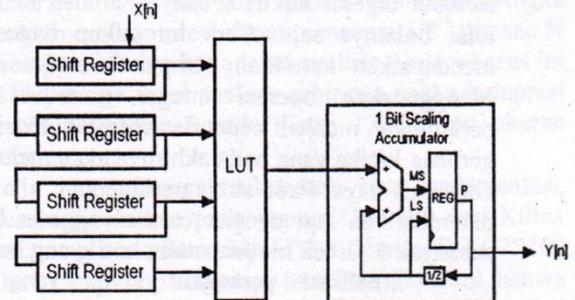
$$H(z) = \sum_{k=0}^{M-1} h[k] \cdot z^{-k} \quad (4)$$

Pada kebanyakan aplikasi, bentuk standar filter FIR adalah seperti ditunjukkan pada gambar 8. Setiap tap filter terdiri dari sebuah elemen tunda (delay), sebuah penjumlah (adder) dan sebuah pengali (multiplier) [7].



Gambar 8. Struktur filter FIR langsung

Implementasinya pada hardware secara blok diagram seperti ditunjukkan pada gambar 9. FIR terdiri dari empat shift register yang dikaskade, sebuah tabel cari lihat (*look up table, LUT*) dan sebuah akumulator.



Gambar 9. Komponen blok FIR pada FPGA

LUT berisi semua kemungkinan yang terdapat pada table koefisien wavelet Daubechies. Wavelet Daubechies dipilih karena diketahui memiliki keunggulan dalam hal kompresi data [9].

3.2. Implementasi bank filter Daubechies orde-4

Penelitian ini mengimplementasikan bank filter orthogonal Daubechies 4 (db4) dimana koefisien-koefisien filter tersebut sebenarnya dinyatakan dalam bilangan real. Untuk implementasi secara hardware, koefisien-koefisien tersebut harus

dikuantisasi kedalam bilangan biner yang mungkin akan menyebabkan berkurangnya tingkat ketelitian atau akurasi. Akurasi diukur sebagai perbandingan sinyal terhadap noise dari sinyal yang telah direkonstruksi. Sehingga proses kuantisasi harus dilakukan dengan tepat agar kualitas dari transformasi wavelet tetap terjaga.

Filter Daubechies telah digunakan secara luas karena mempunyai sifat orthogonal dan mudah dibuat karena merupakan filter yang pendek. Juga mempunyai kondisi hasil rekonstruksi yang sempurna dan memuaskan. Koefisien-koefisien filter ditunjukkan oleh tabel 1 dan 2.

Tabel 1. Koefisien-koefisien filter LPF analisis

Koefisien	H_1	Pembulatan
0.4830	61.824	62
0.8365	107.072	107
0.2241	28.6848	29
-0.1294	-16.5632	-17

Tabel 2. Koefisien-koefisien filter HPF analisis

Koefisien	H_2	Pembulatan
0.1294	8.2816	8
0.2241	14.3424	14
-0.8365	-53.536	-54
0.4830	30.912	31

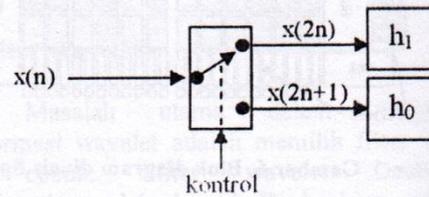
Implementasi perhitungan floating-point untuk proses DWT membutuhkan banyak gerbang logika sehingga mahal. Kerugian lainnya adalah bahwa operasi floating-point lebih lambat daripada operasi integer. Oleh karena itu, untuk implementasi ke hardware dengan mudah, koefisien pecahan tersebut digeser kekiri sebanyak 2^7 dan mengambil nilai bulatnya saja. Cara ini cukup bagus untuk mendapatkan ketelitian sampai 3 angka desimal. Menggunakan operasi integer membuat proses perhitungan menjadi cepat dan membutuhkan sedikit gerbang logika yang pada akhirnya akan menurunkan konsumsi daya. Pada akhir perhitungan, nilai output dikembalikan lagi dengan cara menggeser kekanan sebanyak 2^7 untuk mendapatkan hasil yang benar.

