

**RANCANG BANGUN PENGIRIM CDMA DENGAN  
MENGUNAKAN DSK TMS320C6713 BERBASISKAN  
MATLAB SIMULINK**

**SKRIPSI**

**DICKY JONATHAN  
04 04 03 02 88**



**DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS INDONESIA  
DEPOK  
MEI 2008**

**RANCANG BANGUN PENGIRIM CDMA DENGAN  
MENGUNAKAN DSK TMS320C6713 BERBASISKAN  
MATLAB SIMULINK**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana**

**DICKY JONATHAN  
04 04 03 02 88**



**DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS INDONESIA  
DEPOK  
MEI 2008**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi/Tesis/Disertasi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.**



**Nama: Dicky Jonathan**

**NPM: 0404030288**

**Tanda Tangan:**

**Tanggal: 23 Mei 2008**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh : Dicky Jonathan  
Nama : Dicky Jonathan  
NPM : 0404030288  
Program Studi : Elektro  
Judul Skripsi : RANCANG BANGUN PENGIRIM CDMA  
DENGAN MENGGUNAKAN DSK  
TMS320C6713 BERBASISKAN MATLAB  
SIMULINK

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

### DEWAN PENGUJI

Pembimbing : Ir. Arifin Djauhari, MT ( )  
Penguji : Dr. Ir. Arman Djohan, M.Eng ( )  
Penguji : Ir. Purnomo Sidi. P, PhD ( )

Ditetapkan di : Depok  
Tanggal : 24 Juni 2008

## KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Dr. Ir. Arman Djohan, M.Eng dan Ir. Arifin Djauhari, MT, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini.
- (2) Orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral; dan
- (3) Sahabat yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 23 Mei 2008

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

---

**Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:**

Nama : Dicky Jonathan  
NPM : 0404030288  
Program Studi : Elektro  
Departemen : Elektro  
Fakultas : Teknik  
Jenis Karya : Skripsi

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul :

**RANCANG BANGUN PENGIRIM CDMA DENGAN MENGGUNAKAN DSK TMS320C6713 BERBASISKAN MATLAB SIMULINK**

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada Tanggal : 23 Mei 2008  
Yang menyatakan

(Dicky Jonathan)

## ABSTRAK

Nama : Dicky Jonathan  
Program Studi : Teknik Elektro  
Judul : Rancang Bangun Pengirim CDMA Dengan Menggunakan DSK TMS320C6713 Berbasis Matlab Simulink

Penggunaan prosesor DSP TMS320C6x yang memiliki banyak fungsi ini sudah semakin meluas ke berbagai peralatan elektronik. Aplikasi dari penggunaan prosesor yang dibahas pada skripsi ini merupakan aplikasi dari penggunaan prosesor TMS320C6713 sebagai komponen utama pengirim CDMA. Penggunaan prosesor tersebut diaplikasikan ke teknologi CDMA karena pemrograman prosesor tersebut dapat dilakukan hanya dengan menggunakan komputer dan C6713 DSK. Selain itu, prosesor ini mudah untuk didapatkan di pasar bebas dengan harga terjangkau serta teknologi CDMA sebagai teknologi *spread spectrum* memiliki beberapa keuntungan dan penggunaannya sudah meluas. Proses pemrograman terdiri dari dua proses yaitu perancangan model pengirim CDMA dengan menggunakan perangkat lunak Matlab 7.4.0 Simulink dan proses instalasi bahasa mesin. Setelah proses rancang bangun model pengirim dilakukan, algoritma model tersebut diinstalasi ke dalam memori C6713DSK dengan menggunakan Matlab 7.4.0 Simulink<sup>TM</sup> bekerja sama dengan perangkat lunak CCS untuk membangun sebuah pengirim CDMA yang menggunakan prosesor TMS320C6713. Pengambilan sampel sinyal keluaran dilakukan pada *port* LINE OUT pengirim dengan menggunakan *digital storage oscilloscope* dan hasilnya dibandingkan dengan sampel sinyal keluaran hasil simulasi model simulasi. Proses analisis yang dilakukan meliputi analisis sampel bentuk sinyal (baik sinyal keluaran sebenarnya maupun sinyal keluaran hasil simulasi) serta analisis pengaruh kode PN dan frekuensi gelombang pembawa yang digunakan terhadap spektrum sinyal. Hasil dari analisis menunjukkan adanya pengaruh kode PN dan frekuensi gelombang pembawa terhadap spektrum sinyal dan terjadi kesesuaian (kesamaan) antara bentuk sinyal keluaran sebenarnya dengan bentuk sinyal keluaran hasil simulasi.

### Kata Kunci:

CDMA, Kode PN, Model pengirim, Model Simulasi, Data, RTDX, Sampel, Matlab 7.4.0, Simulink, *Frame*, Waktu cuplik

## ABSTRACT

Name : Dicky Jonathan  
Study Program : Electrical Engineering  
Title : CDMA Transmitter Design Using DSK TMS320C6713  
Based On Matlab Simulink

TMS320C67x DSP processor using had become more widely for several electronic device. Application of processor utilizing that is being researched in that minithesis is an application of TMS320C6713 processor utilizing as a main part of CDMA transmitter. The processor utilizing is being applied to CDMA technology because the processor can be programmed only use computer and C6713 DSK. Be side of that, the processor can be gotten easily at the market with the achievable cost and CDMA technology as a spread spectrum technology has a several advantages and its utilizing is being more widely too. Programming process can be divided into two process that are CDMA transmitter model designing which use Matlab 7.4.0 Simulink and assembler installation process. After designing process had been done, algorithm of transmitter is being installed to the C6713 DSK memory which use Matlab 7.4.0 cooperate with CCS to build a CDMA transmitter which use TMS320C6713 processor. Output signal sampling process had been done at transmitter's LINE OUT port using digital storage oscilloscope and the result had been compared with simulation output signal. Analyzing process that had been done include signal waveform analysis (real output signal and simulation output signal), and analyzing process about influence of PN code and carrier wave frequency to signal spectrum. The result are PN code and carrier wave frequency influential to the signal spectrum and real output signal waveform had appropriate with the simulation output signal waveform.

Keywords : CDMA, PN Code, Transmitter model, Simulation model, Data, RTDX, Sample, Matlab 7.4.0, Simulink, Frame, Sampling time

## DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR	iv
HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
DAFTAR SINGKATAN	xv
DAFTAR ISTILAH	xvi
<b>1. PENDAHULUAN</b>	<b>1</b>
1.1 LATAR BELAKANG	1
1.2 PERUMUSAN MASALAH	2
1.3 TUJUAN PENELITIAN	2
1.4 BATASAN MASALAH	2
1.5 METODOLOGI PENELITIAN	2
1.6 SISTEMATIKA PENULISAN	3
<b>2. TINJAUAN PUSTAKA</b>	<b>4</b>
2.1 TEKNOLOGI CDMA SECARA UMUM	4
2.2 SPEKTRUM TERSEBAR	5
2.3 TEORI DASAR DS-CDMA	6
2.4 <i>SPREADING</i>	6
2.5 MODULATOR $\Pi / 4$ <i>QPSK</i>	6
2.6 <i>ADDITIVE WHITE GAUSSIAN NOISE</i>	7
2.7 ANALISA SPEKTRUM DENGAN METODE <i>DFT</i>	8
2.8 SIMULINK	8
2.9 PENGENALAN TMS320C6713 DSP	11
<b>3. RANCANG BANGUN</b>	<b>14</b>
3.1 PENJELASAN MODEL PENGIRIM CDMA SECARA UMUM	14
3.2 BLOK "SUMBER DATA"	15
3.2.1 Sub Blok "From RTDX"	16
3.2.2 Sub Blok "Unbuffer"	16
3.2.3 Sub Blok "Buffer"	16
3.2.4 Sub Blok "Repeater Constant"	18
3.2.5 Sub Blok "Repeater Product"	18
3.2.6 Sub Blok "Repeater Integer to Bit Converter"	18
3.2.7 Sub Blok "Repeater Frame Conversion"	18
3.2.8 Sub Blok "Data Synchronizer"	18
3.3 BLOK "PEMBANGKIT I DAN Q"	19
3.3.1 Sub Blok "Pembangkit kode PN" dan Sub Blok "XOR"	19

3.3.2	Sub Blok "Sample Conversion"	20
3.3.3	Sub Blok "Modulator QPSK"	20
3.3.4	Sub Blok "Bit to Integer Converter"	20
3.4	BLOK "TRIGGER"	20
3.4.1	Sub Blok "Step" dan Sub Blok "Integer Delay"	21
3.4.2	Sub Blok "If"	21
3.5	BLOK "RATE TRANSITION"	21
3.6	BLOK "MIXER"	22
3.6.1	Sub Blok "Sine Wave" dan Sub Blok "Cosine Wave"	22
3.6.2	Sub Blok "Sine Product" dan Sub Blok "Cosine Product"	22
3.6.3	Sub Blok "Add"	23
3.7	BLOK "RF ENCODER"	
3.7.1	Sub Blok "Sign I" dan Sub Blok "Sign Q"	24
3.7.2	Sub Blok "Add I", Sub Blok "Add Q" dan Sub Blok "Constant"	24
3.7.3	Sub Blok "Sign After Add I", dan Sub Blok "Sign After Add Q"	24
3.7.4	Sub Blok "Mux" dan Sub Blok "Frame Conversion Encoder"	24
3.7.5	Sub Blok "Bit to Integer Converter Encoder" dan Sub Blok "Sample Conversion Encoder"	24
3.7.6	Sub Blok "To LED Encoder"	24
3.8	BLOK "OUTPUT"	25
3.8.1	Sub Blok "LED"	25
3.8.2	Sub Blok "DAC"	25
<b>4.</b>	<b>UJI COBA DAN ANALISIS MODEL PENGIRIM CDMA</b>	<b>28</b>
4.1	PROSEDUR UJI COBA DAN PENGAMBILAN SAMPEL	28
4.2	ANALISIS	39
4.2.1	Analisis Sampel Bentuk Sinyal	39
4.2.2	Analisis Sampel Spektrum Sinyal	41
4.2.2.1	<i>Analisis Pengaruh Frekuensi Carrier Terhadap Spektrum Sinyal</i>	41
4.2.2.2	<i>Analisis Pengaruh Kode PN Terhadap Spektrum Sinyal</i>	43
<b>5.</b>	<b>KESIMPULAN</b>	<b>46</b>
	<b>DAFTAR ACUAN</b>	<b>47</b>
	<b>DAFTAR PUSTAKA</b>	<b>48</b>
	<b>LAMPIRAN</b>	<b>49</b>

## DAFTAR GAMBAR

Gambar 2.1.	Blok Diagram Stasiun Pengirim CDMA .....	4
Gambar 2.2.	Modulator <i>QPSK</i> .....	7
Gambar 2.3.	Tab Menu “Real Time Workshop” .....	9
Gambar 2.4.	Tab Menu “Solver” .....	9
Gambar 2.5.	Tab Menu “Optimization” .....	9
Gambar 2.6.	Tab Menu “Hardware Implementation” .....	10
Gambar 2.7.	Tab Menu “Link for CCS” .....	10
Gambar 2.8.	Alur instalasi algoritma model ke dalam memori C6713 DSK sebagai <i>Embedded Target</i> .....	11
Gambar 2.9.	Papan C6713 DSK dengan bagian-bagiannya .....	12
Gambar 2.10.	Blok diagram papan C6713 DSK .....	12
Gambar 3.1.	Beberapa pesan yang ditampilkan saat proses pembuatan <i>file .OUT</i> .....	14
Gambar 3.2.	Model pengirim CDMA .....	15
Gambar 3.3.	Blok “Sumber Data” .....	16
Gambar 3.4.	Ilustrasi dari sub blok ”Buffer” .....	17
Gambar 3.5.	Blok ”Pembangkit I dan Q” .....	19
Gambar 3.6.	Proses pembangkitan kode PN .....	19
Gambar 3.7.	Blok ”Trigger” .....	21
Gambar 3.8.	Blok ”Rate Transition” .....	22
Gambar 3.9.	Blok”Mixer” .....	22
Gambar 3.10.	Blok ”RF Encoder” .....	23
Gambar 3.11.	Blok ”Output” .....	25
Gambar 3.12.	Rangkaian modulator analog yang dapat dipakai dari penggunaan keluaran LED .....	26
Gambar 3.13.	Pemrosesan data pada model pengirim saat vektor data adalah pengulangan dari (1 0) .....	27
Gambar 4.1.	Keluaran sebenarnya saat vektor data adalah pengulangan dari (1 0) .....	29
Gambar 4.2.	Keluaran sebenarnya (dalam bingkai waktu yang berbeda) saat vektor data adalah pengulangan dari (1 0) .....	29
Gambar 4.3.	Keluaran blok “Mixer” (hasil simulasi) saat vektor data adalah pengulangan dari (1 0) .....	29
Gambar 4.4.	Keluaran sebenarnya saat vektor data adalah pengulangan dari (0 1 1) .....	30
Gambar 4.5.	Keluaran sebenarnya (dalam bingkai waktu yang berbeda) saat vektor data adalah pengulangan dari (0 1 1) .....	30
Gambar 4.6.	Keluaran blok “Mixer” (hasil simulasi) saat vektor data adalah pengulangan dari (0 1 1) .....	30
Gambar 4.7.	Keluaran sebenarnya saat vektor data adalah pengulangan dari (1 0 1) .....	31
Gambar 4.8.	Keluaran sebenarnya (dalam bingkai waktu yang berbeda) saat vektor data adalah pengulangan dari (1 0 1) .....	31

Gambar 4.9.	Keluaran blok “Mixer” (hasil simulasi) saat vektor data adalah pengulangan dari (1 0 1) .....	31
Gambar 4.10.	Keluaran sebenarnya saat vektor data adalah pengulangan dari (1 1 0) .....	31
Gambar 4.11.	Keluaran sebenarnya (dalam bingkai waktu yang berbeda) saat vektor data adalah pengulangan dari (1 1 0) .....	32
Gambar 4.12.	Keluaran blok “Mixer” (hasil simulasi) saat vektor data adalah pengulangan dari (1 1 0) .....	32
Gambar 4.13.	Keluaran sebenarnya saat vektor data adalah pengulangan dari (0 1 0) dengan <i>data rate</i> tinggi dengan frekuensi pembawa = 100 Hz .....	32
Gambar 4.14.	Keluaran sebenarnya saat vektor data adalah pengulangan dari (0 1 0) dengan <i>data rate</i> rendah dengan frekuensi pembawa = 25 Hz .....	32
Gambar 4.15.	Keluaran blok “Mixer” (hasil simulasi) saat vektor data adalah pengulangan dari (0 1 0) .....	33
Gambar 4.16.	Keluaran sebenarnya saat vektor data adalah pengulangan dari (0 0 1) dengan <i>data rate</i> tinggi dengan frekuensi pembawa = 100 Hz .....	33
Gambar 4.17.	Keluaran sebenarnya saat vektor data adalah pengulangan dari (0 0 1) dengan <i>data rate</i> rendah dengan frekuensi pembawa = 25 Hz .....	33
Gambar 4.18.	Keluaran blok “Mixer” (hasil simulasi) saat vektor data adalah pengulangan dari (0 0 1) .....	33
Gambar 4.19.	Blok model simulasi .....	34
Gambar 4.20.	Spektrum sinyal sebenarnya saat data adalah pengulangan dari vektor data (1 0), frekuensi pembawa = 200 Hz dan <i>data rate</i> rendah .....	34
Gambar 4.21.	Spektrum sinyal sebenarnya saat data adalah pengulangan dari vektor data (1 0), frekuensi pembawa = 200 Hz dan <i>data rate</i> standar .....	35
Gambar 4.22.	Spektrum sinyal sebenarnya saat data adalah pengulangan dari vektor data (1 0), frekuensi pembawa = 200 Hz dan <i>data rate</i> tinggi .....	35
Gambar 4.23.	Spektrum sinyal hasil simulasi saat data adalah pengulangan dari vektor data (1 0), frekuensi pembawa = 200 Hz dan <i>data rate</i> rendah .....	35
Gambar 4.24.	Spektrum sinyal hasil simulasi saat data adalah pengulangan dari vektor data (1 0), frekuensi pembawa = 200 Hz dan <i>data rate</i> standar .....	36
Gambar 4.25.	Spektrum sinyal hasil simulasi saat data adalah pengulangan dari vektor data (1 0), frekuensi pembawa = 200 Hz dan <i>data rate</i> tinggi .....	36
Gambar 4.26.	Spektrum sinyal sebenarnya saat data adalah pengulangan dari vektor data (0 1 0), frekuensi pembawa = 100 Hz dan <i>data rate</i> tinggi .....	36
Gambar 4.27.	Spektrum sinyal sebenarnya saat data adalah pengulangan dari vektor data (0 1 0), frekuensi pembawa = 25 Hz dan .....	

	<i>data rate</i> rendah .....	37
Gambar 4.28.	Spektrum sinyal keluaran hasil simulasi saat data adalah pengulangan dari vektor data (0 1 0), frekuensi pembawa = 100 Hz dan <i>data rate</i> tinggi .....	37
Gambar 4.29.	Spektrum sinyal keluaran hasil simulasi saat data adalah pengulangan dari vektor data (0 1 0), frekuensi pembawa=25Hz dan <i>data rate</i> rendah .....	37
Gambar 4.30.	Spektrum sinyal keluaran hasil simulasi saat data adalah pengulangan dari vektor data (0 1 0), frekuensi pembawa=50Hz dan <i>data rate</i> standar .....	38
Gambar 4.31.	Keluaran sebenarnya saat vektor data adalah pengulangan dari (1 0), frekuensi pembawa = 200 Hz dan <i>data rate</i> = 100 / 6 bps .....	40
Gambar 4.32.	Keluaran sebenarnya saat vektor data adalah pengulangan dari (1 0), frekuensi pembawa = 200 Hz dan <i>data rate</i> = 100 / 3 bps .....	40
Gambar 4.33.	Spektrum sinyal keluaran hasil simulasi saat lebar kode PN pertama adalah 3 bit per bit data .....	43
Gambar 4.34.	Spektrum sinyal keluaran hasil simulasi saat lebar kode PN kedua adalah 3 bit per bit data .....	43
Gambar 4.35.	Spektrum sinyal keluaran hasil simulasi saat lebar kode PN ketiga adalah 7 bit per bit data .....	44
Gambar 4.36.	Spektrum sinyal keluaran hasil simulasi saat lebar kode PN keempat adalah 7 bit per bit data .....	44
Gambar 4.37.	Spektrum sinyal keluaran hasil simulasi saat lebar kode PN kelima adalah 15 bit per bit data .....	44
Gambar 4.38.	Spektrum sinyal keluaran hasil simulasi saat lebar kode PN kelima adalah 15 bit per bit data (dengan bingkai frekuensi yang lebih lebar) .....	45

## DAFTAR TABEL

Tabel 2.1	Pergeseran Fasa Sebagai Fungsi Informasi .....	7
Tabel 3.1	Deskripsi cara kerja sub blok "Buffer" .....	17
Tabel 3.2	Keluaran dari beberapa sub blok penyusun blok "RF Encoder" pada seluruh kemungkinan masukan .....	26
Tabel 4.1	Pola keberulangan 18 bit hasil <i>spreading</i> untuk data (1 0 0), (0 1 0), dan (0 0 1) .....	40



## DAFTAR LAMPIRAN

Lampiran 1	M-file “driver2.m” .....	48
Lampiran 2	M-file “spectrum_analyzer.m” .....	49
Lampiran 3	M-file “direct_spectrum_analyzer.m” .....	50



## DAFTAR SINGKATAN

AWGN	Additive White Gaussian Noise
CCS	Code Composer Studio
CDMA	Code Division Multiple Access
DFT	Discrete Fourier Transform
DS	Direct Sequence
DSK	Digital Starter Kit
DSP	Digital Signal Processing
FDMA	Frequency Division Multiple Access
FFT	Fast Fourier Transform
FH	Frequency Hoping
JTAG	Joint Team Action Group
PN	Pseudo Noise
QPSK	Quaternary Phase Shift Keying
RTDX	Real Time Data Exchange
TDMA	Time Division Multiple Access
TH	Time Hoping
USB	Universal Serial Bus
XOR	Exclusive OR



## DAFTAR ISTILAH

1. *Bandwidth*: suatu jargon dalam dunia telekomunikasi yang berarti jarak atau lebar antara frekuensi yang masih memungkinkan untuk digunakan untuk mentransmisikan data.
2. *Chip*: Suatu kelompok bit-bit yang memodulasi bit data menjadi suatu *spread spectrum* yang lebih lebar. Dalam teknologi CDMA, *chip* adalah kode PN itu sendiri.
3. *Compile*: Proses pengubahan bentuk suatu algoritma dari bentuk bahasa yang tingkatnya lebih tinggi menjadi bentuk bahasa yang tingkatnya lebih rendah atau sederhana.
4. *Data*: Bit-bit informasi efektif sebelum proses pengulangan yang dikirimkan dari komputer menuju *embedded target* TMS320C6713 DSK.
5. *Data rate*: Jumlah bit data yang dikirimkan per detik.
6. *Despreading*: Proses pemisahan bit-bit informasi dari sinyal dan kode PN-nya.
7. *Digital spread spectrum*: Spektrum suatu sinyal yang kontinu dengan nilai frekuensi batas bawah dan frekuensi batas atas tertentu dan digunakan dalam sistem komunikasi digital.
8. *Error Control Coding*: Sistem pengkodean yang diterapkan pada bit-bit yang akan ditransmisikan dengan tujuan untuk mengurangi efek negatif dari *noise* yaitu berupa kerusakan bit-bit informasi.
9. *Frame*: Gabungan dari beberapa *sample*. Pengiriman sebuah *frame* ekuivalen dengan pengiriman beberapa *sample* secara bersamaan (serempak).
10. *Jamming*: Suatu interferensi kuat yang terjadi secara disengaja karena sinyal-sinyal berfrekuensi berdekatan atau berspektrum sama.
11. *spread spectrum*: Spektrum suatu sinyal yang kontinu dengan nilai frekuensi batas bawah dan frekuensi batas atas tertentu
12. *Traffic capacity*: Suatu ukuran banyaknya sinyal yang dapat ditransmisikan pada suatu kanal tanpa menimbulkan kerusakan informasi pada sinyal yang ditransmisikan
13. Model pengirim: Model yang algoritmanya dapat dibangkitkan dalam bentuk kode-kode bahasa C atau C++ agar dapat di-*compile* menjadi bahasa mesin prosesor TMS320C6713.
14. Model simulasi: Model yang hanya diperuntukkan untuk tujuan simulasi dan algoritmanya tidak dapat digunakan untuk memprogram prosesor TMS320C6713.
15. *Multiple Access User*: Pengguna-pengguna yang dapat melakukan akses secara bersamaan.
16. *Multipath fading*: Fenomena variasi arah dari sinyal-sinyal yang ditransmisikan akibat terjadinya pantulan atau pembiasan.
17. RTDX (*Real Time Data Exchange*): Suatu fasilitas yang disediakan oleh perangkat lunak CCS untuk menunjang terjadinya komunikasi data antara komputer dengan TMS320C6713 DSK.
18. *Sample*: Bit-bit data yang telah diulang sedemikian sehingga satu bit data diperbanyak (diulang) menjadi N bit data dengan N adalah jumlah bit kode PN dalam satu urutan (*sequence*).

