

Merancang dan Mensimulasi *Infinite Impulse Response Chebyshev Low-Pass Digital Filter* Menggunakan Perangkat FPGA

Mariza Wijayanti^{1a}

¹*Teknik Elektro, Fakultas Teknologi Industri, Universitas Gunadarma, Jakarta, Indonesia
Jl. Margonda Raya No. 100, Pondok Cina, Depok 1642*

^a*mariza_w@staff.gunadarma.ac.id*

Abstraksi

Pada artikel ini, menyajikan desain filter. Langkah pertama adalah menentukan spesifikasi filter, melakukan perhitungan matematis low pass, high pass dan band-pass filter IIR, kemudian mensimulasi hasil perhitungan desain menggunakan MATLAB dan akhirnya desain diimplementasikan menggunakan FPGA. Direct-form dianggap pendekatan dalam mewujudkan filter digital. IIR filter diimplementasikan di Spartan-IIIxc3s500c-4fg320 FPGA dan disimulasikan dengan bantuan Xilinx ISE 9.2i. Perangkat lunak ini digunakan untuk sintesis dan simulasi kode. Kode untuk langsung bentuk tetap titik filter IIR yang telah direalisasikan.. Modul yang digunakan seperti multiplier, adder, dan delay. Hasil simulasi data kemudian dibandingkan dengan output program Matlab. Berdasarkan hasil pengujian dari rangkaian perbandingan filter, kesalahan terjadi karena komponen penggunaan Divider yang membuat proses pembagian dengan hasil pembulatan.

Kata kunci: *Filter, IIR, FPGA, Xilinx-Ise 9.2i, Matlab*

Designing and Simulating Infinite Impulse Response Chebyshev Low-Pass Digital Filter Using FPGA Device

Abstract

This study will present filter design. The first step is determining the filter specification, next, calculating the low pass, high pass, and band-pass of IIR filter, then simulating the design calculation result using MATLAB, and last, implementing the design using FPGA. Direct form is considered to be the approach in realizing digital filter. IIR filter is implemented in Spartan-IIIxc3s500c-4fg320 FPGA and simulated by assisted with Xilinx ISE 9.2i. The software is used in code synthesis and simulation. The code of IIR filter fixed focus form has been realized. The modules used are multiplier, adder, and delay. The result of data simulation then is compared with Matlab program output. Based on the result of filter comparisons, the error occurs because of the component of Divider use which rounding the result of dividing process.

Keywords: *Filter, IIR, FPGA, Xilinx-Ise 9.2i, Matlab*

PENDAHULUAN

Dalam pemrosesan sinyal, fungsi filter adalah untuk menghapus bagian yang tidak diinginkan dari sinyal, seperti random noise,

atau untuk mengambil bagian yang berguna dari sinyal, seperti komponen yang berada di dalam rentang frekuensi tertentu [Ruggiero, 1982].



Gambar. 1. Sebuah diagram blok dasar filter. [Ruggiero, 1982]

Ada dua jenis filter yaitu analog dan digital. IIR Filter adalah jenis filter digital, yang dapat digunakan untuk melakukan semua jenis pe-nyaringan yaitu high-pass, low-pass, band-pass dan band-reject [Freeuk, 2011]. Ada dua tipe dasar filter digital, Finite Impulse Response (FIR) dan Infinite Impulse Response (IIR) filter. Desain filter FIR digital sangat tergantung pada rekan-rekan analog mereka karena ada banyak sumber daya yang bekerja dan metode desain straight forward analog desain umpan balik filter sementara hampir tidak ada untuk filter IIR digital. Akibatnya, biasanya, ketika sebuah IIR filter digital akan diimplementasikan, filter analog (misalnya Chebyshev filter, filter Butterworth, Elliptic filter) pertama kali merancang dan kemudian diubah menjadi filter digital dengan menerapkan teknik diskretisasi seperti mengubah Bilinear atau invarian Impulse.