Realisasi perangkat keras yang paling mungkin dan ekonomis adalah menggunakan FPGA. Disini perlu dipertimbangkan kapasitas keping FPGA yang digunakan, disamping kapasitas memori RAM. IC seri XC4000 seperti XC4010XL mempunyai kapasitas sebesar 10.000 *logic gates*. Sedangkan IC seri SpartanII seperti XC2S100 mempunyai kapasitas 100.000 *logic gates* dan mempunyai blok-RAM internal sebanyak 40Kb. Karena SpartanII mempunyai RAM internal, diharapkan disain hardware dapat menjadi lebih sederhana.

3.4. Implementasi blok downsampling

Downsampling merupakan suatu proses untuk mendapatkan urutan data genap atau ganjil saja untuk kemudian di filter. Biasanya proses

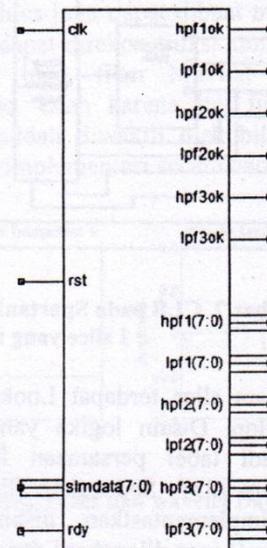
downsampling menggunakan modul decimator, tetapi karena keterbatasan slices FPGA dan untuk efisiensi program maka proses down-sampling disini menggunakan toggle switch yang dimanfaatkan sebagai multiplexer, dimana jika sinyal toggle bernilai '0' (nol) maka data akan masuk ke FIR highpass, dan jika bernilai '1' (satu) maka data akan masuk ke FIR lowpass. $X(n)$ yang merupakan sinyal input akan mempunyai output $y(n) = x(2n)$ untuk menghasilkan urutan data genap, dan $y(n) = x(2n + 1)$ untuk menghasilkan urutan data ganjil. Sehingga proses downsampling seperti yang ditunjukkan oleh gambar 3 dapat disederhanakan menjadi seperti yang ditunjukkan oleh gambar 10.



Gambar 10. Blok downsampling

3.5. Implementasi blok wavelet level skematik

Blok filter FIR dan blok downsampling yang telah dibuat kemudian digabung menjadi sebuah blok yang lebih besar dalam sebuah blok skematik. Bentuk skematik blok analisis dalam modul wavelet ditunjukkan pada gambar 11.



Gambar 11. Komponen wavelet level skematik

Gambar 11 merupakan *top-level* dari gambar 3. Pada bagian input terdapat jalur 'clk' sebagai clock untuk menjalankan semua fungsi logika, jalur 'rst' sebagai sinyal reset untuk mengembalikan ke kondisi awal, jalur 'simdata' dengan lebar 8-bit sebagai masukan data dari bagian ADC yang akan diolah dan jalur 'rdy' sebagai sinyal penanda bahwa data dari

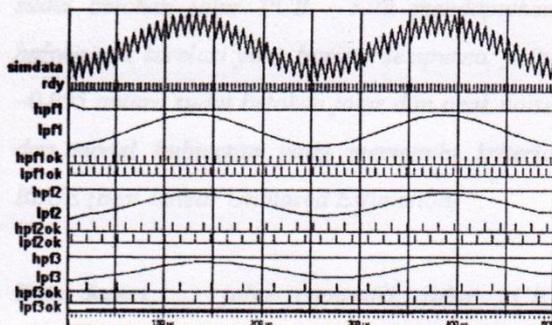
ADC adalah *valid*. Sedangkan pada bagian output terdapat hasil keluaran dari masing masing filter sebanyak 6 buah dengan indeks hpf_n dan lpf_n beserta sinyal untuk sinkronisasi dengan indeks $hpf_{n,ok}$ dan $lpf_{n,ok}$. Sinyal-*ok* ini akan aktif menandakan bahwa proses didalam filter yang bersangkutan telah selesai.