19. *Sample Time*: Waktu cuplik *sample*.
20. *Spreading*: Proses modulasi bit-bit data dengan kode PN atau dengan kata lain penggabungan spektrum bit-bit data dengan spektrum kode PN menghasilkan *spread spectrum*.
21. *Transceiver*: Sistem pengirim dan penerima



# BAB I

## PENDAHULUAN

### 1.1 LATAR BELAKANG

Komunikasi dengan menggunakan teknologi spektrum tersebar telah digunakan sejak lama dalam dunia militer selama lebih dari setengah abad. Hal tersebut terjadi karena dua alasan yaitu teknologi ini dapat mengatasi efek dari *jamming* dan dapat menyembunyikan informasi tersembunyi dalam suatu sinyal dari pihak yang tidak berhak mengetahuinya.

Sebagai tindak lanjut untuk mengatasi pertumbuhan permintaan akan sistem komunikasi bergerak, teknologi spektrum tersebar digital dapat memberikan efisiensi *bandwidth* yang lebih tinggi bagi alokasi spektrum sistem komunikasi nirkabel, dapat melayani lebih banyak jumlah *multiple access user* jika dibandingkan dengan teknologi telekomunikasi analog ataupun digital lainnya. Selain itu, teknologi spektrum tersebar digital dapat bekerja atau digunakan pada alokasi frekuensi yang sama untuk setiap pengguna.

Teknologi CDMA (*Code Division Multiple Access*) merupakan salah satu teknologi spektrum tersebar digital. Penggunaan sistem komunikasi CDMA sekarang telah sangat populer. Hal ini dikarenakan teknologi CDMA memungkinkan penggunaan frekuensi pengguna dan slot waktu yang sama. Jika kita bandingkan dengan teknologi FDMA yang menggunakan pembagian alokasi frekuensi untuk pengguna ataupun teknologi TDMA yang menggunakan pembagian alokasi slot waktu untuk pengguna, teknologi CDMA memungkinkan penggunaan frekuensi dan slot waktu yang sama namun dengan spektrum yang lebar. Selain itu, teknologi CDMA juga memberikan beberapa keuntungan seperti:

- a. Mengurangi pemakaian daya pada pemancar
- b. Mengurangi interferensi pada peralatan elektronika
- c. Dapat meningkatkan *traffic capacity* pada saluran telepon
- d. Dapat digunakan untuk komunikasi data pada beberapa teknologi seperti faksimili, dan internet.
- e. Dapat meningkatkan kualitas suara dan mengurangi efek negatif dari *multipath fading*

Ide teknologi CDMA (yang diterapkan pada sebuah prosesor DSP) diharapkan dapat menjadi suatu ide yang cukup baik bagi perluasan jaringan komunikasi tanpa harus terlalu dibatasi oleh keterbatasan *bandwidth*. Penggunaan prosesor DSP sudah semakin meluas ke berbagai jenis perangkat elektronik modern termasuk telepon selular. Selain itu, prosesor DSP tersebut mudah didapat di pasaran, berkecepatan tinggi serta dapat digunakan untuk mengoperasikan banyak aplikasi atau algoritma. Oleh karena itu, prosesor DSP yang salah satunya adalah prosesor TMS320C6713 dapat disebut sebagai prosesor yang memiliki banyak fungsi. Oleh karena itu, penggunaan prosesor tersebut sebagai salah satu komponen rancang bangun pengirim CDMA dapat dilakukan. Rancang bangun pengirim CDMA tersebut dapat dilakukan dengan melakukan pemrograman prosesor TMS320C6713 dengan menggunakan komputer (perangkat lunak Matlab 7.4.0 Simulink 6.0 dan Code Composer Studio) dan C6713 DSK yang telah tersedia bagi penulis untuk menunjang proses pemrograman tersebut.

## **1.2 PERUMUSAN MASALAH**

Masalah yang dibahas adalah pemodelan pengirim CDMA dengan menggunakan SIMULINK yang terdapat dalam MATLAB dan mengaplikasikan model pengirim CDMA tersebut ke dalam DSP *Starter Kit* TMS320C6713.

## **1.3 TUJUAN PENELITIAN**

Tujuan dari penelitian ini adalah memperoleh suatu perangkat pengirim CDMA yang dirakit dalam bentuk sebuah prosesor DSP berbasis Matlab Simulink.

## **1.4 BATASAN MASALAH**

Hal-hal yang tidak dilakukan di dalam penelitian ini yaitu:

- a. Tidak melakukan uji coba pada berbagai data input.
- b. Tidak melakukan pengujian pada perangkat penerima CDMA.

## 1.5 METODOLOGI PENELITIAN

Metodologi penelitian yang dipakai adalah proses simulasi model pengirim CDMA melalui model simulasinya dan mengaplikasikan atau menginstalasi model pengirim tersebut ke dalam DSP *Starter Kit* untuk membangun sebuah pengirim CDMA. Sinyal keluaran dari pengirim CDMA yang dibangun dibandingkan dengan sinyal keluaran hasil simulasi untuk dianalisa.

## 1.6 SISTEMATIKA PENULISAN

### BAB 1 PENDAHULUAN

Bab ini akan membahas tentang latar belakang penulisan, tujuan penulisan, batasan masalah dan sistematika penulisan skripsi untuk memberikan gambaran umum mengenai permasalahan yang dibahas dalam seminar ini.

### BAB 2 TINJAUAN PUSTAKA

Bab ini akan membahas tentang dasar teori mengenai blok diagram pengirim CDMA dan proses yang terjadi di setiap blok bagian. Selain itu, bab ini juga membahas mengenai perangkat lunak yang digunakan sebagai penunjang instalasi algoritma model ke dalam *embedded target*.

### BAB 3 RANCANG BANGUN

Bab ini akan menjelaskan mengenai perancangan dan algoritma yang merupakan realisasi dari pemrograman blok diagram pengirim CDMA ke dalam sebuah DSP prosesor.

### BAB 4 UJI COBA DAN ANALISIS MODEL PENGIRIM CDMA

Bab ini menjelaskan prosedur uji coba, pengambilan data dari pengirim CDMA (yang dibangun dari algoritma model pengirim CDMA) serta analisis data yang terjadi.

### BAB 5 KESIMPULAN

Bab ini berisi kesimpulan dari hasil yang diperoleh dari penelitian ini.

## BAB 2 TINJAUAN PUSTAKA

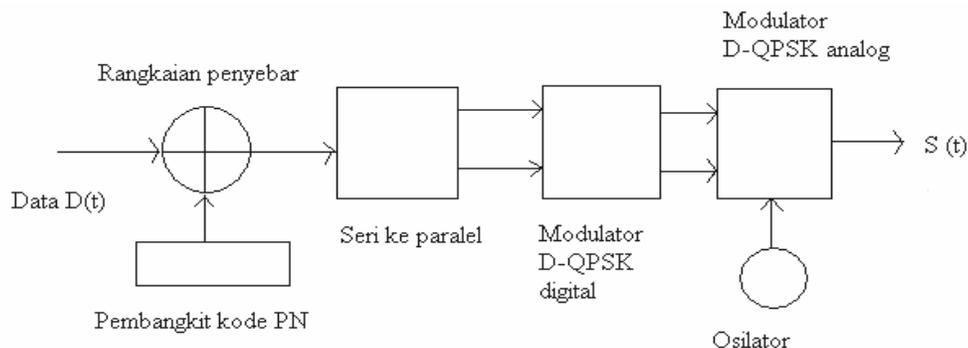
### 2.1 TEKNOLOGI CDMA SECARA UMUM

Sinyal CDMA (*Code Division Multiple Access*) merupakan spektrum tersebar digital yang merepresentasikan bit data 1 dan bit 0 dengan suatu kumpulan sinyal *chip*. Sinyal *chip* itu sendiri merupakan sinyal yang terdiri dari kumpulan bit kode. Sinyal atau kode *chip* merupakan suatu kode identitas yang harus unik untuk setiap penerima. Panjang bit dari kode tersebut akan mempengaruhi pemakaian *bandwidth* yang digunakan. Sinyal *chip* untuk bit 0 berkebalikan dengan sinyal *chip* untuk bit 1. Setiap pasangan komunikasi *transceiver* memiliki identitas sinyal *chip* yang unik (yang merupakan kode PN) sehingga dapat terjadi pemilihan sinyal secara tepat. Proses pemisahan sinyal terjadi pada bagian penerima yang dinamakan proses *despreading*. Spektrum sinyal CDMA terdiri dari frekuensi pembawa, frekuensi bit *chip*, frekuensi bit data, dan frekuensi lain yang berasal dari komposisi acak bit data. Frekuensi bit *chip* membuat spektrum sinyal menjadi cukup lebar.

Secara umum, rangkaian pengirim CDMA terdiri dari:

- 1) Pembangkit kode PN
- 2) *Serial to parallel*
- 3) Modulator *QPSK* digital
- 4) Rangkaian pembanding fasa dan osilator (modulator *QPSK* analog)

Diagram blok rangkaian pengirim CDMA ditunjukkan Gambar 2.1 yang terlampir di halaman berikutnya.



Gambar 2.1. Blok diagram stasiun penerima CDMA [3].

## 2.2 SPEKTRUM TERSEBAR

Spektrum tersebar (*Spread Spectrum*) adalah sistem transmisi yang sinyalnya sengaja disebarkan jauh melebihi pita frekuensi yang disalurkan. Pada bagian pemancar penyebaran sinyal dilakukan dengan menggunakan kode penyebar yang polanya acak. Sinyal tersebar ini kemudian dimampatkan kembali dengan menggunakan kode penyebar yang sama dan sinkron dengan kode penyebar pada bagian pemancar. Pendekatan ini adalah pendekatan yang sangat berbeda dari pendekatan teori informasi Shannon yang mengalokasikan sumber frekuensi atau waktu kepada setiap pengguna, seperti yang kita kenal dalam teknologi *FDMA* ataupun *TDMA*. Setiap pengguna pada sistem ini menggunakan slot frekuensi dan waktu yang bersamaan selama waktu yang dibutuhkan. Dengan cara ini, setiap pengguna akan berkontribusi menghasilkan *noise* yang mempengaruhi pengguna lain, gangguan ini akan mengganggu limit kapasitas, tetapi karena waktu dan bandwidth alokasi dapat digunakan secara bebas, resultan akhir kapasitas akan bertambah secara signifikan dibandingkan sistem konvensional.

Secara umum terdapat 3 metode dalam teknik spektrum tersebar, yaitu :

### a. *Frequency Hoping (FH) Spread Spectrum*

Pada frekuensi *hopping* (FH), frekuensi pembawa tidak konstan, tetapi mengalami loncatan (*hopping*) frekuensi yang berubah-ubah secara periodik. Sistem FH ini mempunyai 2 jenis loncatan, yaitu *fast hopping*, dimana kecepatan loncatan frekuensi lebih besar dari kecepatan data, sedangkan *slow hopping* adalah sistem pola loncatan bila kecepatan *hopping* frekuensi lebih kecil dari kecepatan data.

### b. *Time Hoping (TH) Spread Spectrum*

Pada metode TH-SS, waktu untuk transmisi data dibagi dalam beberapa interval waktu yang disebut dengan bingkai waktu (*frame*). Tiap *frame* dibagi atas beberapa slot waktu, sehingga data yang ditransmisikan adalah data yang menduduki satu slot waktu dari satu *frame* dengan mengalami loncatan (*hopping*) waktu dari satu *frame* dengan mengalami loncatan (*hopping*) waktu yang ditentukan oleh deret kode.

### c. *Direct Sequence (DS) Spread Spectrum*

Pada DS-SS, modulasi sinyal data dilakukan langsung oleh deretan kode PN (*Pseudo Noise code*) digital yang mempunyai kecepatan kode PN jauh lebih besar dari kecepatan data. Deret kode PN adalah deret kode acak yang dibangkitkan oleh generator yang disebut dengan generator PN (*Pseudo Noise Generator*)

## 2.3 TEORI DASAR DS-CDMA

Dengan menggunakan bantuan kode PN yang dihasilkan oleh generator PN, data yang ditransmisikan disebar (*spreading*) sehingga menduduki pita frekuensi yang lebar. Terbentuknya pita frekuensi yang lebar disebabkan oleh kecepatan kode PN yang jauh lebih besar dari kecepatan data. Kode PN yang dihasilkan oleh generator PN merupakan kode yang acak sehingga penyebaran data dilakukan secara acak tergantung dari keacakan kode PN. Data yang telah disebar kemudian dimodulasi sebelum ditransmisikan. Pada bagian penerima, sinyal data didemodulasi dan kemudian dimampatkan (*dispreading*) untuk mendapatkan data asli dari pemancar.

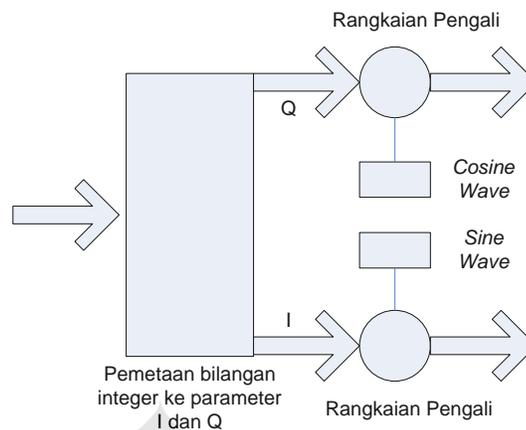
## 2.4 SPREADING

Proses spreading merupakan proses modulasi bit-bit data dengan kode PN menghasilkan kode *chip* baru. Spektrum dari informasi yang semula merupakan spektrum bit-bit data diubah menjadi spektrum kode *chip* yang didominasi dengan spektrum kode PN. Proses ini melibatkan generator kode PN. Pada proses ini, dilakukan operasi NXOR antara bit-bit data dengan kode PN dengan bit *rate* kode PN jauh lebih tinggi dari bit *rate* bit data.

## 2.5 MODULATOR QPSK

Secara umum, sistem modulasi QPSK dapat dibagi menjadi 2 subsistem yaitu sistem pemetaan dan sistem penumpangan gelombang pembawa. Sistem penumpangan gelombang pembawa

Blok diagram modulator QPSK dapat dilihat pada Gambar 2.2 di bawah ini:



Gambar 2.2. Modulator QPSK.

Data masukan modulator berupa bilangan integer (0 sampai 3) yang mewakili 2 bit data dalam setiap periode cuplik. Selanjutnya, bilangan integer tersebut dipetakan ke parameter I dan Q dengan ketentuan hubungan yang ditunjukkan Tabel 2.1 di bawah ini.

Tabel 2.1 Pergeseran fasa sebagai fungsi informasi

Bilangan integer	I	Q
0	0.7071	0.7071
1	-0.7071	0.7071
2	0.7071	-0.7071
3	-0.7071	-0.7071

## 2.6 ADDITIVE WHITE GAUSSIAN NOISE (AWGN)

*Noise* termal merupakan salah satu *noise* dalam sistem komunikasi yang disebabkan oleh pergerakan elektron di dalam konduktor yang ada dalam sistem telekomunikasi, misalnya pada perangkat penerima (*receiver*). *Noise* termal mempunyai nilai kepadatan spektrum yang sama untuk daerah frekuensi yang lebar, yaitu sebesar  $N_0/2$ . *Noise* yang memiliki karakteristik *white* disebut dengan *white noise*. Besarnya nilai *noise* termal berubah secara acak terhadap waktu. Perubahan secara acak tersebut dapat diperkirakan secara statistik dengan mengikuti distribusi Gaussian, dengan nilai tengah = 0. *White noise* ini merusak sinyal dalam bentuk aditif sehingga *noise* termal pada perangkat penerima disebut dengan *Additive White Gaussian Noise (AWGN)*.

## 2.7 ANALISIS SPEKTRUM DENGAN METODE DFT

Metode DFT (*Discrete Fourier Transform*) berfungsi untuk mentransformasikan sinyal  $x(n)$  yang direpresentasikan dalam bentuk vektor menjadi vektor  $X(k)$ . Sinyal  $x(n)$  adalah sinyal diskrit dengan acuan waktu diskrit.

Persamaan transformasi dari  $x(n)$  ke  $X(k)$  dan transformasi dari  $X(k)$  ke  $x(n)$  secara berturut-turut yaitu:

$$X(k+1) = \sum_{n=0}^{N-1} x(n+1)W_N^{kn} \quad (2.1)$$

$$x(n+1) = \frac{1}{N} \sum_{k=0}^{N-1} X(k+1)W_N^{-kn} \quad (2.2)$$

dengan

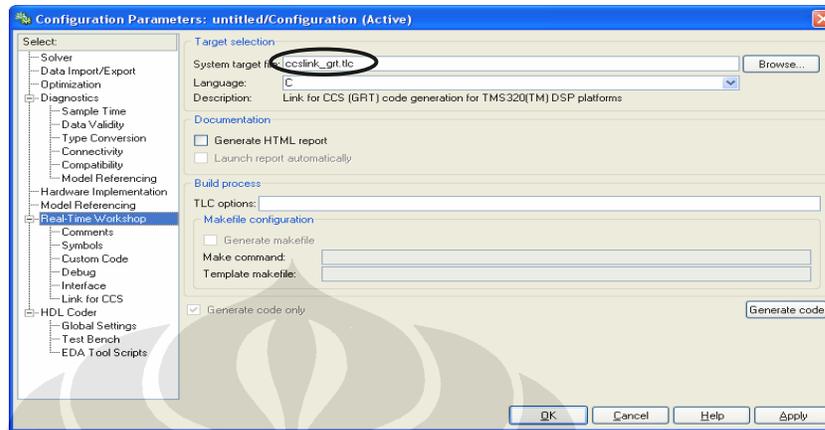
$$W_N = e^{-j\left(\frac{2\pi}{N}\right)} \quad (2.3)$$

Dengan menggunakan metode ini, kita dapat melihat spektrum dari sinyal  $x(n)$  yang direpresentasikan dalam bentuk diskrit.