FPGA ada pada tahun 1984 sebagai alternatif untuk perangkat Programmable logika (PLDs) dan Asics. FPGA menawarkan manfaat signifikan mudah diprogram, memberikan desainer kesempatan ganda untuk tweak sirkuit. FPGA terdiri dari sebuah array dari blok logika yang dikonfigurasi dengan menggunakan pe-rangkat lunak. Programmable blok input/output mengelilingi blok-blok logika. Keduanya dihubungkan oleh interkoneksi diprogram. FPGA menawarkan jutaan gerbang kapasitas logika, beroperasi pada 300 MHz, dan

menawarkan fungsi yang terintegrasi seperti prosesor dan memori. FPGA menawarkan semua fitur yang dibutuhkan untuk menerapkan desain yang paling kompleks. Manajemen Jam difasilitasi oleh on-chip PLL (phase-locked loop) atau DLL (delay-locked loop) sirkuit. Blok memori khusus dapat dikonfigurasi sebagai dasar satu-port RAM, ROM, FIFOs, atau CAM. Sekarang FPGAs adalah sumber daya sistem bangunan berkecepatan tinggi serial input / output, aritma-tika modul, prosesor tertanam, dan sejumlah besar memori.

Sirkuit elektronik yang dibangun melalui beberapa tahap seperti desain, simulasi, perakitan, dan pengujian. Desain dilakukan dengan mengakui kebutuhan dan kegunaan dari sirkuit. Simulasi pada saat desain yang diperlukan untuk meminimalkan tingkat kesa-lahan ketika sirkuit dilakukan dalam perakitan, sehingga dapat mengurangi biaya kegagalan sirkuit elektronik yang dirancang dan dibangun. Sekarang masalahnya adalah Bagaimana merancang sebuah filter IIR Chebyshev digital menggunakan trans-formasi bilinear dan Bagaimana Filter ini dirancang dan disimulasikan menggunakan Xilinx ISE 9.2i.

TINJAUAN PUSTAKA

IIR memiliki respon Impulse, $h(n)$, dengan jumlah tak terbatas sampel.

Persamaan IIR Filter input-output:

$$y[n] = a_0x[n] + a_1x[n-1] + \dots + a_{M-1}x[n-(M-1)] - b_1y[n-1] - b_2y[n-2] - \dots - b_Ny[n-(N)] \quad (1)$$

Filter Chebyshev

Filter Chebyshev adalah analog atau digital filter memiliki lebih tajam roll-off dan lebih ripple passband (tipe I) atau stopband ripple (tipe II) daripada filter Butterworth. Filter Chebyshev memiliki properti untuk meminimal-kan kesalahan antara karakteristik Filter ideal dan aktual selama rentang filter, tetapi dengan ripple di passband. Jenis filter ini dinamai untuk

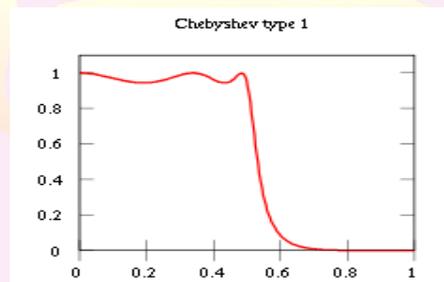
menghormati Pafnuty Chebyshev karena karakteristik matematika mereka berasal dari Chebyshev polinomial.

Karena passband ripple melekat dalam filter Chebyshev, filter yang memiliki respon halus di passband tapi respon yang lebih teratur di stopband lebih disukai untuk beberapa aplikasi [Williams, 2006].

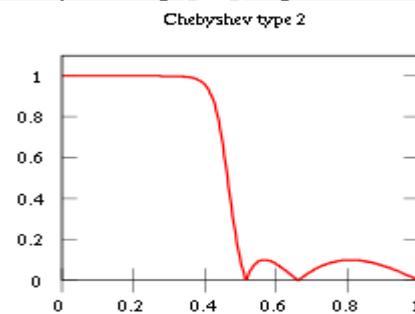
Untuk menemukan urutan Chebyshev, digunakan persamaan berikut:

$$n = \frac{\log \left[\frac{10^{\alpha_{min}/10-1}}{10^{\alpha_{max}/10-1}} \right]}{2 \log \left(\frac{f_{stop}}{f_p} \right)} \quad (2)$$

$$\epsilon = \sqrt{10^{\alpha_{max}/10} - 1} \quad (3)$$



Gambar 2. Chebyshev tipe 1 Lowpass Filter [Ifeachor, 2002]