3.6. Modul Kompresi

Modul kompresi yang dibuat yaitu memberi nilai *threshold* untuk data-data hasil pemfilteran FIR *highpass*. Data yang berada dibawah nilai *threshold* akan dianggap nol. Nilai nol ini akan dihitung seberapa banyak jumlahnya, dan kemudian jumlah dari nilai nol akan dituliskan di sebelah kanan nilai nol. Misalkan data output yang dihasilkan oleh pemfilteran FIR *Highpass* sebanyak 15 data, yaitu 12, 13, 10, 9, 5, 6, 7, 3, 4, 0, 0, 1, 13, 15, 16 dan diberikan nilai *threshold* adalah 11, maka data hasil kompresi adalah 12, 13, 0, 10, 13, 15, 16 yaitu sebanyak 7 data. Nilai 0 merupakan nilai hasil kompresi, dan 10 menunjukkan jumlah data yang terkompresi. Untuk kasus ini kompresi berhasil dilakukan sebesar 55% yaitu dari data yang semula sebanyak 15 data menjadi 7 data.

4. PENGUJIAN DAN ANALISA

Algoritma DWT diuji dan disimulasikan terlebih dahulu pada software Matlab[®] 6.0 untuk mendapatkan respon yang diinginkan. Untuk beberapa jenis *quadratic mirror filter* yang disimulasikan, filter *Daubechies orde-4* akhirnya dipilih untuk implementasi. Selanjutnya dilakukan penulisan kode VHDL yang kemudian disimulasikan pada software ModelSim[®] 6.0, sehingga dapat diamati proses pada masing-masing tahap analisis beserta besarnya sumber daya FPGA yang digunakan. Untuk tahap simulasi VHDL pada ModelSim[®] 6.0 diujikan sinyal input sinus dengan frekuensi 4kHz yang dijumlahkan dengan sinyal sinus frekuensi 125kHz. Sedangkan frekuensi *sampling* yang digunakan adalah 200kHz. Frekuensi *cut off* FIR *highpass* dan *lowpass* yang diperoleh menggunakan koefisien *wavelet induk Daubechies 4* adalah 6 KHz. Gambar 12 menunjukkan hasil simulasi.



Gambar 12. Hasil simulasi VHDL

Pada gambar 12 terlihat bahwa modul analisis pada gambar 11 diuji menggunakan sebuah sinyal sinusoida dengan gangguan. Setelah melewati filter maka terlihat bahwa pada keluaran filter *high-pass* hampir datar. Artinya apabila kompresi dikerjakan pada keluaran filter *high-pass* akan didapatkan faktor kompresi yang cukup besar. Garis-garis seperti jarum menunjukkan bahwa data genap dan data ganjil masing-masing diproses pada filter *high-pass* dan filter *low-pass*. Garis-garis seperti jarum tersebut juga menunjukkan bahwa data telah mengalami proses *downsampling*.

Besarnya angka kompresi akhir dipengaruhi oleh nilai kompresi pada masing-masing level. Dalam pengujian dengan sinyal uji diatas, keluaran filter *low-pass* setelah dikuantisasi tidak mengalami kompresi sehingga kompresi akhir ditentukan oleh hasil kuantisasi dari masing-masing keluaran *high-pass* filter, seperti ditunjukkan pada tabel 3.

Tabel 3. Kompresi pada level-level *highpass* filter

FIR	Data Awal	Data Hasil Kompresi	Jumlah kompresi
Highpass 1	62	62	0%
Highpass 2	31	2	93,54%
Highpass 3	15	2	84,67%
Lowpass 3	15	15	0%

Data tersebut diperoleh pada proses kompresi dengan nilai batas (*threshold*) 2 dan -2. Secara keseluruhan total kompresi yang didapat untuk sinyal uji yang diberikan adalah 65,32% atau 34,68% dari jumlah data yang sebenarnya.