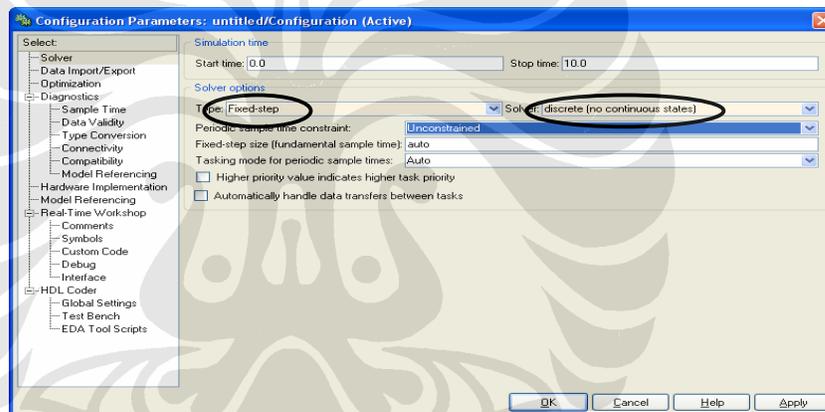
## 2.8 SIMULINK®

Simulink merupakan sub program dari perangkat lunak Matlab 7.4.0 yang berfungsi untuk membantu para perancang untuk merancang suatu model yang akan digunakan untuk proses simulasi. Modul atau sub program tersebut dapat diaktifkan dengan mengetikkan kata "simulink" pada *command window* Matlab 7.4.0. Simulink memberikan banyak pilihan blok yang dapat digunakan untuk mewakili blok-blok yang digunakan sebagai bagian dari model simulasi. Model yang telah disimulasikan dapat berupa model fisik, model matematis, dan model logika (*logic*). Sinyal keluaran hasil simulasi dari suatu model dapat dilihat dengan menggunakan blok "Scope" atau "Time Scope". Simulink juga dapat secara otomatis membangkitkan kode-kode bahasa pemrograman (khususnya dalam basis bahasa C atau C++) dari model yang telah dibentuk. Pemrograman prosesor TMS320C6713 dapat dilakukan dengan menggunakan fasilitas tersebut. Jika pembuatan model telah dilakukan, beberapa parameter konfigurasi harus di-*setting* ulang. Hal ini dapat dilakukan dengan memilih **Simulation** → **Configuration Parameters**. Beberapa parameter yang diubah ditunjukkan oleh Gambar 2.3 hingga Gambar 2.7. Setelah parameter-parameter tersebut di-*setting*

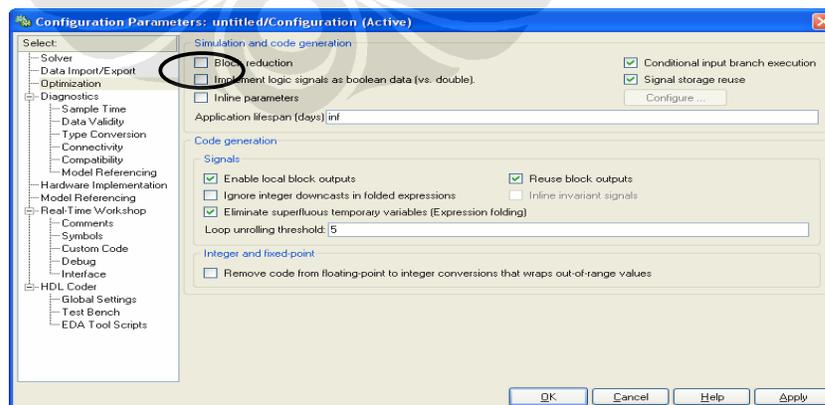
ulang, proses pembangkitan kode-kode program dengan basis bahasa pemrograman C atau C++ dapat dilakukan oleh Matlab 7.4.0 secara otomatis setelah kita menekan kunci **Ctrl + B**.



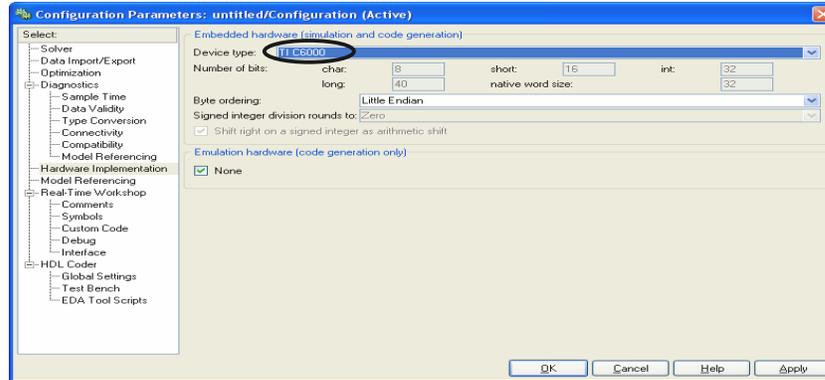
Gambar 2.3. Tab menu “Real Time Workshop”.



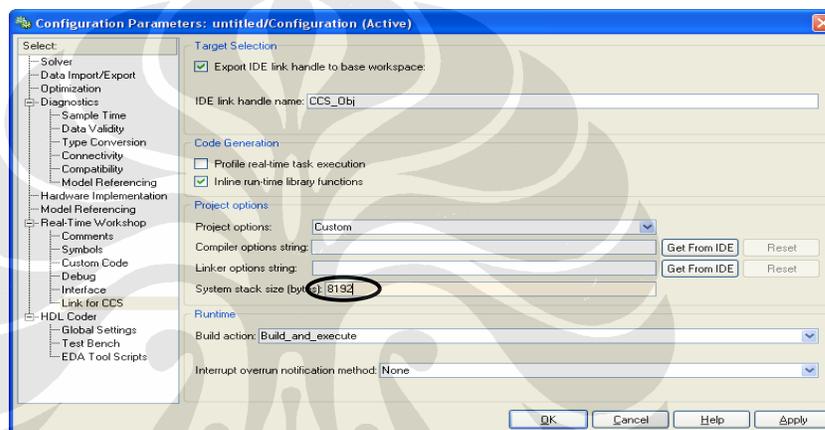
Gambar 2.4. Tab menu “Solver”.



Gambar 2.5. Tab menu “Optimization”.

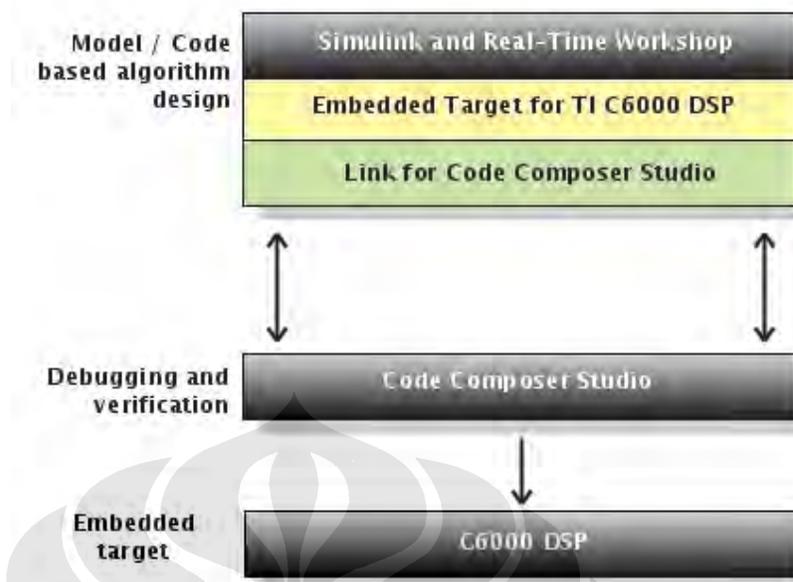


Gambar 2.6. Tab menu "Hardware Implementation".



Gambar 2.7. Tab menu "Link for CCS".

Pembangkitan kode-kode program dan pemrograman prosesor DSP tersebut dilakukan secara otomatis oleh Simulink bekerja sama dengan perangkat lunak CCS (Code Composer Studio™). Alur pemrograman prosesor TMS320C6713 ditunjukkan oleh Gambar 2.8 yang terlampir di halaman berikutnya. Perangkat lunak CCS mengubah (meng-*compile*) kode-kode bahasa C atau C++ yang dibangkitkan oleh Simulink menjadi bahasa mesin dari prosesor TMS320C6713.



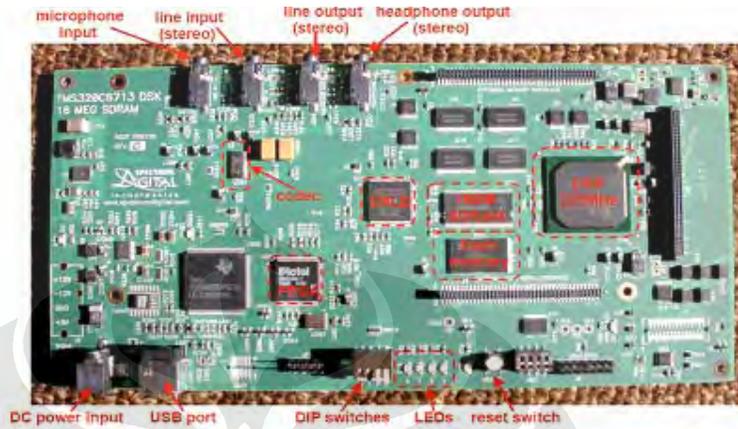
Gambar 2.8. Alur instalasi algoritma model ke dalam memori C6713 DSK sebagai *Embedded Target* [2].

## 2.9 PENGENALAN TMS320C6713 DSP

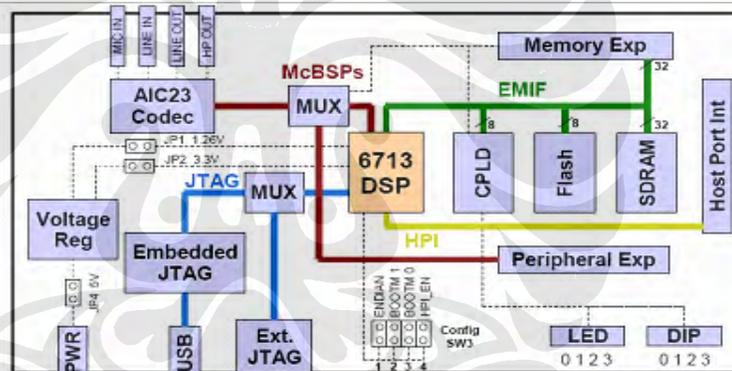
*Digital Signal Processor* (DSP) seperti TMS320C6x merupakan mikroprosesor yang memiliki arsitektur khusus dan sekumpulan instruksi yang dikhususkan untuk pemrosesan sinyal secara *real time*. Prosesor TMS320C6713 memiliki kecepatan yang cukup tinggi dan penggunaan yang cukup baik pada serangkaian proses komputasi numerik yang kompleks. Prosesor TMS320C6x dapat digunakan pada bidang komunikasi, kontrol, pemrosesan gambar dan pemrosesan sinyal suara. Oleh karena itu, penggunaannya sering ditemukan pada beberapa peralatan elektronik seperti HDTV, telepon selular, printer, kamera digital, dan lain-lain. Gambar dari papan elektronik C6713 DSK ditunjukkan oleh Gambar 2.9. Sedangkan blok diagram modul-modulnya ditunjukkan oleh Gambar 2.10.

Penggunaan prosesor ini bersamaan dengan penggunaan TI DSK (*DSP Starter Kit*) sebagai satu kesatuan yang tidak terpisahkan. Pemrograman DSK dapat dilakukan dengan fasilitas perangkat lunak (*software*) yang dikenal dengan nama Code Composer Studio (CCS). Basis bahasa pemrograman yang digunakan perangkat lunak ini yaitu bahasa C dan C++. Proses instalasi program oleh komputer ke memori tertentu pada DSK (*loading*) dilakukan dengan

menggunakan *port* USB yang merupakan bagian dari modul JTAG. Selain menggunakan CCS secara langsung, proses pembuatan kode program dapat dilakukan secara tidak langsung dengan menggunakan Matlab Simulink versi 7.4.0.



Gambar 2.9. Papan C6713 DSK dengan bagian-bagiannya [1].



Gambar 2.10. Blok diagram papan C6713 DSK [1].

TMS320 C6713 DSK memiliki beberapa modul input dan output yang diantaranya adalah:

a. LINE IN dan LINE OUT

LINE IN merupakan modul input yang mengubah sinyal analog menjadi sinyal diskrit dengan frekuensi cuplik tertentu. Sedangkan LINE OUT merupakan modul output yang mengubah sinyal diskrit menjadi sinyal kontinu yang terinterpolasi. Jenis proses interpolasi sinyal yang digunakan adalah *Zero Order Hold*. Frekuensi cuplik yang digunakan pada modul LINE IN maupun LINE OUT cukup bervariasi, kisaran nilai dimulai dari 8 kHz, 32 kHz, 44,1 kHz hingga yang tertinggi yakni 96 kHz. Penggunaan kedua modul ini

(termasuk pula modul MIC IN sebagai modul input dan HP OUT sebagai modul output) melibatkan *codec* AIC23.

b. LED dan DIP Switch

Modul LED merupakan modul output yang sinyal outputnya direpresentasikan dalam bentuk penyalan LED. Sedangkan DIP Switch merupakan modul input yang sinyal inputnya direpresentasikan dalam bentuk perubahan tegangan hasil dari penekanan (*switching*) tombol DIP Switch.

c. JTAG (*Joint Team Action Group*)

Modul ini memungkinkan terjadinya komunikasi antara DSK dengan komputer. Modul ini merupakan modul input sekaligus modul output. Melalui *port* USB (*Universal Serial Bus*), pertukaran data dan kontrol serta pengawasan proses pengekseskuan program dapat dilakukan antara DSK dengan komputer. Proses pertukaran data secara *real time* antara DSK dengan komputer dengan menggunakan JTAG dikenal dengan nama RTDX (*Real Time Data Exchange*). Proses RTDX dilakukan dengan cukup intensif karena proses ini memerlukan *timing* yang teratur dan tepat antara komputer dengan DSK serta sinkronisasi diantara keduanya. Jika dibandingkan dengan kecepatan cuplik pada modul LINE IN dan modul LINE OUT, maka kecepatan rata-rata dari proses pertukaran sebuah data terjadi lebih lambat.

## BAB 3 RANCANG BANGUN

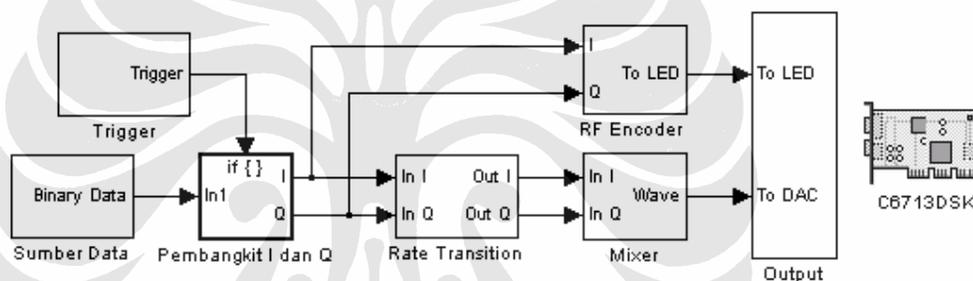
### 3.1 PENJELASAN MODEL PENGIRIM CDMA SECARA UMUM

Bab ini menjelaskan setiap blok yang menyusun model pengirim CDMA dan model simulasi yang digunakan. Sub bab 3.2 hingga sub bab 3.8 menjelaskan setiap blok yang menyusun sebuah pengirim CDMA yang dibangun. Perancangan model pengirim CDMA dilakukan dengan menggunakan Matlab 7.4.0 Simulink 6.0. Model pengirim CDMA secara keseluruhan ditunjukkan oleh Gambar 3.2. Sub blok "From RTDX" mendapatkan data dalam bentuk *frame* (yang terdiri satu paket *sample*) dari komputer melalui fasilitas RTDX. Pada proses selanjutnya, data *frame* tersebut dipecah sampai seukuran *frame* yang terdiri dari dua *sample* dengan menggunakan sub blok "Buffer" dan sub blok "Unbuffer". Data *frame* berukuran lebih kecil tersebut ditransfer menuju blok "Pembangkit I dan Q" untuk diproses menjadi parameter I dan Q. Gambar 3.10 menunjukkan arah aliran data pada setiap blok jika data yang dikirim merupakan pengulangan dari vektor [1 0].

```
.....
### Initial pass through model to cache user defined code
..
### Caching model source code
.....
### Writing header file pengirim_CDMA_types.h
### Writing header file pengirim_CDMA.h
### Writing header file pengirim_CDMA_private.h
.
### Writing source file pengirim_CDMA.c
### Writing source file pengirim_CDMA_data.c
.|
### Writing header file rt_nonfinite.h
### Writing source file rt_nonfinite.c
### Writing source file pengirim_CDMA_main.c
.
### TLC code generation complete.
### Generating the DSP/BIOS configuration file ...
### Creating project in Code Composer Studio(tm)...
### Building Code Composer Studio(tm) project...
### Build complete
### Downloading COFF file
### Downloaded: pengirim_CDMA.out
>>
```

Gambar 3.1 Beberapa pesan yang ditampilkan saat proses pembuatan *file* .OUT.

Blok "Rate Transition" digunakan untuk memfasilitasi perpindahan data antara dua blok dengan tingkat kecepatan proses data yang berbeda. Parameter I dan Q ditransfer menuju blok "Mixer" untuk dimodulasikan dengan gelombang pembawa diskrit. Hasil modulasi berupa sinyal analog yang dipancarkan melalui blok "DAC" (yang merupakan representasi dari *port* LINE OUT) dengan frekuensi cuplik sebesar 32 kHz. Blok "C6713DSK" (yang merupakan bagian dari model pemancar CDMA) berfungsi untuk menginisialisasi jenis perangkat keras yang menjadi target pemrograman atau instalasi algoritma model pengirim tersebut. Sebelum proses instalasi dilakukan, pembangkitan kode-kode bahasa C atau C++ dilakukan secara otomatis dari model pengirim yang telah dirancang. Pesan-pesan yang ditampilkan selama proses pembangkitan kode-kode tersebut ditunjukkan oleh Gambar 3.1.

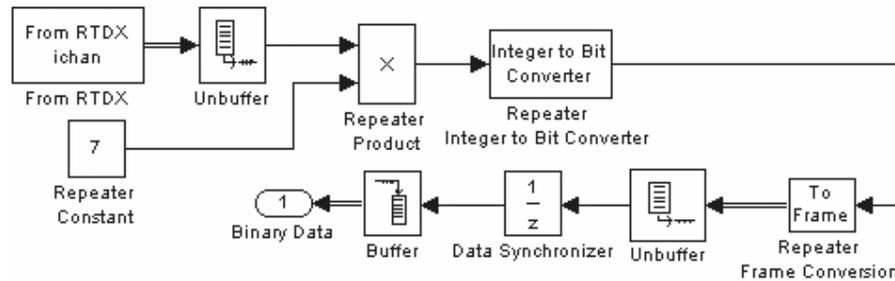


Gambar 3.2. Model pengirim CDMA.

### 3.2 BLOK "SUMBER DATA"

Blok "Sumber Data" merupakan blok penghasil data biner yang beratribut *frame* (data *frame*). Sebuah data *frame* terdiri dari dua *sample* bit. Oleh karena itu, penggunaan sub blok yang berfungsi sebagai *serial to parallel* tidak diperlukan karena data sudah diterima dalam bentuk *frame*. Simbol anak panah tunggal yang terlihat pada model menunjukkan arus data yang beratribut *sample*. Sedangkan gambar anak panah ganda menunjukkan arus data yang beratribut *frame*. Bagian-bagian blok ini ditunjukkan oleh Gambar 3.3 di bawah ini. Selain berfungsi sebagai blok penerima data biner dari komputer, blok ini berfungsi pula untuk mengulang data biner tersebut sedemikian sehingga satu bit biner diulang menjadi N bit biner (N bit *sample*) dengan N setara dengan jumlah bit kode PN dalam satu

urutan (satu periode urutan). Oleh karena itu, blok ini dapat melakukan proses pengulangan bit (*repeating process*) terhadap setiap bit data yang diterima.