Gambar 3. Chebyshev tipe 2 Lowpass Filter [Ifeachor, 2002]

Bentuk Direct Form

Direct form dapat dikutip dalam persamaan matematis di bawah ini:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{(a_0 + a_1z^{-1} + a_2z^{-2} + \dots + a_kz^{-k})}{(1 + b_1z^{-1} + b_2z^{-2} + \dots + b_kz^{-k})}$$

$$\Leftrightarrow H(z) = \frac{\sum_{i=0}^k a_i z^{-i}}{1 + \sum_{j=0}^k a_j z^{-j}} \quad (4)$$

Direct Form I

Pendekatan langsung untuk realisasi IIR filter Direct Form I, di mana persamaan perbedaan dievaluasi secara langsung. Form ini praktis untuk filter kecil, tapi mungkin tidak efisien dan tidak praktis (numerik tidak

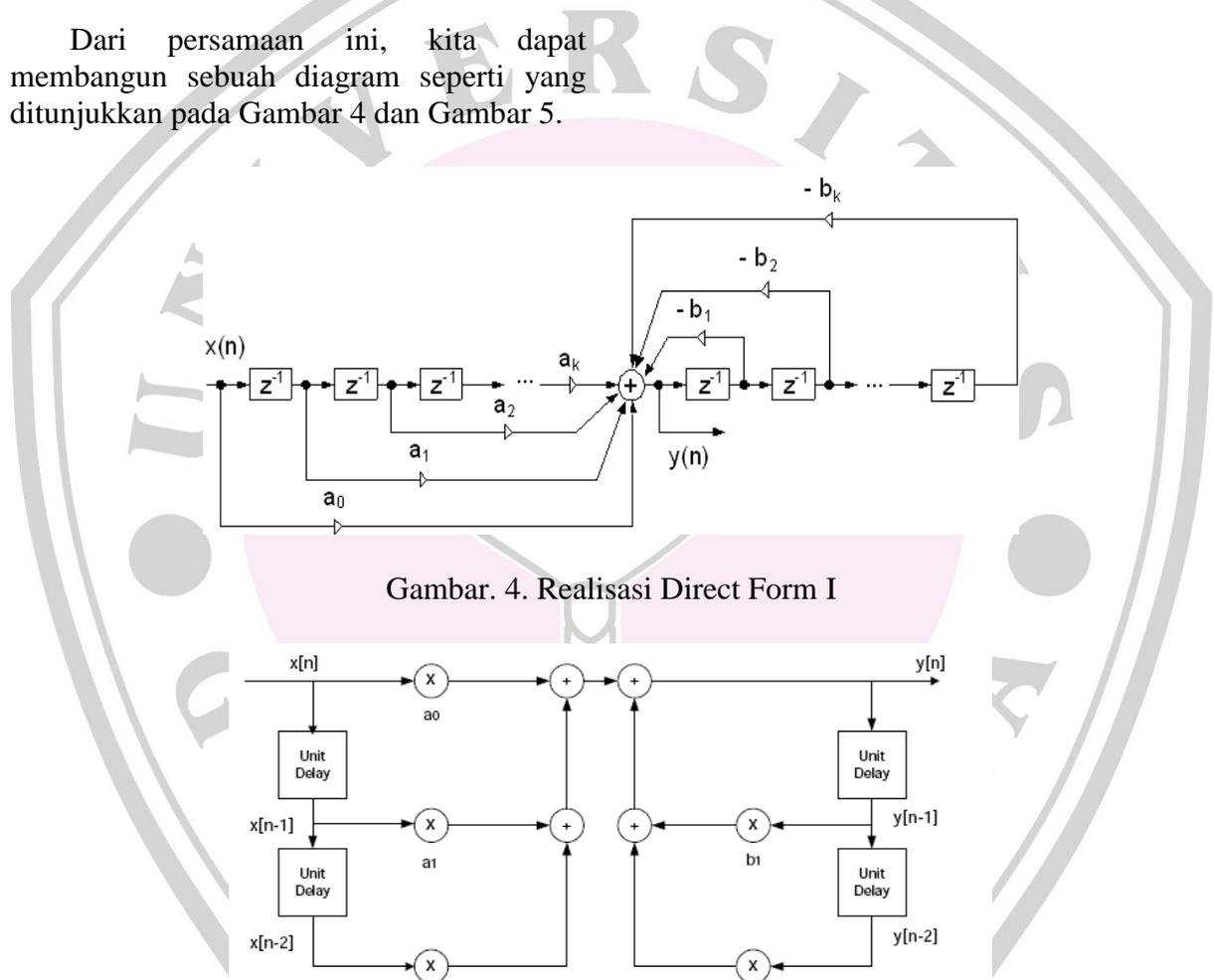
stabil) untuk desain yang kompleks [Ifeachor, 2002].