Secara teori angka kompresi tersebut tidak tetap dan dipengaruhi oleh bentuk sinyal, ketidak linieran sinyal, dan juga proses kuantisasi. Angka maksimum kompresi dapat mencapai $1/2^N$ dimana N merupakan banyaknya level analisis. Dalam kasus ini terdapat tiga level analisis sehingga angka kompresi dapat dipastikan berada diatas 12,5% dari ukuran sebenarnya.

Untuk memastikan kelayakan implementasi, desain dimuatkan pada papan XSA-100 dari Xilinx Inc, yang berintikan chip FPGA Spartan II XC2S100. Dari tahap implementasi tersebut diketahui bahwa jumlah *slice* dari chip FPGA XC2S100 yang terpakai sebesar 74%.

5. KESIMPULAN

Berdasarkan hasil pengujian dan analisa yang telah di lakukan dapat diambil beberapa kesimpulan. Dengan bantuan simulasi pada software ModelSim 6.0 dapat diketahui total kompresi untuk sinyal uji yang diberikan adalah 65,32% yang didapat dengan nilai batas (*threshold*) 2 dan -2.

Jumlah total *slice* atau CLB yang digunakan dalam implementasi DWT pada FPGA Spartan II XC2S100 sebesar 74%.

Kualitas hasil kompresi yang sebenarnya baru akan diketahui pada tahapan dekompresi

dimana rekonstruksi sinyal haruslah dapat mengembalikan detail informasi penting dari sinyal. Untuk dekompresi sedang dalam tahap pengujian dan diharapkan akan segera diketahui hasilnya dalam waktu dekat.

[14] T. I. Laakso and V. Valimaki, "Splitting the Unit Delay", *IEEE Signal Processing Magazine*, pp. 30-60, Jan. 1996.

DAFTAR REFERENSI

- [1] Rioul, O., and Vetterli, M., "Wavelets and Signal Processing", *IEEE Signal Processing Magazine*, Vol. 8, No. 4, pp. 14-38, October 1991.
- [2] Mallat, S. G., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Recognition and Machine Intelligence*. Vol. 11, No. 7, pp. 674-693, July 1989.
- [3] Daubechies, I., "The Wavelet Transform, Time-Frequency Localization and Signal Analysis", *IEEE Transactions on Information Theory*. Vol. 36, No. 5, pp. 961-1005, September 1990.
- [4] S.A. White, "Applications of Distributed Arithmetic to Digital Signal Processing", *IEEE ASSP Magazine*, Vol. 6(3), pp. 4-19, July 1989.
- [5] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [6] Xilinx Inc., *The Programmable Logic Data Book*, 1999.
- [7] C. H. Dick and f. j. harris, "FPGA Multirate Filters: A Case Study Using Virtex", *ICSPAT'99*, Orlando, Florida, Oct. 1999.
- [8] C. H. Dick and f. j. harris, "High-Performance FPGA Filters Using Sigma-Delta Modulation Encoding", *The International Conferences on Acoustics Speech and Signal Processing - (ICASSP'99)*, Phoenix Arizona, March 15-19 1999.
- [9] E. B. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation", *IEEE. Trans. Acoust., Speech Signal Processing*, Vol. 29, No. 2, pp. 155-162, April 1981.
- [10] Brigham, E. O., *The Fast Fourier Transform*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.
- [11] Allen, J. B. and Rabiner, L. R., "A Unified Approach to Short-Time Fourier Analysis and Synthesis", *Proceedings of IEEE*. Vol. 65, No. 11, pp. 1558-1564, 1977
- [12] Burrus, C. S.; Gopinath, R. A.; and Guo, H., *Introduction to Wavelets and Wavelet Transforms: A Primer*, Prentice Hall Inc., Upper Saddle River, New Jersey, 1998.
- [13] Strang, G. and Nguyen, T., *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesley, Massachusetts, 1996.