Gambar 3.3. Blok "Sumber Data".

### 3.2.1 Sub Blok "From RTDX"

Sub blok ini berfungsi sebagai media transmisi data antara C6713 DSK dengan komputer dengan menggunakan fasilitas RTDX. Keluaran dari sub blok ini merupakan satu paket *sample* yang berbentuk sebuah *frame*. Satu paket *sample* dapat terdiri dari ratusan *sample* yang akan ditransmisikan dengan menggunakan C6713 DSK yang telah terinstalasi model pengirim. Sistem pengiriman data dilakukan secara "queue" yang memproses *sample* dengan sistem *FIFO* (*First In First Out*) setelah jumlah *sample* yang terkumpul sebanyak satu paket. *M-File* "driver2.m" digunakan untuk mengendalikan proses transfer data antara komputer dengan C6713 DSK. *File* ini mengaktifkan fasilitas RTDX antara komputer (Matlab 7.4.0) dengan C6713 DSK melalui komputer (Matlab 7.4.0). Isi dari *M-file* tersebut terlampir pada skripsi ini.

### 3.2.2 Sub Blok "Unbuffer"

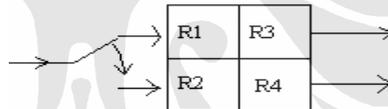
Sub blok ini berfungsi untuk mengubah data *frame* sebagai masukan menjadi data *sample* sebagai keluaran dengan *rate* keluaran yang lebih tinggi dari *rate* masukan. Penggunaan sub blok ini mengakibatkan terjadinya *delay* antara *frame* masukan dengan *sample* keluaran yang pertama sebesar jumlah *sample* dalam sebuah *frame* masukan dikali dengan nilai *sample time* yang digunakan pada sub blok ini.

### 3.2.3 Sub Blok "Buffer"

Sub blok "Buffer" merupakan sub blok untuk memecah alur sinyal atau bit menjadi 2 alur dan dikeluarkan secara serempak. Sub blok "Buffer" merepresentasikan rangkaian *serial to parallel* dengan rasio jumlah saluran bit

masuk dengan bit keluaran sebesar 1 : 2. Oleh karena itu bit *rate* sinyal keluaran adalah setengah kali bit *rate* sinyal masukan. Proses tersebut terdiri dari 2 bagian besar yaitu proses cuplik dan tahan serta proses pemancaran bit. Masukan dari sub blok "Buffer" yaitu sebuah bit dalam bentuk *sample* dan keluaran sub blok ini adalah dua buah bit yang berbentuk *frame*. Penggunaan sub blok ini mengakibatkan delay sebesar *sample time* dari *sample* masukan dikali dengan jumlah *sample* dalam sebuah *frame* keluaran (kapasitas registernya).

Sub blok "Buffer" dapat dideskripsikan sebagai rangkaian yang memiliki empat register dan sebuah *switch* yang bergerak bolak-balik dengan periode sebesar periode cupliknya (ditunjukkan Gambar 3.4). Deskripsi proses kerja dari sub blok ini ditunjukkan Tabel 3.1. Register R1 dan R2 berfungsi menampung bit yang dicuplik, sedangkan register R3 dan R4 berfungsi menahan dua bit yang sedang dipancarkan selama dua periode cuplik (proses tahan). Agar kedua bit dapat dipancarkan, maka bit-bit tersebut digeser dari register R1 dan R2 menuju register R3 dan R4.



Gambar 3.4. Ilustrasi dari sub blok "Buffer".

Tabel 3.1. Deskripsi cara kerja sub blok "Buffer".

Pemancaran bit	R4	I	I	B2	B4	B6	B8
	R3	I	I	B1	B3	B5	B7
Proses Tahan	R2	I	B2	B2	B4	B6	B8
	R1	B1	B1	B3	B3	B5	B5
Waktu cuplik k =		1	3	5	7	9	

Simbol "B1", "B2", "B3" dan seterusnya menunjukkan urutan bit yang telah dicuplik. Sedangkan simbol "I" menunjukkan *initial condition* dari register. Saat bit "B2", bit "B1" sudah tersimpan pada register R1. Saat bit "B2" tersimpan pada register R2, bit "B1" sudah tersimpan pada register R1 dan pada periode cuplik selanjutnya, kedua bit tersebut dipindahkan menuju register R3 dan R4 untuk dipancarkan selama 2 periode cuplik. Selama bit "B1" dan "B2"

dipancarkan, bit "B3" dan "B4" dicuplik berurutan dan masing-masing bit tersebut mengisi register R1 dan R2.

#### **3.2.4 Sub Blok "Repeater Constant"**

Sub-sub blok yang terlibat dalam proses perulangan yang dilakukan terhadap setiap bit data memiliki nama blok yang diawali dengan kata "Repeater". Sub blok "Repeater Constant" merupakan salah satu blok yang terlibat dalam proses perulangan tersebut. Sub blok ini berfungsi untuk menentukan jumlah bit hasil pengulangan dari satu bit data. Sub blok ini merupakan sub blok yang dapat menghasilkan keluaran bilangan *real* yang konstan. Namun agar dapat digunakan untuk menentukan jumlah bit hasil pengulangan satu bit data, keluarannya harus merupakan bilangan bulat. Pada Gambar 3.3 di atas, blok tersebut menghasilkan keluaran bilangan bulat bernilai tujuh yang berarti bilangan biner yang bernilai 1 1 1 atau bilangan biner selebar tiga bit. Oleh karena itu, pengulangan dari satu bit data menghasilkan tiga bit *sample*.

#### **3.2.5 Sub Blok "Repeater Product"**

Sub blok ini berfungsi untuk menghasilkan bilangan bulat yang dapat mewakili nilai setiap bit data yang diulang. Jika bit data yang diulang bernilai nol maka sub blok ini akan menghasilkan keluaran nol. Jika bit data yang diulang bernilai satu maka sub blok ini akan melewatkan keluaran dari sub blok "Repeater Constant".

#### **3.2.6 Sub Blok "Repeater Integer to Bit Converter"**

Sub blok ini berfungsi untuk mengubah setiap nilai *integer* (bilangan bulat) yang merupakan keluaran sub blok "Repeater Product" menjadi bilangan biner dengan lebar bit sebesar lebar bit dari satu urutan (sekuens) kode PN.

#### **3.2.7 Sub Blok "Repeater Frame Conversion"**

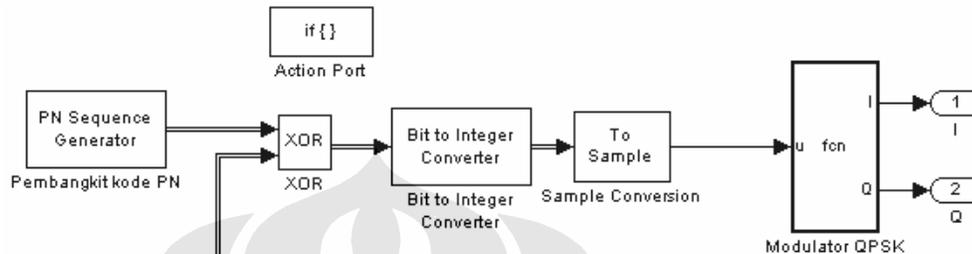
Sub blok ini berfungsi untuk memberikan atribut *frame* kepada bilangan biner yang terdiri dari N *sample* yang juga merupakan keluaran sub blok sebelumnya.

#### **3.2.8 Sub Blok "Data Synchronizer"**

Sub blok ini berfungsi untuk mensinkronkan waktu pengiriman data dengan waktu kerja dari blok "Pembangkit I dan Q" dan waktu kerja blok "Trigger".

### 3.3 BLOK "PEMBANGKIT I DAN Q"

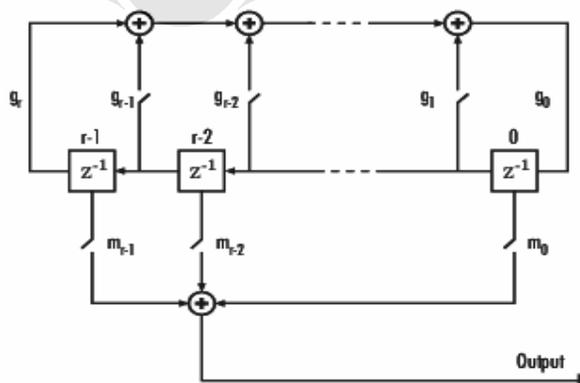
Blok "Pembangkit I dan Q" merupakan blok yang berfungsi untuk mengubah data biner menjadi data fasa yang dapat dimodulasi pada blok "Mixer" untuk mendapatkan bentuk sinyal (gelombang) CDMA sebenarnya yang dapat ditransmisikan. Bagian-bagian dari blok ini ditunjukkan oleh Gambar 3.5.



Gambar 3.5. Blok "Pembangkit I dan Q".

#### 3.3.1 Sub Blok "Pembangkit kode PN" dan Sub Blok "XOR"

Proses pembangkitan kode PN ditunjukkan oleh Gambar 3.6. Sub blok "Pembangkit kode PN" dari Matlab Simulink memiliki 2 parameter penting yaitu *initial state* dan *polynomial generator*. *Initial states* merupakan nilai awal dari isi register  $Z^{-1}$ . Sedangkan *polynomial generator* merupakan nilai dari  $g$  (suatu nilai yang menunjukkan ada atau tidaknya umpan balik dari setiap register). Nilai  $g_0$  tidak boleh bernilai 0. Besarnya *degree* dari polinomial sebesar jumlah anggota vektor *initial states* dikurangi satu. Bit atau kode yang merupakan output dari sub sub blok "pembangkit kode PN" berulang setiap  $2^n - 1$  (dengan  $n$  adalah *degree* dari polinomial yang dimaksud). Sub blok "pembangkit kode PN" merupakan bagian dari rangkaian penyebar.



Gambar 3.6. Proses pembangkitan kode PN [4].

Sub blok "XOR" adalah sub blok untuk mengubah bit-bit data menjadi sinyal *chip* (kumpulan bit-bit yang adalah kode PN). Sub blok tersebut merupakan operator XOR yang sekaligus merupakan salah satu bagian dari rangkaian penyebar. Rangkaian penyebar memiliki bit *chip* keluaran yang merupakan hasil operasi XOR antara bit *chip* masukan yang berasal dari sub sub blok "pembangkit kode PN" dengan bit data. Oleh karena itu sinyal CDMA merupakan sinyal dengan pengulangan kumpulan bit *chip* setiap satu atau beberapa periode bit data.

### 3.3.2 Sub Blok "Sample Conversion"

Sub blok ini merupakan salah satu sub blok penunjang yang berfungsi untuk mengubah atau menyesuaikan atribut data keluaran sub blok "Bit to Integer Converter" (yang memiliki kemungkinan beratribut *frame*) menjadi data yang beratribut *sample*. Hanya data yang beratribut *sample* yang dapat diproses oleh sub blok "Modulator QPSK".

### 3.3.3 Sub Blok "Modulator QPSK"

Modulator yang dipakai dalam proses simulasi adalah modulator *QPSK*. Dengan menggunakan sub blok "Modulator QPSK", dapat dimodelkan sebuah modulator *QPSK*. Algoritma yang digunakan pada sub blok "Modulator QPSK" yaitu:

```
function [I,Q] = fcn(u)
M=(fix(u/2))-0.5;
L=(mod(u,2))-0.5;
Q=-double(0.7071*sign(M));
I=-double(0.7071*sign(L));
```

Algoritma di atas memetakan parameter *u* yang merupakan bilangan bulat (0 – 3) ke parameter I dan Q dengan relasi pemetaan yang ditunjukkan Tabel 2.1.

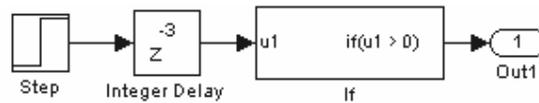
### 3.3.4 Sub Blok "Bit to Integer Converter"

Sub blok ini merupakan salah satu sub blok penunjang yang mengubah keluaran 2 bit biner yang berbentuk *frame* dari sub blok sebelumnya menjadi sebuah bilangan bulat dengan kemungkinan nilai mulai dari nol hingga tiga.

## 3.4 BLOK "TRIGGER"

Blok ini berfungsi untuk memicu kerja blok "Pembangkit I dan Q" pada saat data (*frame*) yang berasal dari blok "Sumber Data" telah dikirimkan menuju blok "Pembangkit I dan Q". Blok ini berfungsi untuk mensinkronkan waktu

pengiriman data yang menuju blok "Pembangkit I dan Q" dengan waktu kerjanya sendiri. Bagian-bagian dari blok "Trigger" ditunjukkan Gambar 3.7.



Gambar 3.7. Blok "Trigger".

### 3.4.1 Sub Blok "Step" dan Sub Blok "Integer Delay"

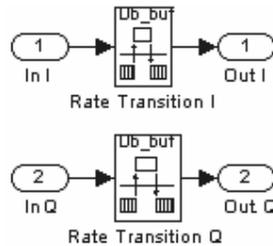
Kedua sub blok ini berfungsi sebagai pembangkit sinyal pemicu (*triggered time*) yang mengaktivasi blok "If". Waktu picu yang digunakan oleh sub blok "Step" disinkronkan (d disesuaikan) dengan periode pengiriman data dari blok "Sumber Data" sedemikian sehingga data yang dikirimkan oleh blok "Sumber Data" dapat diproses tepat waktu oleh blok "Pembangkit I dan Q". Sub blok "Integer Delay" berfungsi untuk mensinkronkan waktu picu yang dihasilkan oleh sub blok "Step" dengan waktu pengiriman (penyampaian) data. Waktu picu yang digunakan sebesar jumlah *sample* per paket dikali dengan waktu cuplik sub blok "From RTDX".

### 3.4.2 Sub Blok "If"

Sub blok ini merupakan sub blok pemicu atau sub blok yang mengaktivasi kerja blok "Pembangkit I dan Q". Sub blok ini menggunakan sinyal kontrol untuk memicu kerja blok "Pembangkit I dan Q". Sinyal kontrol akan dibangkitkan ketika keluaran dari sub blok "Integer Delay" bernilai satu.

## 3.5 BLOK "RATE TRANSITION"

Blok ini berfungsi untuk mengatur transfer data antara blok "Pembangkit I dan Q" (yang memproses data pada tingkat kecepatan yang rendah) dengan blok "Mixer" yang memproses data pada tingkat kecepatan yang lebih tinggi. Blok ini memiliki dua sub blok yaitu sub blok "Rate Transition I" dan sub blok "Rate Transition Q". Sub blok "Rate Transition I" berfungsi untuk mengatur perpindahan data (parameter I) dari blok "Pembangkit I dan Q" menuju blok "Mixer". Sub blok "Rate Transition Q" berfungsi untuk mengatur perpindahan data (parameter Q) dari blok "Pembangkit I dan Q" menuju blok "Mixer". Bagian-bagian dari blok ini ditunjukkan oleh Gambar 3.8.



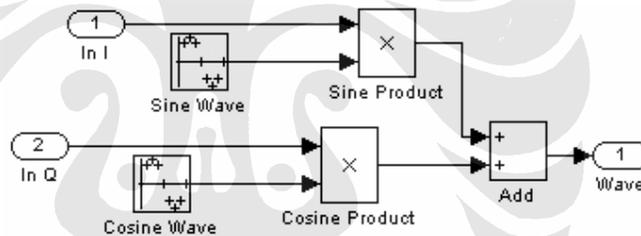
Gambar 3.8. Blok “Rate Transition”.

### 3.6 BLOK ”MIXER”

Blok ini berfungsi untuk menumpangkan parameter I dan Q ke dalam parameter gelombang pembawa dengan hubungan yang dapat dinyatakan oleh persamaan di bawah ini:

$$s(t) = I \sin(2\pi ft) + Q \cos(2\pi ft) \quad (3.1)$$

dengan  $s(t)$  merupakan sinyal yang ditransmisikan. Gambar 3.9 menunjukkan blok diagram dari bagian-bagian penyusun blok ”Mixer”.



Gambar 3.9. Blok ”Mixer”.

#### 3.6.1 Sub Blok ”Sine Wave” dan Sub Blok ”Cosine Wave”

Kedua sub blok ini berfungsi sebagai pembangkit gelombang pembawa dalam bentuk diskrit (osilator). Sub blok ”Sine Wave” berfungsi untuk membangkitkan gelombang sinusoidal diskrit. Sub blok ”Cosine Wave” berfungsi untuk membangkitkan gelombang sinusoidal diskrit yang fasanya bergeser 90 derajat. Periode cuplik (*sample time*) yang digunakan di kedua sub blok ini sama dengan periode cuplik dari blok ”DAC”.

#### 3.6.2 Sub Blok ”Sine Product” dan Sub Blok ”Cosine Product”

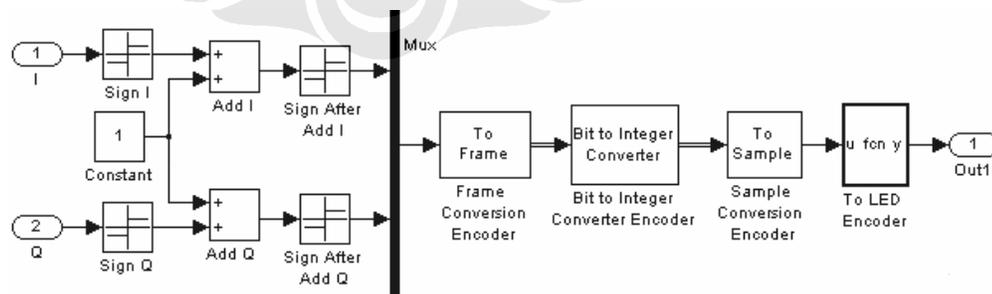
Sub blok ”Sine Product” dan sub blok ”Cosine Product” berfungsi untuk memodulasikan parameter I dan Q ke dalam gelombang pembawa (diskrit) dengan cara mengalikan setiap parameter tersebut dengan tipe gelombang pembawa yang sesuai.

### 3.6.3 Sub Blok "Add"

Sub blok "Add" menjumlahkan hasil perkalian (keluaran sub blok "Sine Product" dan sub blok "Cosine Product") satu dengan yang lainnya menjadi persamaan yang menyatakan sinyal yang ditransmisikan. Hubungan matematis yang menyatakan proses modulasi tersebut ditunjukkan oleh persamaan 3.1.

### 3.7 BLOK "RF ENCODER"

Blok ini berfungsi sebagai penunjang komunikasi (penghubung) antara C6713 DSK dengan rangkaian modulator analog yang bekerja pada frekuensi gelombang radio. Modulator analog yang dimaksud merupakan modulator analog yang digunakan untuk sistem modulasi QPSK. Blok ini menggunakan jalur yang menghubungkan LED dengan prosesor untuk mengirimkan sinyal yang berupa sinyal atau parameter I dan Q dari blok pengirim CDMA (yang terprogram di dalam prosesor DSP) menuju rangkaian modulator analog QPSK. Oleh karena itu, blok ini memungkinkan pengirim CDMA yang dibangun (yang frekuensi sinyal keluarannya sangat rendah) dapat di-*upgrade* menjadi pengirim CDMA yang menghasilkan sinyal CDMA yang memiliki frekuensi gelombang radio (*Radio Frequency*). Gambar 3.12 menunjukkan diagram rangkaian modulator analog (menggunakan osilator berfrekuensi radio) yang dapat digunakan dari penggunaan blok ini dan LED sebagai keluaran. Tabel 3.2 menunjukkan keluaran dari beberapa sub blok penyusun blok "RF Encoder" pada seluruh kemungkinan masukan blok tersebut. Parameter I direpresentasikan oleh LED 2 dan LED 4 dan nilai parameter Q direpresentasikan oleh LED 1 dan LED 3.



Gambar 3.10. Blok "RF Encoder"

### 3.7.1 Sub Blok "Sign I" dan Sub Blok "Sign Q"

Sub blok "Sign I" dan sub blok "Sign Q" berfungsi untuk menentukan tanda (positif atau negatif) dari setiap nilai parameter I dan Q yang diterima. Keluaran dari kedua sub blok adalah nilai 1 (jika masukannya lebih besar dari nol), nilai -1 (jika masukannya lebih kecil dari nol) atau nilai nol (jika masukannya sama dengan nol).