Secara umum, bentuk ini membutuhkan elemen delay $2N$ (untuk sinyal input dan output) untuk filter order N .

Direct Form I realisasinya langsung dapat dikutip dalam persamaan matematis:

$$y(n) = \sum_{i=0}^k a_i(xn - i) - \sum_{j=0}^k a_j(y_n - j) \quad (5)$$

Dari persamaan ini, kita dapat membangun sebuah diagram seperti yang ditunjukkan pada Gambar 4 dan Gambar 5.



Gambar. 4. Realisasi Direct Form I

Gambar. 5. Skema Diagram IIR filter.

Direct Form II:

Alternatif Direct Form II hanya membutuhkan N keterlambatan unit, dimana N adalah urutan filter, berpotensi setengah sebanyak Form Direct I. Struktur ini diperoleh dengan membalik urutan pembilang

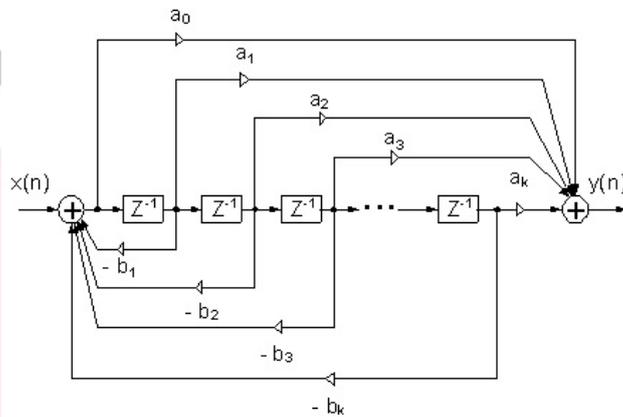
dan penyebut bagian dari Direct Form I, karena mereka sebenarnya dua sistem linear, dan properti komutatif berlaku. Kemudian, salah satu akan melihat bahwa ada dua kolom penundaan (z^{-1}) yang memanfaatkan off net pusat, dan ini dapat dikombinasikan karena mereka banyak.

Kelemahannya adalah bahwa Direct Form II meningkatkan kemungkinan aritmatika over-flow filter Q tinggi atau resonansi [Ruggiero, 1982]. Telah terbukti bahwa Q meningkat, suara bulat-off dari kedua direct form topologi meningkat tanpa batas. Hal ini karena, secara konseptual,

sinyal pertama melewati filter semua-kutub (yang biasanya meningkatkan gain pada frekuensi resonansi) sebelum hasil yang jenuh, kemudian melewati Filter semua-nol (yang sering melemahkan sebanyak setengah semua kutub menguatkan). Realisasinya Form II Direct dapat dikutip dalam persamaan matematis:

$$W(z) = \frac{X(z)}{D(z)} = \frac{X(z)}{(1 + b_1z^{-1} + b_2z^{-2} + \dots + b_kz^{-k})} \quad (6)$$

Dari persamaan ini, kita dapat membangun sebuah diagram seperti pada Gambar 6.