### 3.7.2 Sub Blok "Add I", Sub Blok "Add Q", dan Sub Blok "Constant"

Ketiga sub blok ini merupakan sub-sub blok yang berfungsi mengubah nilai dari parameter I dan Q dari nilai positif dan negatif menjadi nilai positif dan nilai nol.

### 3.7.3 Sub Blok "Sign After Add I" dan Sub Blok "Sign After Add Q"

Sub blok "Sign After Add I" dan sub blok "Sign After Add Q" berfungsi untuk mengubah sinyal keluaran dari sub blok "Add I" dan sub blok "Add Q" yang kemungkinan nilainya adalah dua atau nol menjadi sinyal biner yang kemungkinan nilainya adalah nol atau satu.

### 3.7.4 Sub Blok "Mux" dan Sub Blok "Frame Conversion Encoder"

Kedua sub blok tersebut berfungsi untuk menggabungkan sinyal-sinyal biner keluaran sub-sub blok sebelumnya yang beratribut *sample* menjadi sebuah sinyal yang beratribut *frame*.

### 3.7.5 Sub Blok "Bit to Integer Converter Encoder" dan Sub Blok "Sample Conversion Encoder"

Sub blok "Bit to Integer Converter Encoder" berfungsi untuk mengubah sinyal biner yang beratribut *frame* menjadi sebuah sinyal *integer* (bilangan bulat). Atribut dari sinyal keluaran sub blok ini belum tentu beratribut *sample*. Sub blok "Sample Conversion Encoder" berfungsi untuk memberikan atribut *sample* kepada sinyal keluaran dari sub blok "Bit to Integer Converter Encoder".

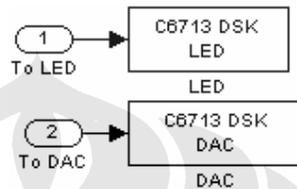
### 3.7.6 Sub Blok "To LED Encoder"

Sub blok ini berfungsi untuk mengubah sinyal *integer* yang memiliki kemungkinan nilai 0, 1, 2 atau 3 menjadi sinyal *integer* yang memiliki kemungkinan nilai 3, 6, 9 atau 12. Sinyal keluaran dari sub blok ini akan mengaktifkan LED dari C6713 DSK. Algoritma yang digunakan pada sub blok tersebut yaitu:

```
function y = fcn(u)
u=u+1;
y=uint8(u*3);
```

### 3.8 BLOK “OUTPUT”

Blok ini merupakan representasi dari output C6713 DSK. Blok ini terdiri dari sub blok “LED” dan sub blok “DAC”. Gambar 3.11 menunjukkan blok “Output”.



Gambar 3.11. Blok “Output”

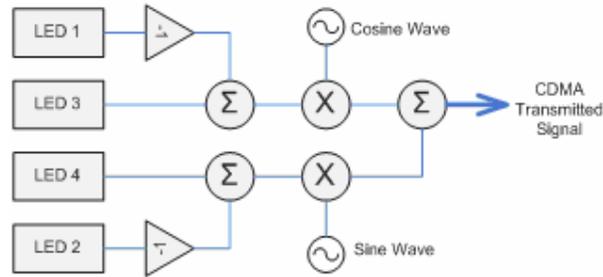
#### 3.8.1 Sub Blok “LED”

Sub blok ini berfungsi untuk mengendalikan penyalaaan C6713 DSK LED yang berjumlah empat LED. Masukan dari sub blok ini adalah bilangan bulat (*integer*) yang nilainya dimulai dari 0 hingga 15. Setiap bit satu dari bentuk biner bilangan bulat ini mewakili penyalaaan setiap LED. Sebaliknya, setiap bit nol dari bentuk biner bilangan bulat ini mewakili pemadaman setiap LED. Jika masukan sub blok tersebut adalah angka nol, maka tidak ada LED yang menyala. Jika masukan sub blok tersebut adalah angka 15, maka semua LED menyala.

#### 3.8.2 Sub Blok “DAC”

Sub blok ini berfungsi untuk menentukan sinyal keluaran *port* LINE OUT yang terdapat pada papan elektronik C6713 DSK. Beberapa parameter penting dari sub blok ini yang dapat dikonfigurasi yaitu:

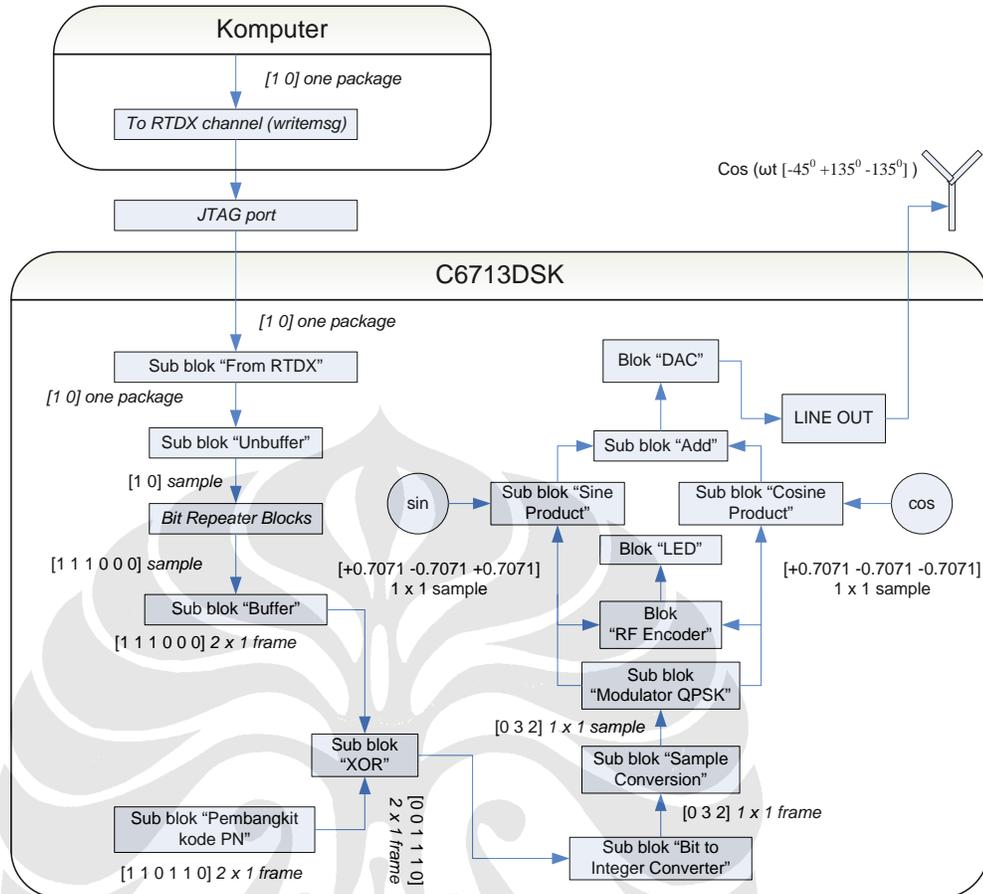
- 1) Parameter “Word length” yang berfungsi untuk menentukan tingkat akurasi dari keluaran DAC,
- 2) Parameter “Sampling rate” yang berfungsi untuk menentukan periode cuplik (periode keluaran) dari DAC,
- 3) Parameter “Scaling” dan parameter “Overflow mode”.



Gambar 3.12. Rangkaian modulator analog yang dapat dipakai dari penggunaan keluaran LED

Tabel 3.2. Keluaran dari beberapa sub blok penyusun blok "RF Encoder" pada seluruh kemungkinan masukan

Keadaan Parameter I / Q	0.7071 / 0.7071	0.7071 / -0.7071	-0.7071 / 0.7071	-0.7071 / -0.7071
Sign I	1	1	-1	-1
Sign Q	1	-1	1	-1
Add I	2	2	0	0
Add Q	2	0	2	0
Sign After Add I	1	1	0	0
Sign After Add Q	1	0	1	0
Mux	(1 1)	(1 0)	(0 1)	(0 0)
S. Conversion Encoder	3	2	1	0
To LED Encoder	12 = 1 1 0 0	9 = 1 0 0 1	6 = 0 1 1 0	3 = 0 0 1 1
LED yang menyala	4 dan 3	4 dan 1	3 dan 2	2 dan 1
LED yang padam	2 dan 1	2 dan 3	1 dan 4	4 dan 3



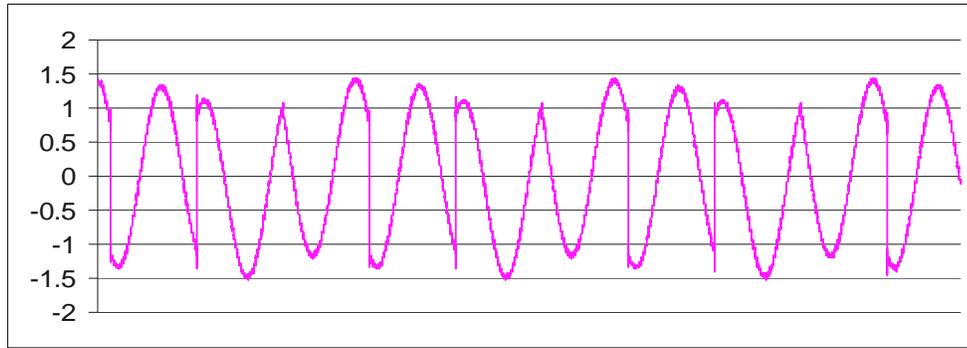
Gambar 3.13. Pemrosesan data pada model pengirim saat vektor data adalah pengulangan dari vektor data (1 0).

## BAB 4

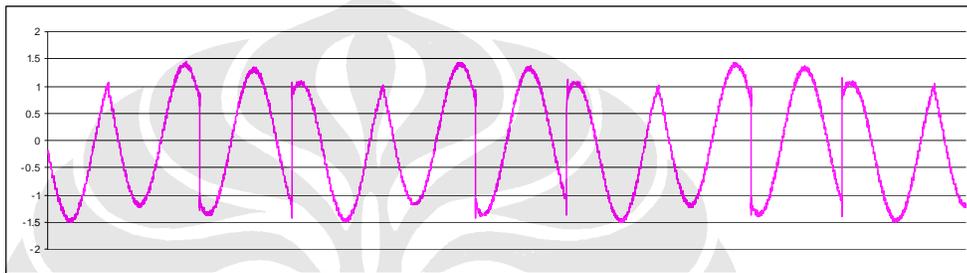
### UJI COBA DAN ANALISIS MODEL PENGIRIM CDMA

#### 4.1 PROSEDUR UJI COBA DAN PENGAMBILAN SAMPEL

Pengambilan sampel yang berupa bentuk gelombang dari sinyal keluaran dilakukan dengan menggunakan *digital storage oscilloscope*. Media disket digunakan untuk menyimpan sinyal keluaran yang ditangkap oleh osiloskop dalam bentuk *spreadsheet file* dengan format “.CSV”. *File* tersebut berisi sampel-sampel hasil pencuplikan yang dilakukan oleh osiloskop terhadap sinyal keluaran yang sebenarnya dan dapat dibaca oleh komputer dengan menggunakan perangkat lunak Microsoft Excel untuk membentuk kembali grafik atau bentuk sinyalnya. Jumlah sampel yang sebanyak 10000 sampel tersebut tidak memiliki interval waktu cuplik yang selalu sama untuk setiap sampel dengan sampel terdekatnya. Waktu cuplik yang digunakan oleh osiloskop yaitu 50 kHz, 100 kHz, dan 250 kHz. Bentuk sinyal keluaran yang ditampilkan oleh osiloskop dan bentuk sinyal keluaran hasil simulasi ditunjukkan oleh Gambar 4.1 hingga Gambar 4.18. Gambar 4.20 hingga Gambar 4.26 menunjukkan bentuk spektrumnya. Pengambilan sampel bentuk spektrum dari sampel sinyal keluaran yang sebenarnya masih menggunakan *spreadsheet file* yang sama yang berformat “.CSV”. *M-file* “*spectrum\_analyzer.m*” digunakan untuk menunjang terjadinya proses FFT terhadap sampel-sampel yang merupakan isi dari *spreadsheet file* tersebut. Sedangkan *M-file* “*direct\_spectrum\_analyzer.m*” digunakan untuk menunjang proses FFT terhadap sinyal keluaran hasil simulasi. *Rectangular windowing* digunakan pada proses FFT sinyal. Sinyal keluaran hasil simulasi sudah memiliki interval waktu cuplik yang konstan yaitu  $1 / 32000$  detik untuk setiap sampel dengan sampel terdekatnya. Simulasi diperlukan untuk mengetahui bentuk sinyal yang seharusnya terjadi. Sinyal keluaran hasil simulasi dibandingkan dengan sinyal keluaran yang sebenarnya.

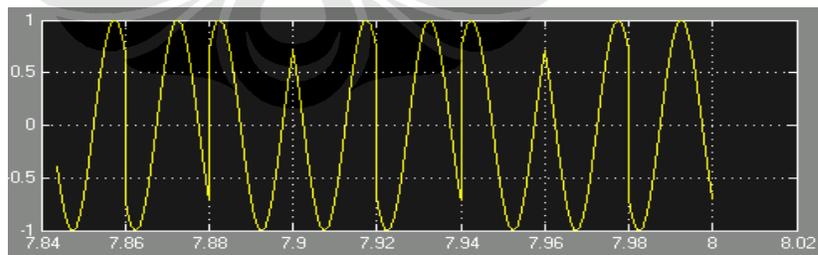


Gambar 4.1. Keluaran sebenarnya saat vektor data adalah pengulangan dari (1 0).

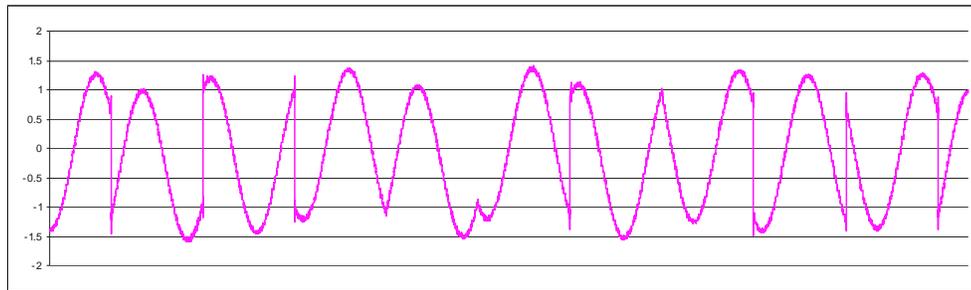


Gambar 4.2. Keluaran sebenarnya (dalam bingkai waktu yang berbeda) saat vektor data adalah pengulangan dari (1 0).

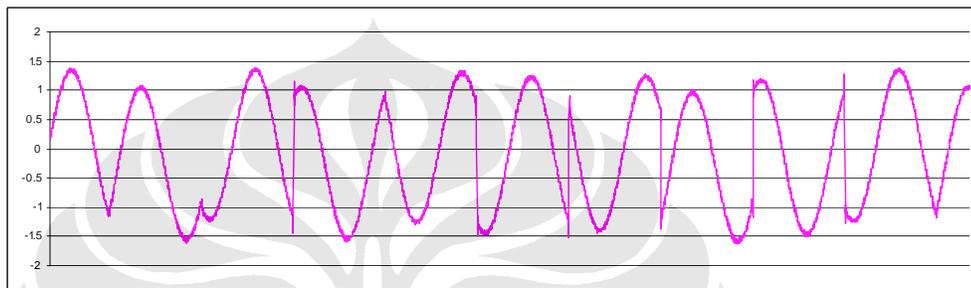
M-file "driver2.m" berfungsi sebagai salah satu fasilitas penunjang proses pengambilan sampel. Setelah data ditransfer dari komputer menuju DSK, setiap bit data diulang sebanyak dua kali di dalam blok "Sumber Data" karena lebar vektor kode PN sebesar tiga bit. Sebagai contohnya, jika kita ingin mengirimkan tiga bit data (contohnya adalah urutan (1 0 1) ) maka format vektor data setelah setiap bit diulang sebanyak dua kali yaitu (1 1 1 0 0 0 1 1 1).



Gambar 4.3. Keluaran blok "Mixer" (hasil simulasi) saat vektor data adalah pengulangan dari (1 0).

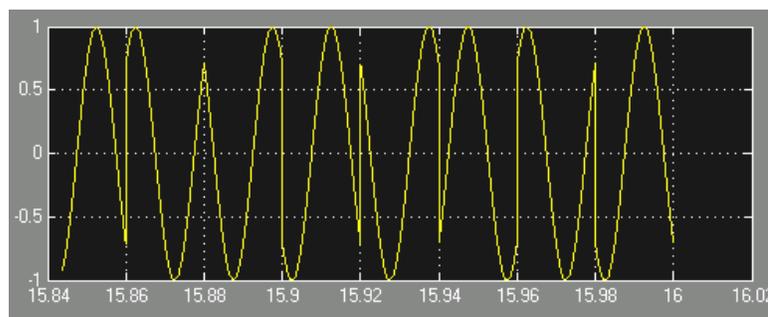


Gambar 4.4. Keluaran sebenarnya saat vektor data adalah pengulangan dari (0 1 1).

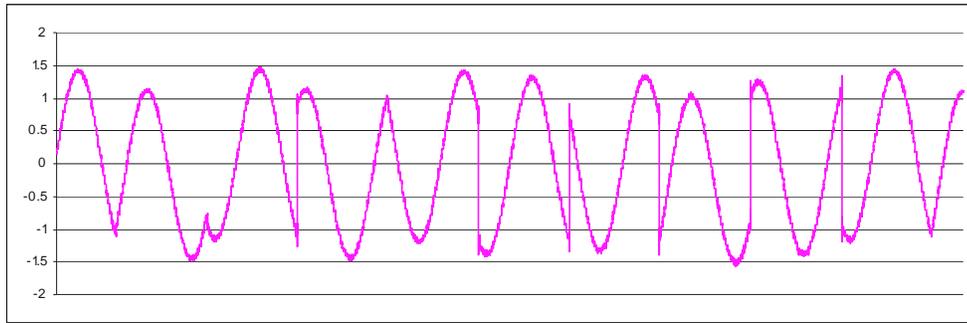


Gambar 4.5. Keluaran sebenarnya (dalam bingkai waktu yang berbeda) saat vektor data adalah pengulangan dari (0 1 1).

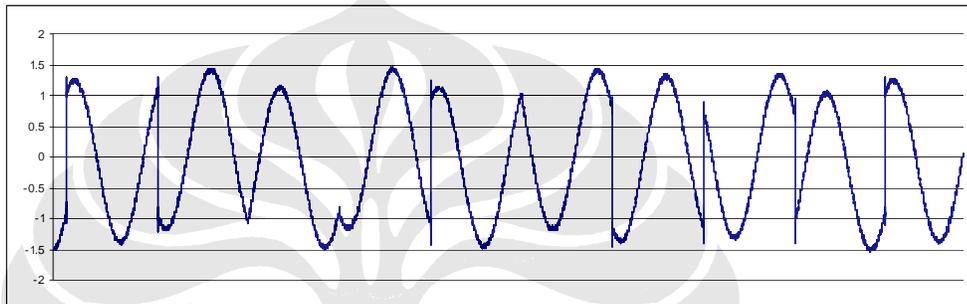
Pengulangan setiap bit data sebanyak dua kali dilakukan agar bit-bit data dapat diproses oleh sub blok "XOR" pada blok "Pembangkit I dan Q" dengan kecepatan yang lebih tinggi. *M-file* "driver2.m" mengakses data yang telah terulang dan tersimpan di dalam *workspace* untuk mentransfernya menuju memori C6713DSK melalui fasilitas RTDX. Penggunaan blok "Signal From Workspace" sebagai bagian dari model simulasi ditujukan untuk menggantikan fungsi sub blok "From RTDX" (pada model pengirim). Blok "Time Scope" merupakan blok yang digunakan untuk melihat keluaran dari blok "Mixer". Blok ini juga merupakan pengganti blok "DAC" pada model pengirim.



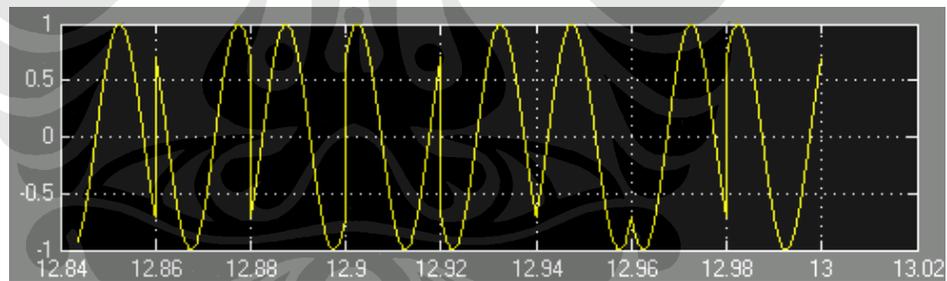
Gambar 4.6. Keluaran blok "Mixer" (hasil simulasi) saat vektor data adalah pengulangan dari (0 1 1).



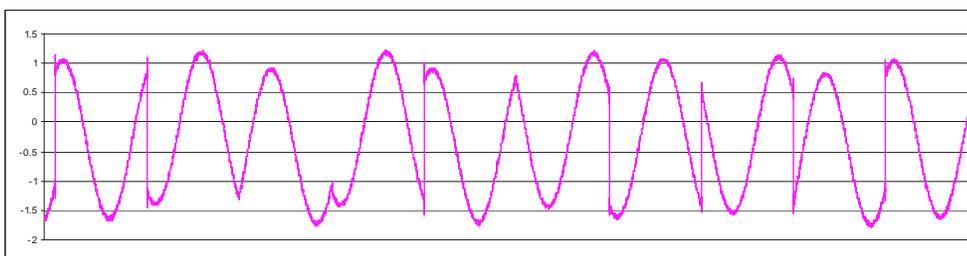
Gambar 4.7. Keluaran sebenarnya saat vektor data adalah pengulangan dari (1 0 1).



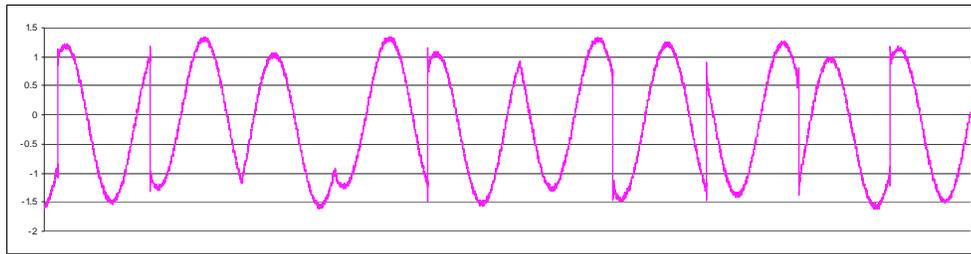
Gambar 4.8. Keluaran sebenarnya (dalam bingkai waktu yang berbeda) saat vektor data adalah pengulangan dari (1 0 1).



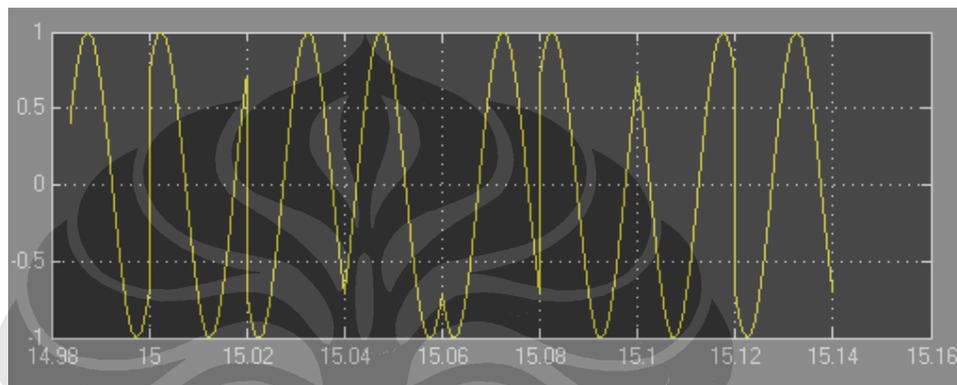
Gambar 4.9. Keluaran blok "Mixer" (hasil simulasi) saat vektor data adalah pengulangan dari (1 0 1).



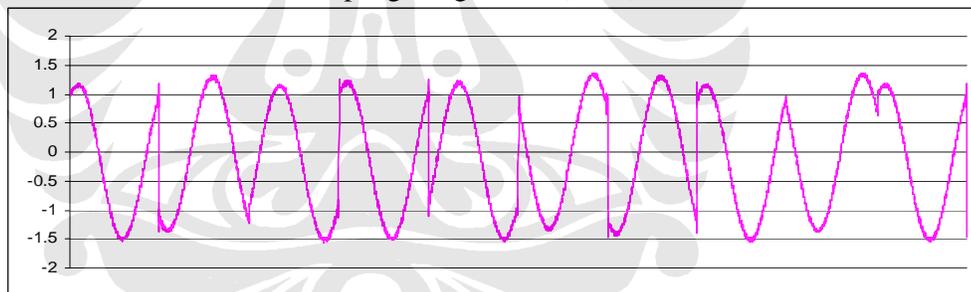
Gambar 4.10. Keluaran sebenarnya saat vektor data adalah pengulangan dari (1 1 0).



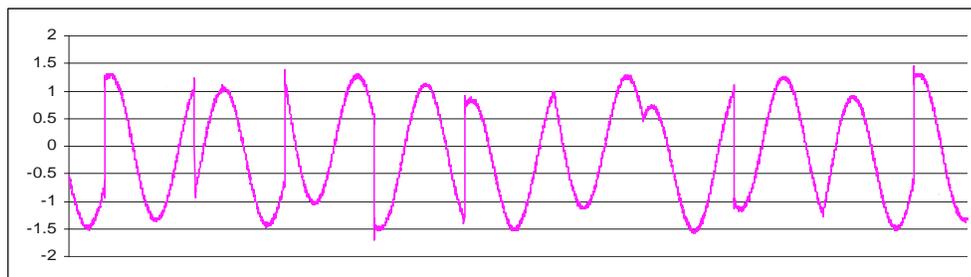
Gambar 4.11. Keluaran sebenarnya (dalam bingkai waktu yang berbeda) saat vektor data adalah pengulangan dari (1 1 0).



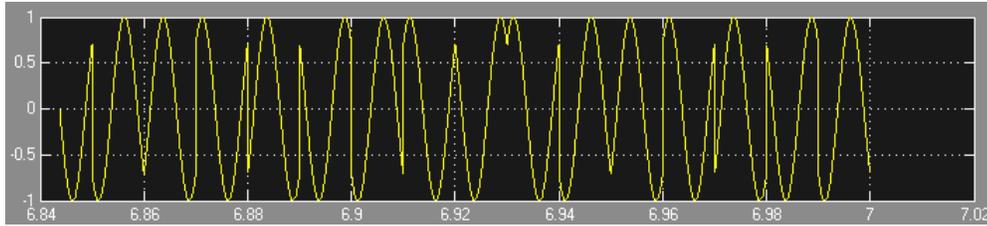
Gambar 4.12. Keluaran blok “Mixer” (hasil simulasi) saat vektor data adalah pengulangan dari (1 1 0).



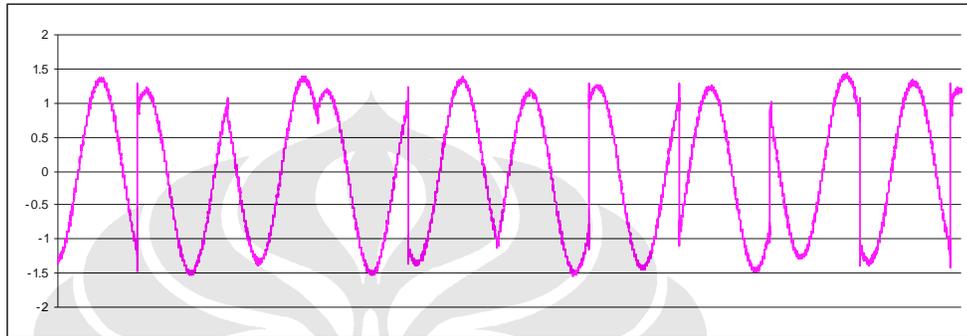
Gambar 4.13. Keluaran sebenarnya saat vektor data adalah pengulangan dari (0 1 0) dengan *data rate* tinggi dengan frekuensi pembawa = 100 Hz.



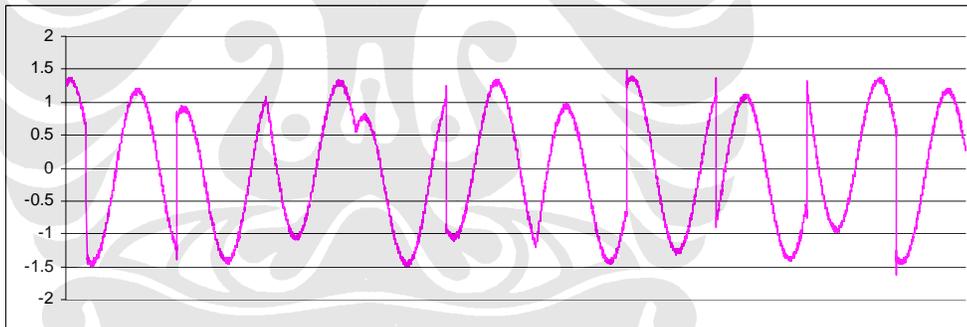
Gambar 4.14. Keluaran sebenarnya saat vektor data adalah pengulangan dari (0 1 0) dengan *data rate* rendah dengan frekuensi pembawa = 25 Hz.



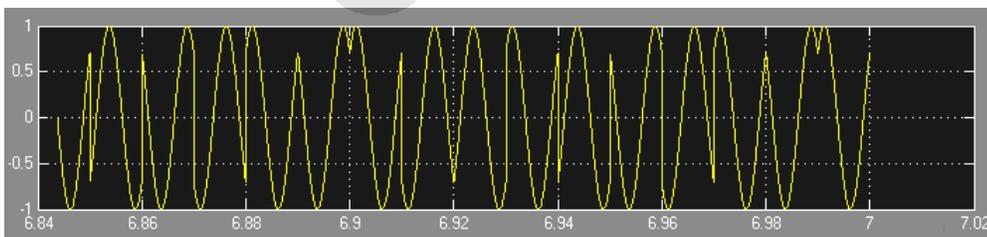
Gambar 4.15. Keluaran blok “Mixer” (hasil simulasi) saat vektor data adalah pengulangan dari (0 1 0).



Gambar 4.16. Keluaran sebenarnya saat vektor data adalah pengulangan dari (0 0 1) dengan *data rate* tinggi dengan frekuensi pembawa = 100 Hz.



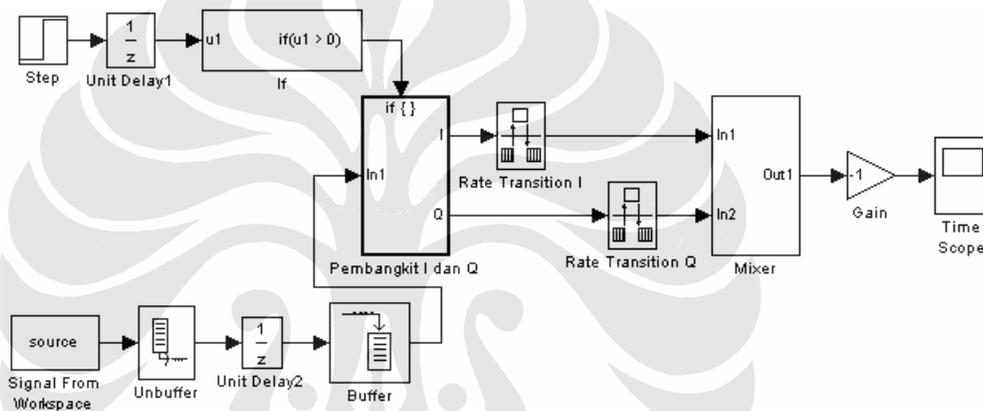
Gambar 4.17. Keluaran sebenarnya saat vektor data adalah pengulangan dari (0 0 1) dengan *data rate* rendah dengan frekuensi pembawa = 25 Hz.



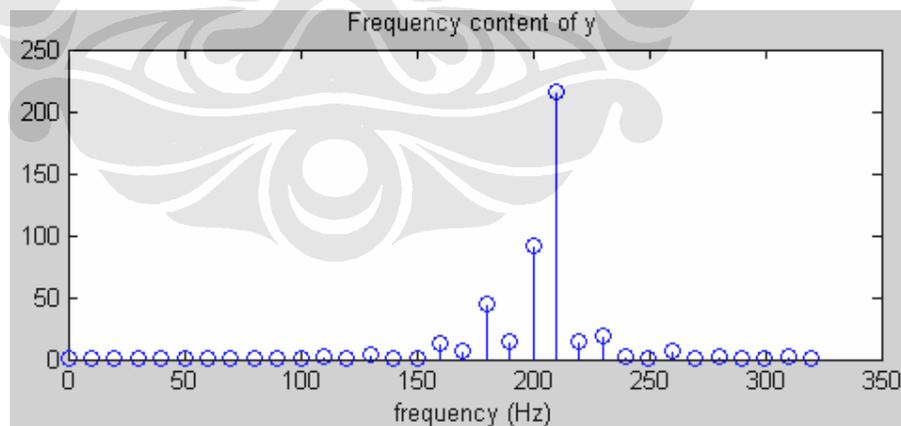
Gambar 4.18. Keluaran blok “Mixer” (hasil simulasi) saat vektor data adalah pengulangan dari (0 0 1).

Jumlah bit data yang dikirimkan dengan menggunakan fasilitas RTDX adalah sebanyak satu paket bit data (334 bit data). Data yang dikirimkan

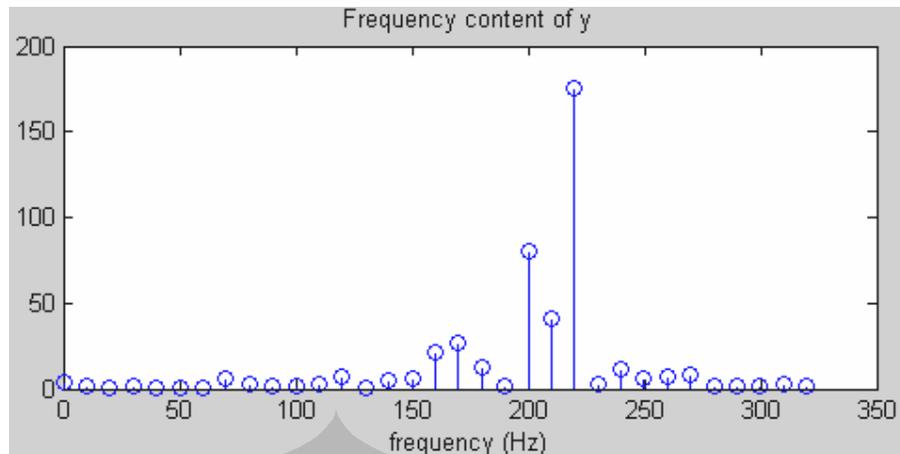
merupakan urutan atau deretan bit yang periodik Hal ini ditujukan untuk mempermudah dalam membandingkan antara keluaran hasil simulasi dengan keluaran sebenarnya. Penggunaan deretan bit yang periodik akan menghasilkan keluaran bentuk gelombang yang periodik pula. Proses penangkapan dan penyimpanan gambar yang ditampilkan layar osiloskop tidak dapat dilakukan dalam bingkai waktu yang diharapkan, namun proses tersebut terjadi dalam bingkai waktu yang acak. Oleh karena itu, penggunaan deretan atau urutan bit yang periodik (seperti urutan (1 0 1) ) akan memudahkan penulis untuk membandingkan kedua hasil keluaran tersebut. Model simulasi yang digunakan ditunjukkan oleh Gambar 4.19.



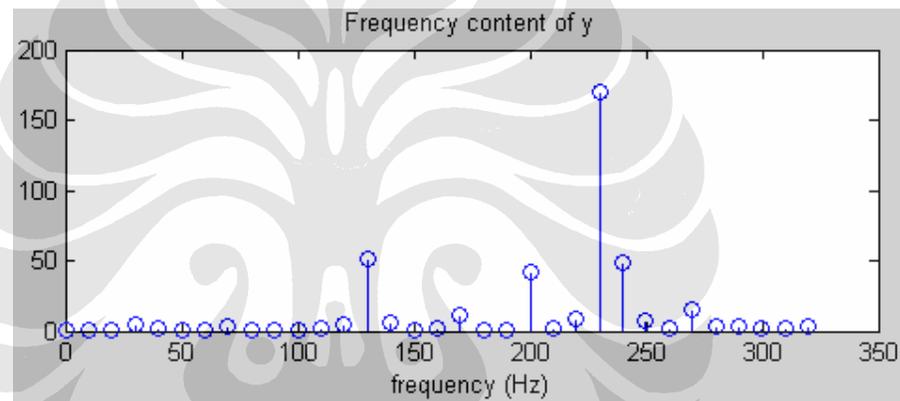
Gambar 4.19. Blok model simulasi.



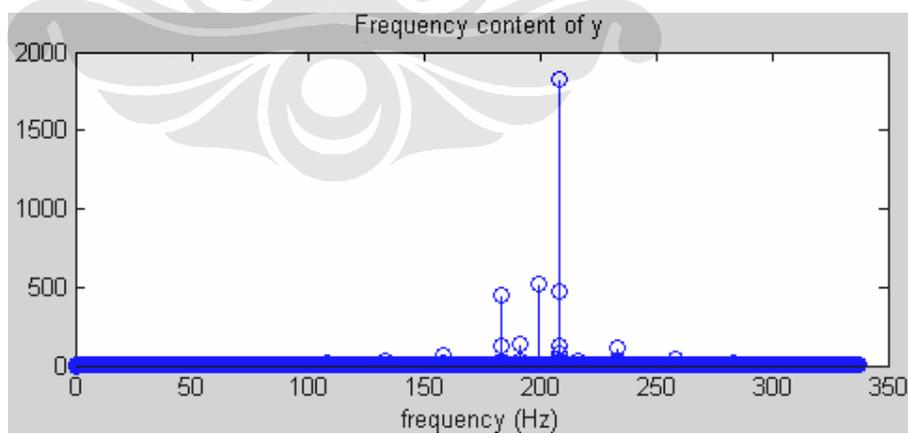
Gambar 4.20. Spektrum sinyal sebenarnya saat data adalah pengulangan dari vektor data (1 0), frekuensi pembawa = 200 Hz dan *data rate* rendah.



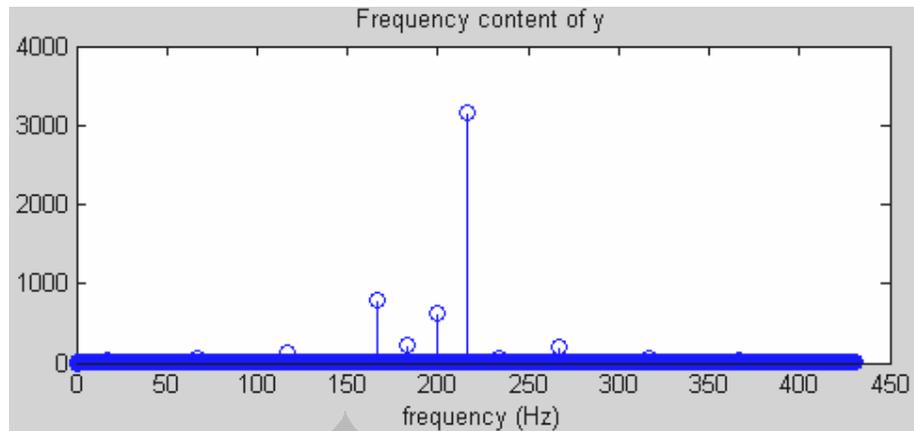
Gambar 4.21. Spektrum sinyal sebenarnya saat data adalah pengulangan dari vektor data (1 0), frekuensi pembawa = 200 Hz dan *data rate* standar.