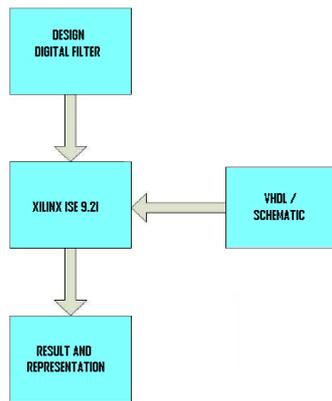


Gambar. 6. Realisasi Direct Form II

Desain Digital Filter

Bagian ini akan membahas perencanaan, mendesain lowpass, highpass chebysev filter digital dengan melakukan perhitungan matematis. Untuk diterapkan dalam Xilinx Ise 9.2i, dengan VHDL singkatan VHSIC

bahasa deskripsi perangkat keras adalah bahasa deskripsi perangkat keras yang digunakan dalam desain otomatisasi elektronik untuk menggambarkan sistem digital dan sinyal campuran seperti bidang programable gate array dan sirkuit terpadu. Blok diagram sistem secara keseluruhan dapat digambarkan sebagai berikut:



Gambar. 7. Blok Diagram Desain Filter Digital

Lokasi kutub berbagai Butterworth dan Chebyshev filter dan koefisien dari polynomials sesuai telah diperoleh dan ditabulasi oleh Weinberg (lihat referensi Tabel 1). Sebuah set disingkat dari koefisien polinomial untuk fungsi Butterworth dan Chebyshev disajikan di sini, bentuk sebagian

faktor dari polinomial ini telah dihitung dan ditabulasikan. Polinomial diberikan dalam bentuk dasar yang dibutuhkan untuk membangun fungsi transfer low-pass. Bentuk umum untuk fungsi transfer yang low-pass dalam tabel adalah:

Tabel 1. Koefisien low-pass filter penyebut polinomial [Winder, 2002]

Order	B0	B1	B2	B3	B4	B5
BUTTERWORTH						
1	1	1				
2	1	1.414	1			
3	1	2	2	1		
4	1	2.613	3.414	2.613	1	
5	1	3.236	5.236	5.236	3.236	1
CHEBYSHEV 0.5 DB RIPPLE						
1	2.863	1				
2	1.516	1.425	1			
3	0.716	1.535	1.253	1		
4	0.379	1.025	1.717	1.197	1	
5	0.179	0.753	1.309	1.937	1.174	1
CHEBYSHEV 1 DB RIPPLE						
1	1.965	1				
2	1.103	1.098	1			
3	0.491	1.238	0.988	1		
4	0.276	0.743	1.454	0.953	1	
5	0.123	0.581	0.974	1.689	0.937	1
CHEBYSHEV 2 DB RIPPLE						
1	1.308	1				
2	0.823	0.804	1			
3	0.327	1.022	0.738	1		
4	0.206	0.517	1.256	0.716	1	
5	0.082	0.459	0.693	1.499	0.706	1
CHEBYSHEV 3 DB RIPPLE						
1	1.002	1				
2	0.708	0.645	1			
3	0.251	0.928	0.597	1		
4	0.177	0.405	1.169	0.582	1	
5	0.063	0.408	0.549	1.415	0.574	1

$$G(s) = \frac{A_0}{B_0 + B_1s + B_2s^2 + \dots + B_k s^k} \quad (7)$$

Koefisien penyebut dari persamaan (7) ditabulasikan dalam Tabel 1 untuk fungsi

Butterworth dan fungsi Chebyshev untuk empat nilai yang mungkin dari ripple

passband (0,5 dB, 1 dB, 2 dB, dan 3 dB). Perintah filter yang digunakan adalah bentuk $k = 1$ sampai $k = 5$. Jika gain di dc yang diinginkan, pembilang konstan dipilih sebagai $A_0 = B_0$ (Catatan: Koefisien didefinisikan sesuai dengan persamaan (7) (5))

Contoh Desain Low-Pass Filter Digital

Desain low-pass filter digital yang berasal dari orde kedua Chebyshev analog Filter

$$G(p) = \frac{1,516}{1,516 + 1,425p + p^2} \quad (8)$$

Frekuensi $\omega_r = 1$ rad / s dalam prototipe harus sesuai dengan $f_r = 100$ Hz dalam filter digital, sehingga desain harus didasarkan

dengan 3 dB frekuensi cutoff 100 Hz. Tingkat sampling dari sistem ini adalah 1KHz dan 0,5 dB ripple filter.