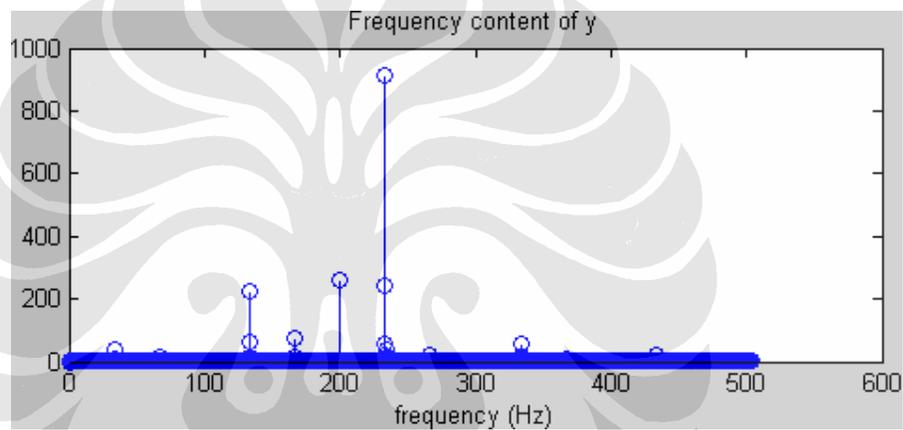
Gambar 4.22. Spektrum sinyal sebenarnya saat data adalah pengulangan dari vektor data (1 0), frekuensi pembawa = 200 Hz dan *data rate* tinggi.



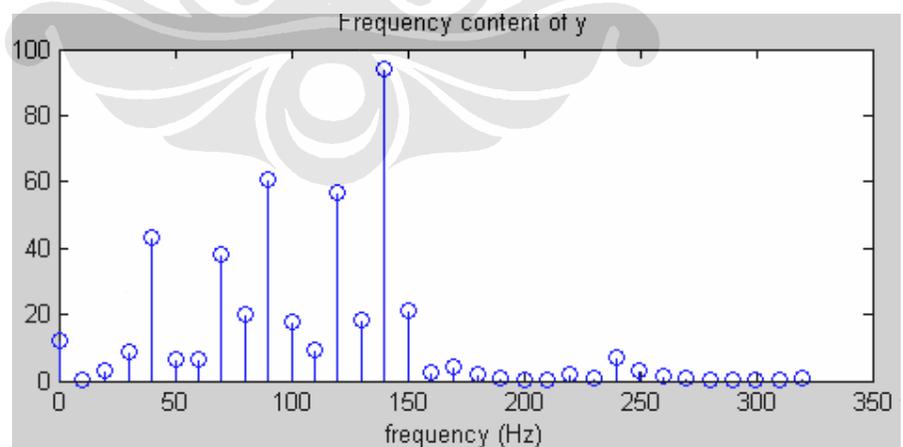
Gambar 4.23. Spektrum sinyal hasil simulasi saat data adalah pengulangan dari vektor data (1 0), frekuensi pembawa = 200 Hz dan *data rate* rendah.



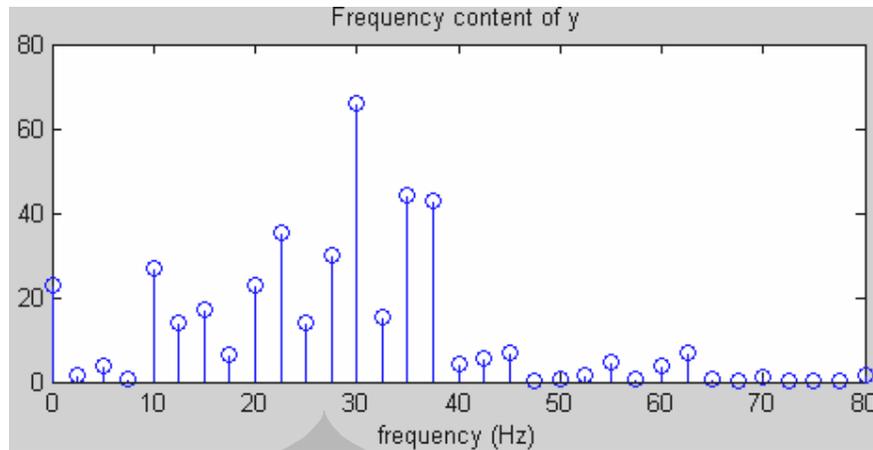
Gambar 4.24. Spektrum sinyal hasil simulasi saat data adalah pengulangan dari vektor data (1 0), frekuensi pembawa = 200 Hz dan *data rate* standar.



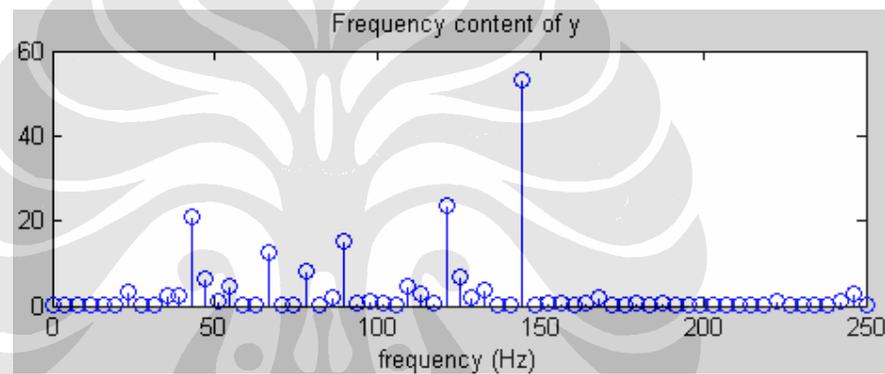
Gambar 4.25. Spektrum sinyal hasil simulasi saat data adalah pengulangan dari vektor data (1 0), frekuensi pembawa = 200 Hz dan *data rate* tinggi.



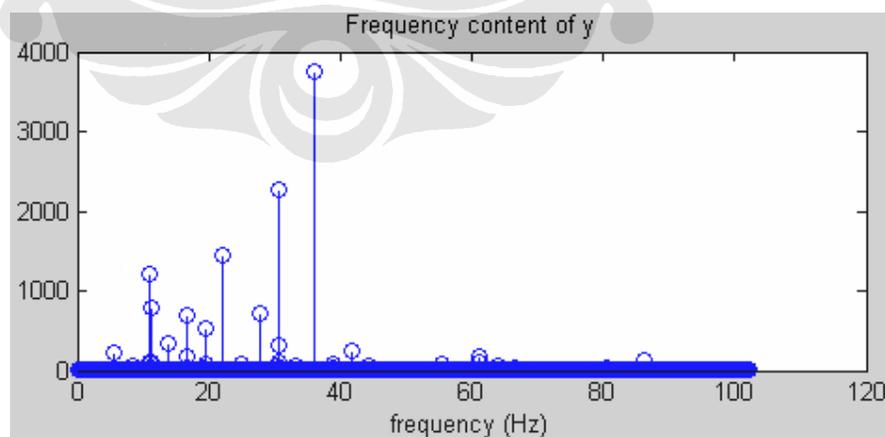
Gambar 4.26. Spektrum sinyal sebenarnya saat data adalah pengulangan dari vektor data (0 1 0), frekuensi pembawa= 100 Hz dan *data rate* tinggi.



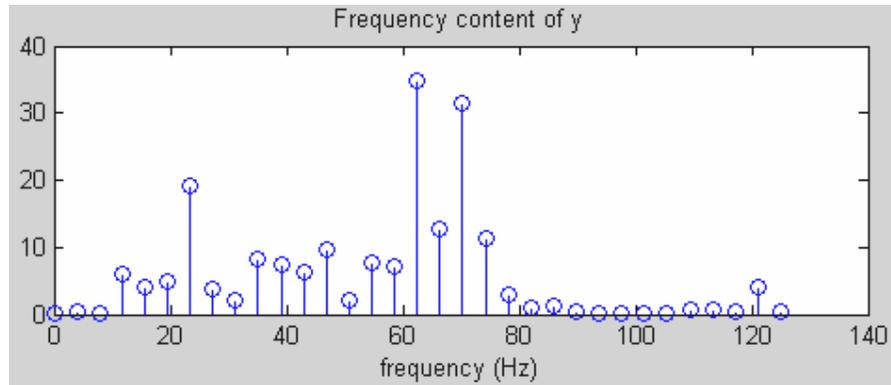
Gambar 4.27. Spektrum sinyal sebenarnya saat data adalah pengulangan dari vektor data (0 1 0), frekuensi pembawa= 25Hz dan *data rate* rendah.



Gambar 4.28. Spektrum sinyal keluaran hasil simulasi saat data adalah pengulangan dari vektor data (0 1 0), frekuensi pembawa=100Hz dan *data rate* tinggi.



Gambar 4.29. Spektrum sinyal keluaran hasil simulasi saat data adalah pengulangan dari vektor data (0 1 0), frekuensi pembawa=25Hz dan *data rate* rendah.



Gambar 4.30. Spektrum sinyal keluaran hasil simulasi saat data adalah pengulangan dari vektor data (0 1 0), frekuensi pembawa=50Hz dan *data rate* standar.

Pengambilan sampel bentuk gelombang atau sinyal yang terjadi menggunakan parameter frekuensi gelombang pembawa diskrit sebesar 50 Hz. *Data rate (bit rate)* yang digunakan sebesar 100 / 3 bit per detik. Perbandingan jumlah bit data dengan jumlah bit kode PN adalah 1 : 3. Satu paket bit data setara dengan 334 bit data sebelum diulang. Kecepatan proses transfer data yang menggunakan saluran RTDX sebesar 100 / 3 bit per detik. Semua proses pengambilan sampel bentuk sinyal dan spektrumnya menggunakan vektor data masukan yang berulang secara periodik dan tidak menggunakan data yang acak.

Parameter frekuensi gelombang pembawa yang digunakan untuk pengambilan data spektrum sinyal yaitu sebesar 25Hz, 50 Hz, 100 Hz, dan 200 Hz. Parameter *data rate* yang digunakan yaitu *data rate* rendah (100 / 6 bps), *data rate* standar (100 / 3 bps), dan *data rate* tinggi (200 / 3 bps).

*File-file* yang dibangkitkan oleh Simulink untuk menunjang proses instalasi ke dalam memori C6713DSK merupakan *file-file* yang berisi kode-kode bahasa C atau C++. Beberapa pesan yang ditampilkan di layar komputer saat Simulink membangkitkan *file-file* tersebut ditunjukkan Gambar 4.39. Beberapa *file* diantaranya merupakan *file* berekstensi “.h” dan berekstensi “.c”.

*File-file* dengan basis bahasa pemrograman C atau C++ tersebut di-*compile* (dikonversi) ke dalam bahasa *assembler* (disimpan dalam bentuk *file* yang berekstensi “.OUT”) yang selanjutnya dapat langsung di-*compile* secara otomatis ke dalam bahasa mesin (bahasa prosesor) saat proses instalasi dilakukan.

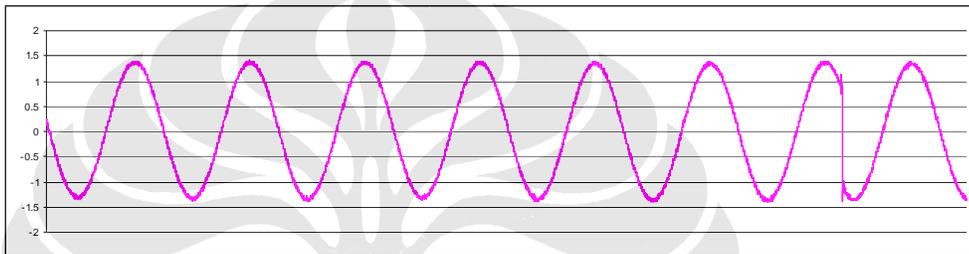
## 4.2 ANALISIS

### 4.2.1 Analisis Sampel Bentuk Sinyal

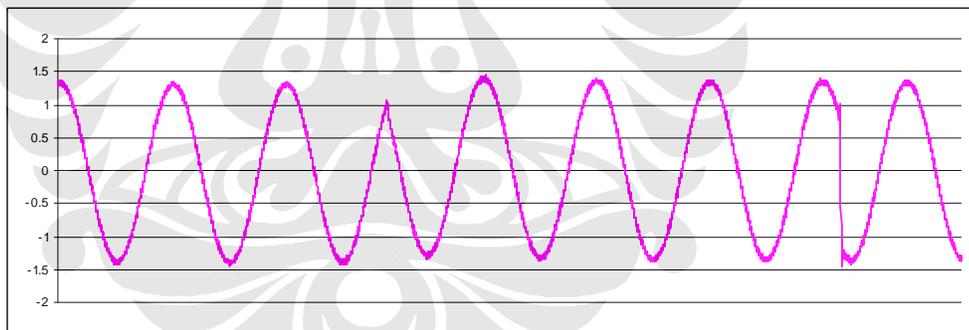
Waktu cuplik sub blok "From RTDX" juga merupakan waktu cuplik sekaligus kecepatan transfer data dari *RTDX channel*. Fungsi **writemsg** pada Matlab 7.4.0) digunakan untuk melakukan pengiriman data dari komputer menuju memori C6713 DSK melalui fasilitas RTDX. Berdasarkan penelitian yang penulis lakukan, pemanggilan fungsi tersebut secara berkali-kali tanpa adanya penundaan waktu yang cukup antara satu pemanggilan dengan pemanggilan berikutnya dapat menyebabkan *bug* pada proses pengiriman data. Namun waktu cuplik sub blok tersebut berbeda dengan *data rate* efektifnya (*data rate*). Hal tersebut terjadi karena setiap bit data yang akan ditransfer menuju memori C6713DSK diulang sebanyak dua kali agar jumlah bit setelah diulang sama dengan jumlah bit kode PN (tiga bit) sehingga satu bit data ekuivalen dengan tiga bit *sample*. Seperti telah dijelaskan sebelumnya, pengambilan sampel bentuk gelombang menggunakan waktu cuplik efektif atau *data rate* sebesar 100 / 3 bit per detik. Adanya sub blok "Buffer" menyebabkan *data rate* yang semula merupakan *sample rate* menjadi *frame rate* dengan lebar dua *sample* per *frame*. Hal ini mengakibatkan setiap *sample* memiliki kecepatan transfer yang turun menjadi setengah dari *sample rate* mula-mula. *Sample rate* awal sebesar 100 Hz dan *frame rate* efektif keluaran sub blok "Buffer" sebesar 50 Hz. Oleh karena itu, kecepatan transfer dari parameter I dan Q menuju blok "Mixer" juga sebesar 50 Hz. Periode gelombang pembawa yang diskrit juga sebesar 50 Hz. Oleh karena itu, rasio antara periode gelombang pembawa dengan periode transfer parameter I dan Q sebesar 1 : 1.

Rasio tersebut merupakan rasio terkecil yang dapat diterima, yang berarti bahwa satu periode gelombang pembawa membawa informasi sepasang parameter I dan Q. Hal tersebut dapat dilihat pada Gambar 4.1 hingga Gambar 4.18 yang menunjukkan kecepatan perubahan fasa dari sinyal CDMA. Fasa dari sinyal sangat cepat berubah setiap periodenya. Sinyal keluaran yang sebenarnya terlihat memiliki kemiripan bentuk dengan sinyal keluaran hasil simulasi. Gambar 4.1 hingga Gambar 4.18 menunjukkan adanya kesamaan fasa (urutan fasa) antara sinyal keluaran yang sebenarnya dengan sinyal keluaran hasil simulasi. Hal ini menunjukkan bahwa pengirim CDMA yang dibangun dari algoritma model sudah

berfungsi sebagaimana mestinya. Namun, bentuk dari kedua sinyal tersebut tidak identik. Sinyal keluaran sebenarnya terlihat memiliki amplitudo yang tidak tetap. Kemungkinan dari penyebab terjadinya hal tersebut adalah respon dari impedansi *port* LINE OUT. Namun, jika perulangan fasa terjadi lebih jarang maka amplitudonya cenderung lebih stabil. Hal tersebut ditunjukkan oleh Gambar 4.31 dan 4.32. Gambar 4.16 dan Gambar 4.17 menunjukkan sampel bentuk sinyal yang tidak sama pada *data rate* yang berbeda. Hal ini terjadi karena pengambilan sampel dilakukan pada *word length* DAC yang sama (16 bit) dan *sampling rate* yang sama (32000 Hz).



Gambar 4.31. Keluaran sebenarnya saat vektor data adalah pengulangan dari (1 0), frekuensi pembawa = 200 Hz dan *data rate* = 100 / 6 bps.



Gambar 4.32. Keluaran sebenarnya saat vektor data adalah pengulangan dari (1 0), frekuensi pembawa = 200 Hz dan *data rate* = 100 / 3 bps.

Tabel 4.1 Pola keberulangan 18 bit hasil *spreading* untuk data (1 0 0), (0 1 0), dan (0 0 1).

<i>Sample</i>	Kode PN = (1 1 0)	Hasil <i>spreading</i>
2 x (1 1 1 1 0 0 0 0 0) = (3 2 0 0 1 3 0 0 0)	2 x (1 1 0 1 1 0 1 1 0)	2 x (0 0 1 1 1 0 1 1 0) = (0 3 2 3 0 1 3 1 2)
2 x (0 0 0 1 1 1 0 0 0) = (0 1 3 0 0 0 3 2 0)	2 x (1 1 0 1 1 0 1 1 0)	2 x (1 1 0 0 0 1 1 1 0) = (3 0 1 3 1 2 0 3 2)

$2 \times (000000111) =$ (000320013)	$2 \times (110110110)$	$2 \times (110110001) =$ (312032301)
---	------------------------	---

Gambar-gambar sinyal keluaran ketika data merupakan pengulangan dari vektor (1 0 0), (0 1 0), dan (0 0 1) menunjukkan adanya urutan fasa yang sama. Hal tersebut dapat dijelaskan pada Tabel 4.1. Tabel tersebut menunjukkan pola keberulangan 18 bit tersebut. Jika satu paket data yang dikirimkan menuju pengirim CDMA merupakan vektor data yang diulang-ulang maka pola perulangan parameter I dan Q sama dengan pola perulangan bilangan bulat hasil *spreading* yang ditunjukkan oleh Tabel 4.1. Hal tersebut mengakibatkan adanya perulangan fasa atau urutan fasa yang sama pada ketiga vektor bit data tersebut. Analogi yang sama dapat digunakan untuk menjelaskan alasan terjadinya perulangan fasa atau urutan fasa sinyal yang sama antara sinyal yang berasal dari vektor (1 1 0) dengan sinyal yang berasal dari vektor (1 0 1).

Gambar-gambar bentuk sinyal juga menunjukkan frekuensi perulangan fasa sinyal yang terjadi. Semakin kecil lebar bit vektor data yang diulang maka semakin sering terjadinya perulangan fasa sinyal. Sebagai contoh, sinyal yang berasal dari perulangan vektor (1 0) lebih sering berubah fasa dibandingkan sinyal yang berasal dari perulangan vektor (1 0 1). Hal tersebut terjadi ketika *data rate* dan periode gelombang pembawa diantara kedua jenis vektor data yang saling berbeda lebar bit tersebut adalah sama.

#### 4.2.2 Analisis Sampel Spektrum Sinyal

##### 4.2.2.1 Analisis Pengaruh Frekuensi Carrier Terhadap Spektrum Sinyal

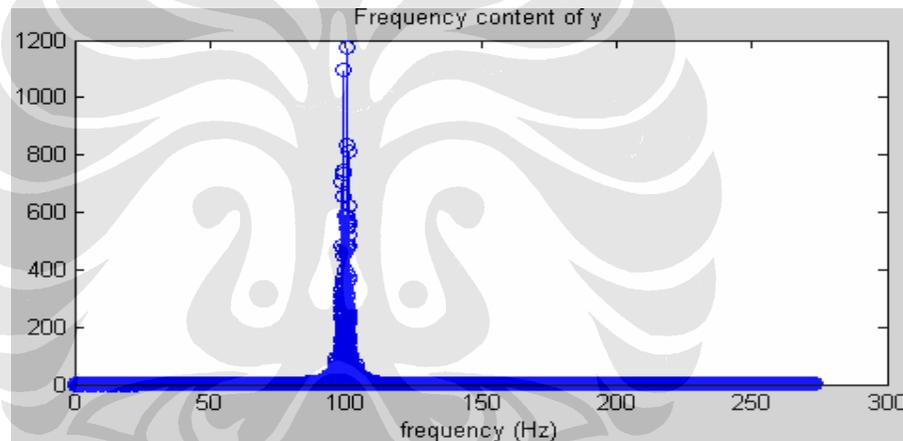
Gambar 4.20 menunjukkan spektrum sinyal saat frekuensi *carrier* (gelombang pembawa) mencapai puncaknya dan nilai *data rate* berada di nilai minimumnya. Hal tersebut membuat sebagian besar bentuk sinyal CDMA yang ditransmisikan merupakan bentuk sinusoidal dengan kecepatan perubahan fasa yang relatif lebih kecil jika dibandingkan dengan frekuensi gelombang pembawa. Hal tersebut juga mengakibatkan pengalokasian daya sinyal lebih banyak terjadi pada frekuensi gelombang pembawa karena perbedaan frekuensi atau *rate* antara data dan pengulangan vektornya dengan gelombang pembawa sudah dianggap cukup jauh. Oleh karena itu, keadaan tersebut cenderung membuat spektrum sinyal lebih stabil pada frekuensi gelombang pembawa dan *data rate* yang tetap

meskipun data yang dipakai adalah data acak (tidak memiliki pola pengulangan). Jenis *windowing* yang dipakai (*rectangular*) mempengaruhi bentuk spektrum.