Solusi

transfer analog normal dari orde kedua Chebyshev Filter dengan 0,5 dB ripple diperoleh dari Tabel 1, dengan menggunakan variabel dummy p, fungsi ini

$$c = \cot \frac{\pi}{2} (0,2) = \cot \frac{\pi}{10} = 3,078 \quad (9)$$

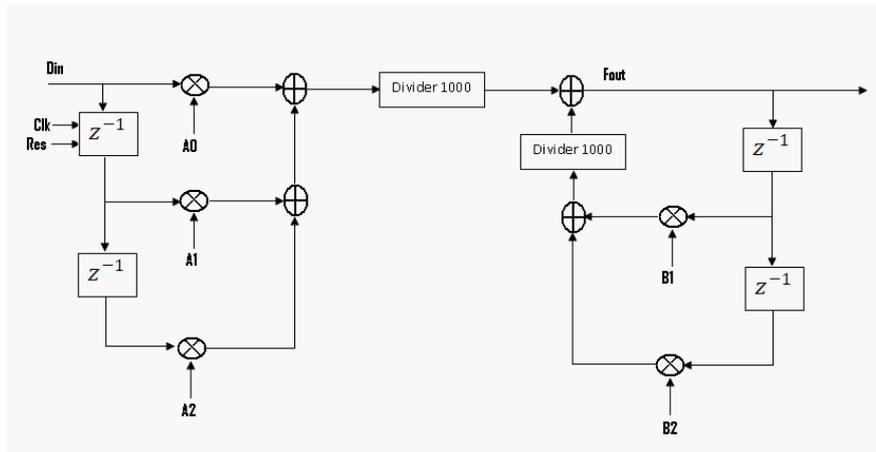
Transformasi yang diperlukan adalah

$$p = 3,078 \frac{(1 - z^{-1})}{(1 + z^{-1})}$$

$$H(z) = G(p) \Big|_{p=3,078 \frac{(1 - z^{-1})}{(1 + z^{-1})}}$$

$$H(z) = \frac{1,516}{1,516 + 1,425p + p^2} \Big|_{p=3,078 \frac{(1 - z^{-1})}{(1 + z^{-1})}}$$

$$\begin{aligned} H(z) &= \frac{1,516}{1,516 + 1,425 \left[3,078 \frac{(1 - z^{-1})}{(1 + z^{-1})} \right] + \left[3,078 \frac{(1 - z^{-1})}{(1 + z^{-1})} \right]^2} \\ &= \frac{1,516}{1,516 + 1,425 \left[3,078 \frac{(1 - z^{-1})}{(1 + z^{-1})} \right] + \left[3,078 \frac{(1 - z^{-1})}{(1 + z^{-1})} \right]^2} \times \frac{(1 + z^{-1})^2}{(1 + z^{-1})^2} \\ &= \frac{1,516 (1 + z^{-1})^2}{1,516 (1 + z^{-1})^2 + 4,386 (1 - z^{-1}) (1 + z^{-1}) + 9,474 (1 - z^{-1})^2} \\ &= \frac{0,098 + 0,197 z^{-1} + 0,098 z^{-2}}{1 - 1,035 z^{-1} + 0,429 z^{-2}} \end{aligned} \quad (10)$$



Gambar. 8. Realisasi Sistem Persamaan 10

Koefisien:

A0 = 0,098	→	98 Desimal = 0062 Hex
A1 = 0,197	→	197 Desimal = 00C5 Hex
A2 = 0,098	→	98 Desimal = 0062 Hex
B1 = -1,035	→	-1035 Desimal = FBF5 Hex
B2 = 0,429	→	429 Desimal = 01AD Hex

Proses Penerapan filter IIR

Sinyal input analog harus sampel pertama dan digital menggunakan ADC (analog-to-digital converter). Bilangan biner yang dihasilkan, mewakili nilai-nilai sampel berturut-turut dari sinyal input, akan ditransfer ke prosesor, yang melakukan perhitungan numerik pada mereka. Perhitungan ini biasanya melibatkan mengalikan nilai input oleh konstanta dan menambahkan produk bersamaan. Jika perlu, hasil perhitungan tersebut, yang kini mewakili nilai-nilai sampel dari sinyal disaring, adalah output melalui DAC (digital to converter analog) untuk mengubah sinyal kembali ke bentuk analog. Modul yang digunakan untuk mengimplementasikan filter IIR adalah sebagai berikut: Perkalian Modul, Penambahan Modul, Modul pembagi bahwa semua itu dikembangkan menggunakan VHDL.