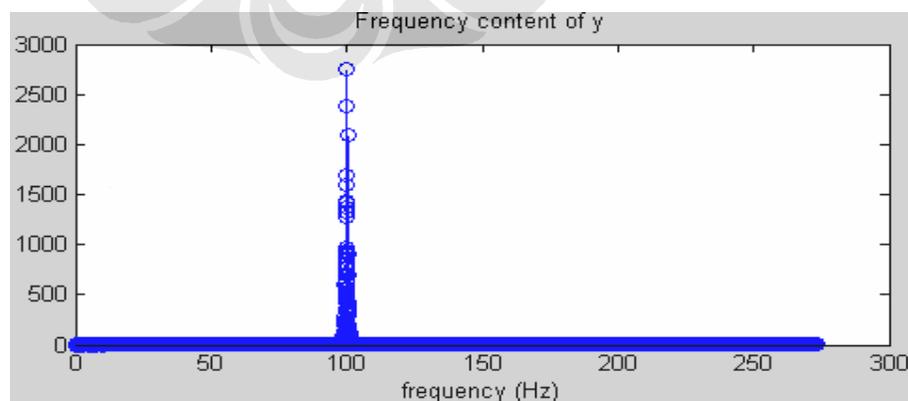
Gambar 4.21 menunjukkan spektrum sinyal saat perbandingan frekuensi gelombang pembawa dengan nilai *data rate* tidak terlalu besar. Spektrum sinyal saat rasio tersebut terjadi lebih kecil dari sebelumnya ditunjukkan oleh Gambar 4.22. Rasio yang lebih kecil tersebut mengakibatkan frekuensi yang mendominasi persentase daya sinyal menjadi semakin jauh dari frekuensi gelombang pembawa atau alokasi persentase daya sinyal pada frekuensi gelombang pembawa menjadi semakin kecil. Gambar 4.21 menunjukkan adanya dua frekuensi yang mendekati frekuensi gelombang pembawa yang mulai mendominasi persentase daya sinyal. Sementara itu, Gambar 4.22 menunjukkan mulai adanya pergeseran frekuensi (yang memiliki daya tertinggi) dari frekuensi gelombang pembawa. Gambar-gambar hasil simulasi yang bersesuaian dengan gambar-gambar tersebut juga menunjukkan hal yang sama. Adanya sedikit perbedaan bentuk spektrum antara spektrum sinyal yang sebenarnya dengan spektrum sinyal hasil simulasi disebabkan oleh perbedaan bingkai waktu dan lebarnya yang digunakan pada proses *windowing*. Lebar bingkai waktu yang digunakan pada proses *windowing* sinyal hasil simulasi beberapa kali lebih lebar dibandingkan dengan lebar bingkai waktu yang digunakan pada proses *windowing* sinyal sebenarnya. Rasio antara frekuensi gelombang pembawa dengan *data rate* yang tidak cukup (tidak memenuhi syarat) dapat membuat spektrum menjadi lebih labil jika vektor datanya berubah-ubah. Spektrum sinyal saat digunakan rasio frekuensi gelombang pembawa dengan *data rate* yang terkecil ditunjukkan oleh Gambar 4.26 hingga Gambar 4.30. Pada Gambar 4.26 hingga 4.30, frekuensi yang mendominasi persentase daya sinyal keluaran bergeser cukup jauh dari frekuensi gelombang pembawanya. Oleh karena itu, rasio terbaik antara frekuensi gelombang pembawa dengan *data rate* efektif yang digunakan untuk pengambilan sampel yaitu 200 : 50 atau 4 : 1. Rasio yang berada di atas nilai tersebut adalah rasio yang lebih baik karena dapat menciptakan spektrum yang lebih stabil dengan *slope* yang lebih tinggi.

#### 4.2.2.2 Analisis Pengaruh Kode PN Terhadap Spektrum Sinyal

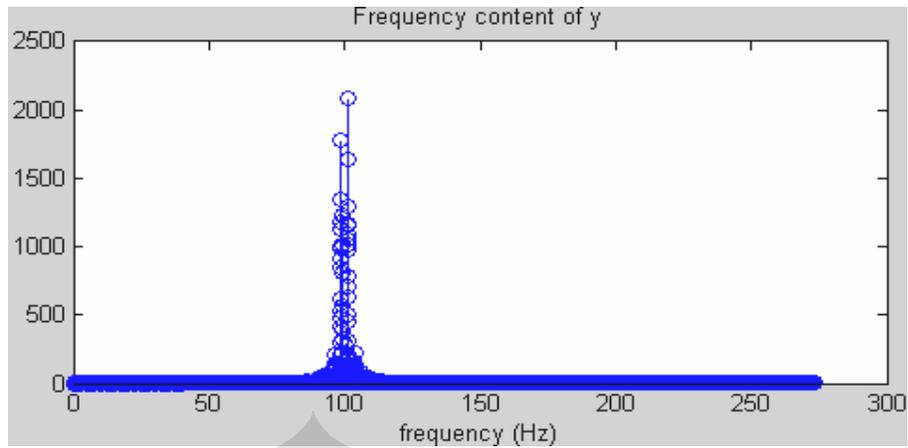
Pengambilan sampel spektrum sinyal keluaran hasil simulasi dengan variasi kode PN dilakukan pada rasio antara frekuensi *carrier* dengan *data rate* yang ideal atau rasio yang nilainya lebih besar dari 4 : 1. Gambar 4.33 hingga Gambar 4.38 menunjukkan spektrum-spektrum sinyal keluaran hasil simulasi dengan kode PN yang bervariasi. Data yang digunakan dalam simulasi yang hasilnya ditunjukkan oleh Gambar 4.33 hingga Gambar 4.38 tersebut merupakan data acak. Rasio antara frekuensi *carrier* dengan *sample rate (bit rate)* kode PN yang digunakan adalah sebesar 20 : 1. Bentuk spektrum berubah ketika kode PN yang digunakan berbeda dari sebelumnya karena penggunaan kode PN memperlebar spektrum data keluaran sub blok "XOR". Bentuk spektrum sinyal pada bingkai frekuensi yang lebih sempit ditunjukkan Gambar 4.37.



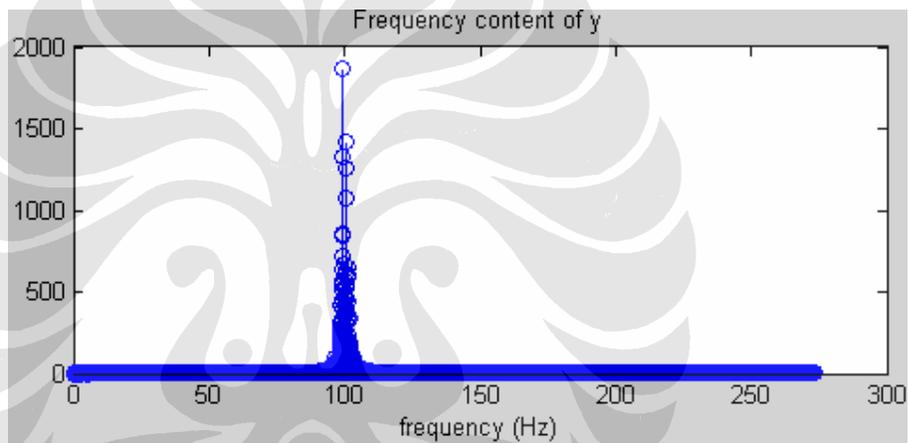
Gambar 4.33. Spektrum sinyal keluaran hasil simulasi saat lebar kode PN pertama adalah 3 bit per bit data.



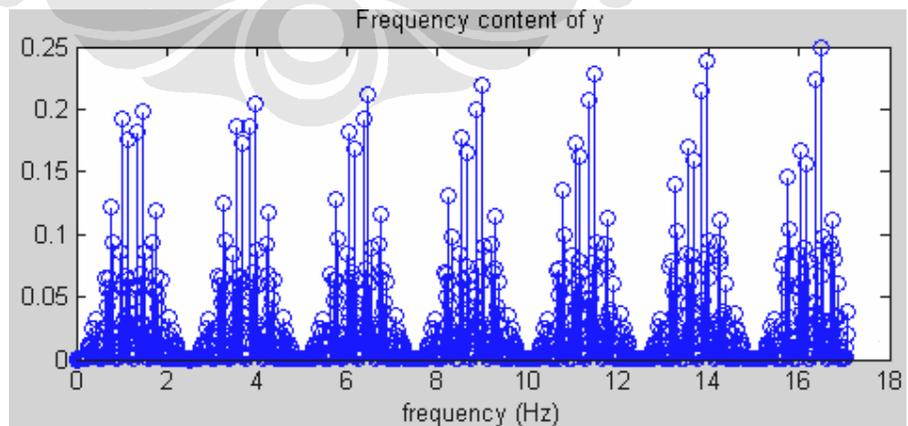
Gambar 4.34. Spektrum sinyal keluaran hasil simulasi saat lebar kode PN kedua adalah 3 bit per bit data.



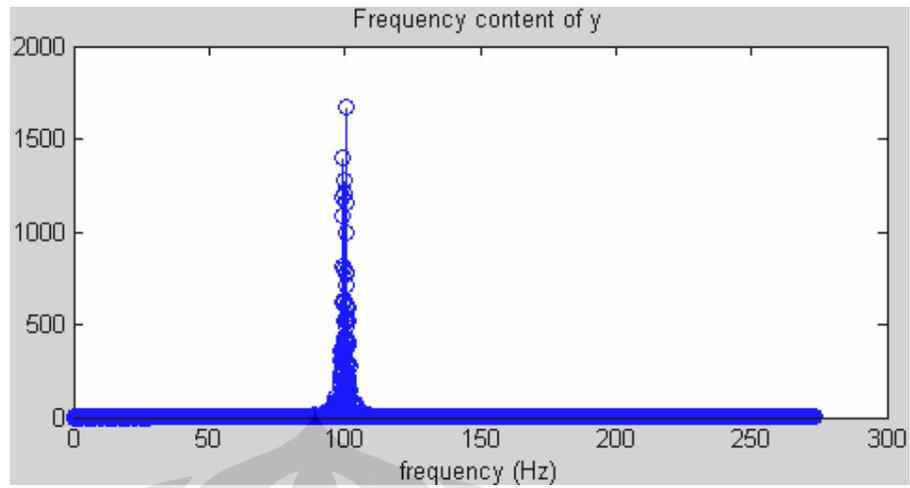
Gambar 4.35. Spektrum sinyal keluaran hasil simulasi saat lebar kode PN ketiga adalah 7 bit per bit data.



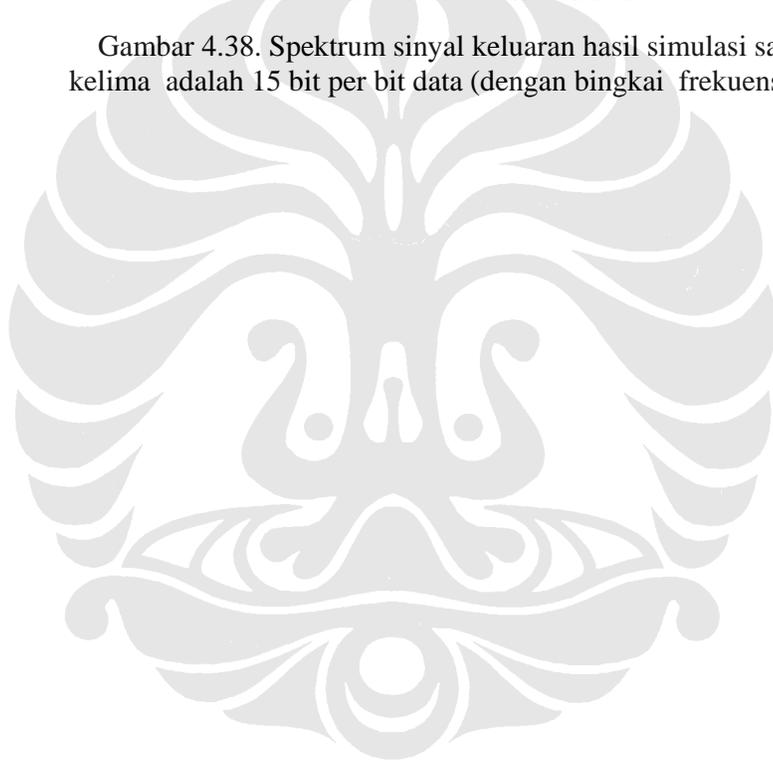
Gambar 4.36. Spektrum sinyal keluaran hasil simulasi saat lebar kode PN keempat adalah 7 bit per bit data.



Gambar 4.37. Spektrum sinyal keluaran hasil simulasi saat lebar kode PN kelima adalah 15 bit per bit data.



Gambar 4.38. Spektrum sinyal keluaran hasil simulasi saat lebar kode PN kelima adalah 15 bit per bit data (dengan bingkai frekuensi yang lebih lebar).



## BAB 5

### KESIMPULAN

Beberapa kesimpulan yang dapat diambil dari pembahasan mengenai pengirim CDMA yang telah dibangun yaitu:

1. Prosesor TMS320C6713 dapat dijadikan sebagai komponen utama dari pengirim CDMA yang dibahas pada skripsi ini.
2. Pengirim CDMA yang dibangun telah berfungsi sebagaimana mestinya.
3. Frekuensi dari sinyal yang ditransmisikan bergantung dari frekuensi gelombang pembawa yang digunakan.
4. Bentuk spektrum dari sinyal yang ditransmisikan dipengaruhi oleh rasio antara frekuensi gelombang pembawa dengan *data rate* dan kode PN yang digunakan.
5. Bit *rate* kode PN lebih besar daripada bit *rate* data.
6. Hanya penerima yang memiliki kode PN yang sama dengan kode PN sinyal yang dapat membaca bit data pada sinyal tersebut, hal tersebut terjadi karena penerima melakukan operasi NXOR terhadap sinyal.
7. Pemanggilan fungsi **writemsg** tanpa adanya penundaan waktu antara pemanggilan saat tertentu dengan pemanggilan berikutnya dapat mengakibatkan *bug* pada sistem RTDX yang sedang digunakan.
8. Rasio yang antara frekuensi gelombang pembawa dengan *data rate* efektif yang diperbolehkan untuk digunakan pada pengirim CDMA ini harus lebih besar dari 4 : 1.

## DAFTAR ACUAN

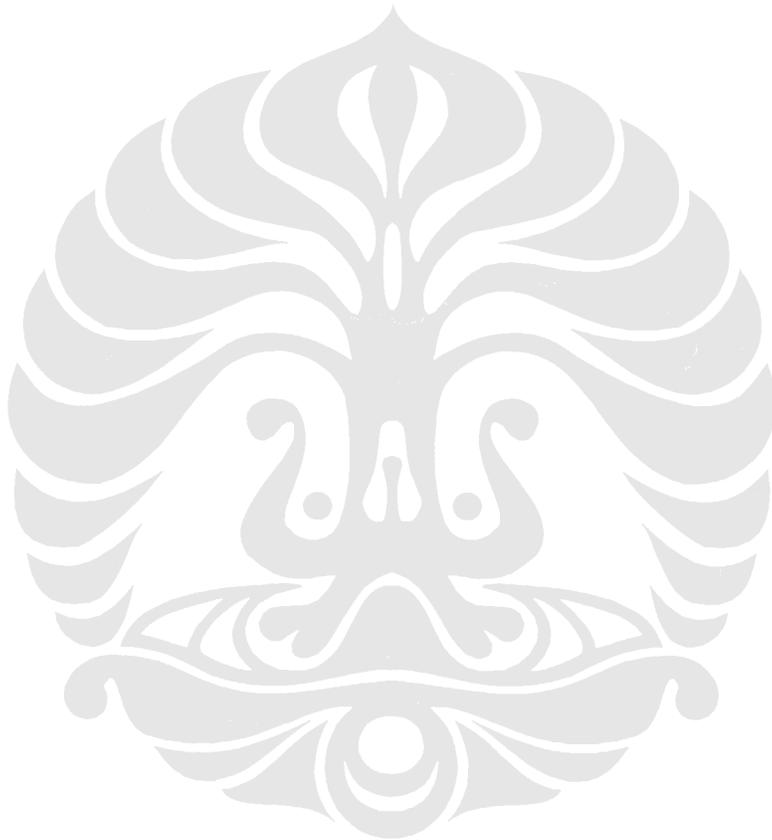
- [1] Frescura, F. , “TMS320C6713 DSK Technical Reference”, Spectrum Digital, Inc., Mei 2003
- [2] Chassaing Rulph, ”Digital Signal Processing and Applications With the C6713 and C6416 DSK”, A John Wiley & Sons, Inc., 2005
- [3] Arman, D.P., ”Penelitian Sistem Pengirim dan Penerima CDMA.” Proposal Penelitian. Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia., 2007.
- [4] Matlab 7.4.0 Help.



## DAFTAR PUSTAKA

Viterbi, Andrew. *Principles of Spread Spectrum Communication*. Addison Wesley Longman. Massachusetts, 1995.

Ipatov, Valery. *Spread Spectrum and CDMA*. John Willey and Sons. England, 2005.



## LAMPIRAN

### Lampiran 1 M-file “driver2.m”

```
function RTDXdriver(modelname)
load ('sumber.mat');
(modelpath,modelname,modelext) = fileparts(modelname);
cc = ccdsp;
set(cc,'timeout',50);
if ~isrtdxcapable(cc)
    error('Processor does not RTDX support');
end

cc.reset; pause(1);
cc.cd(modelpath);
cc.visible(1);

open(cc,sprintf('%s.pjt',modelname));
load(cc,sprintf('%s.out',modelname));

rx = cc.rtdx;
rx.set('timeout', 50); % Reset timeout = 10 seconds

rx.configure(64000,4);

rx.open('ichan','w');

rx.enable; % enable RTDX

cc.run; % cc.enable can be placed here

pause(1);
if isenabled(rx,'ichan')
    writemsg(cc.rtdx,'ichan',uint8(source))
end
pause(40);
RTDXcleanup(cc,rx);
%=====
% Put RTDX back to good state
%=====
function RTDXcleanup(cc,rx)
if isrunning(cc), % if the target DSP is running
    halt(cc); % halt the processor
end

cc.reset;
disable(rx,'ichan');
disable(rx); % disable RTDX

close(cc.rtdx,'ichan');
```

## Lampiran 2 M-file “spectrum\_analyzer.m”

```
%-----  
% create suitable matrix  
%-----  
batas=32;  
t=1;  
p=1;  
for a=1:10000  
    if abs(sinyal_CDMA(a,1))< t  
        t=abs(sinyal_CDMA(a,1));  
        p=a;  
    end  
end  
r=1/abs(sinyal_CDMA(p,1)-sinyal_CDMA(p+1,1));  
sinyal=sinyal_CDMA(:,2);  
%-----  
%clear variable  
%-----  
clear p;  
clear t;  
clear similar_matrix;  
clear time;  
clear sinyal_CDMA;  
clear a;  
clear b;  
%-----  
%fft process  
%-----  
Y=fft(sinyal,10000);  
Pyy = Y.* conj(Y) / 100000;  
f = r*(0:batas)/10000;  
stem(f,Pyy(1:batas+1));  
title('Frequency content of y');  
xlabel('frequency (Hz)');  
%-----  
%clear variable  
%-----  
clear Pyy;  
clear Y;  
clear sinyal;  
clear f;  
clear r;
```

### Lampiran 3 M-file “direct\_spectrum\_analyzer.m”

```
%-----  
%fft process  
%-----  
batas=16384;  
pp=size(simout);  
p=pp(1);  
sinyal=simout;  
Y=fft(sinyal,p);  
Pyy = Y.* conj(Y) /(10*p);  
f = 32000*(0:batas)/p;  
stem(f,Py(1:batas+1));  
title('Frequency content of y');  
xlabel('frequency (Hz)');  
%-----  
%clear variable  
%-----  
clear Pyy;  
clear f;  
clear Y;  
clear sinyal;  
clear pp;  
clear p;  
clear batas;
```