SIMULASI HASIL DAN ANALISIS

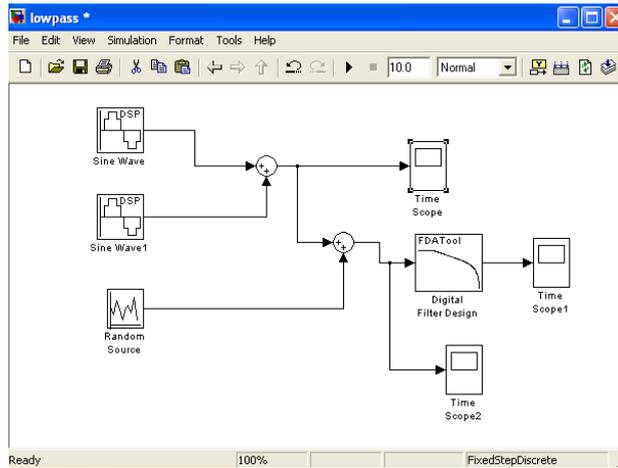
Digital Filter Desain dengan Simulink

Low-Pass Filter Spesifikasi filter seperti:

Respon Jenis: Low-Pass

1. Metode Desain: IIR Chebyshev
2. Filter Order: 2
3. Frekuensi spesifikasi: F_s (1000Hz), F_{pass} (100 Hz)
4. Besaran Keterangan: 0.5 dB

Hasil pengujian dalam program Matlab untuk memasukkan 2 adalah diperoleh tabel berikut tes dalam program Matlab untuk memasukan 2, 6, 8, ..., 20:



Gambar. 9. Simulink Simulasi Low -Pass IIR Chebyshev Filter

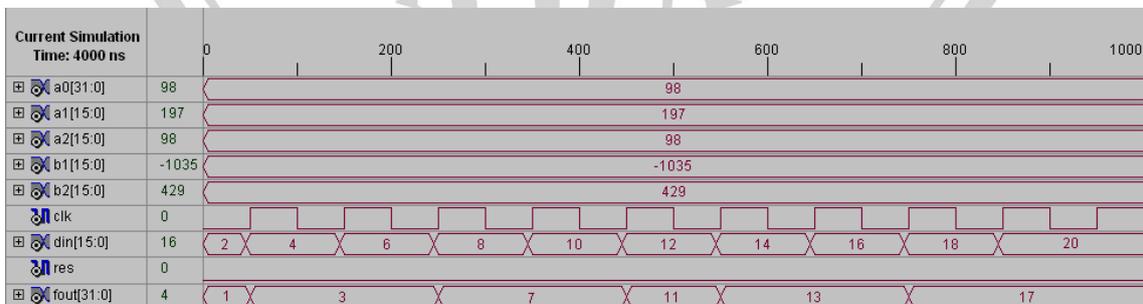
Tabel 2. Hasil simulasi Low-pass filter dengan Simulink

X(n)	Y(n) Simulink
2	1,8
4	3.7
6	5.6
8	7.4
10	9.3
12	11.1
14	13.0
16	14.9
18	16.7
20	18.6

Simulasikan Implementasi Filter Berdasarkan FPGA

Low-Pass Filter: koefisien yang dirancang diberikan dalam persamaan (10). Sekarang kode VHDL ini digunakan untuk

menghasilkan rangkaian menggunakan alat sintesis Xilinx dan disimulasikan menggunakan Xilinx simulator dengan input tertentu (2, 4, 6, ..., 20 Des) dan input dan output bentuk gelombang yang sesuai ditunjukkan pada Gambar 10



Gambar. 10. Perilaku / Post-Route Simulasi Hasil Rendah Pass Filter (LPF)

ANALISIS

Dalam pengujian terjadi kesalahan atau perbedaan hasil antara program Matlab de-

ngan Xilinx. Kesalahan yang terjadi karena keterbatasan tipe data di Xilinx, dengan pembulatan pada fraksi, terlihat pada Tabel 3

Tabel 3. Perbandingan simulasi low-pass filter simulink dengan Xilinx

X(n)	Y(n) Simulink	Y(n) Xilinx
2	1,8	1
4	3,7	3
6	5,6	3
8	7,4	7
10	9,3	7
12	11,1	11
14	13,0	13
16	14,9	13
18	16,7	17
20	18,6	17

Desain sirkuit Band-Pass telah diselesaikan dengan menggunakan VHDL dan implementasi dalam Xilinx Spartan-3E (paket: FG320, kecepatan: -5), dalam proses pembuatan desain ini menggunakan Xilinx

ISE alat desain 9.2i. Ini pemanfaatan tabel berikut sumber daya untuk Spartan-3E pada desain dan pelaksanaan filter digital pengaturan data, lihat Tabel 4 dan Tabel 5.

Tabel 4. Sumber penggunaan data untuk setting rangkaian digital filter untuk Spartan-3e

Circuit	Four Input LUTs {9312}	Occupied Slices {4656}	Bonded IOBs {232}	Total Equivalent Gate Count	Average Connection delay (ns)	Maximum Pin delay (ns)
Low-Pass	147	104	178	1.463	1.475	6.802
Band-Pass	209	114	210	2.408	1.446	4.765
High-Pass	147	104	178	1.463	1.475	6.802

Tabel 5. Laporan advanced HDL Sintesis

Circuit	Total multipliers	Total Adders	Total Registers (Flip-Flops)
Low-Pass	4	5	64
Band-Pass	6	7	256
High-Pass	4	5	64

Periode minimum untuk low-pass filter adalah 1.042ns (Frekuensi Maksimum: 875,362 MHz). Jumlah 4 masukan LUTs untuk filter lowpass adalah 147. Jumlah

gerbang yang digunakan dalam implementasi desain 1463 untuk low-pass.

SIMPULAN

Perancangan Bandpass IIR Digital Filter telah dicapai. Sejumlah temuan telah diidentifikasi selama merancang dan simulasi tahap sebagai berikut:

1. Sejak kutub titik nol di setiap mendesain dalam lingkaran dengan mendesain radius 1 Lowpass Chebyshev filter stabil.
2. Filter IIR yang dapat diimplementasikan berdasarkan FPGAs.
3. Koefisien filter IIR mendesain telah dilakukan dengan perhitungan. VHDL memiliki telah digunakan untuk masuk ke deskripsi hardware. Kode VHDL telah ditulis, disintesis, dipetakan kemudian berhasil dikonfigurasi dan prototyped. Filter IIR dirancang sepenuhnya sesuai dengan persyaratan desain
4. Dalam pengujian rangkaian filter masih ada perbedaan antara hasil program simulasi Matlab program Xilinx. Hal ini dipengaruhi oleh penggunaan komponen pembagi yang melakukan pembagian proses dengan hasil pembulatan. Salah satu masalah dalam simulasi proses program Xilinx adalah penggunaan data integer, dimana data dari fraksi harus pembulatan terlebih dahulu.

DAFTAR PUSTAKA

- [Ruggiero, 1982] Ruggiero, Carmelina, "Signali Biomedici 1", Laboratorio MedInfo, 1982.
- [Freeuk, 2011]Freeuk, "An Introduction to Digital Filter", <http://www.dsptutor.freeuk.com/dfilt1>, access on november, 2011.
- [Williams , 2006] Williams, Arthur Bernard and Taylor, Fred J., "Electronic Filter Design Handbook 4ed", McGraw-Hill, 2006.
- [Ifeachor, 2002] Ifeachor, E. C. and Jervis, B. W., "Digital Signal Processing: A Practical Approach, ser. 2nd Edition". England: Pearson Education Limited, 2002.
- [Winder, 2002] Winder , S., "Analog and Digital Filter Design".British