

PENGGUNAAN PROSESOR DSP SEBAGAI PENGGANTI KOMPONEN RANGKAIAN CDMA

Arman Djohan D.

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Indonesia, Depok, 16424

E-mail: armandjohan@yahoo.com

Abstrak

Dalam penelitian ini dibuat perangkat lunak Prosesor DSP (Digital Signal Processing) jenis TMS320C54X untuk memproses kerja rangkaian CDMA (*Code Division Multiple Access*). Dengan menggunakan beberapa algoritma yaitu algoritma matematis dan algoritma *table lookup* diteliti algoritma manakah yang mempunyai waktu proses yang paling cepat serta algoritma manakah yang memerlukan memori yang paling kecil. Dari hasil pengamatan, Algoritma Matematis memerlukan waktu proses 20.62 milidetik, sedangkan algoritma *table lookup* memerlukan waktu proses 14.08 milidetik. Dari hasil pengamatan dengan menggunakan metode *table lookup* waktu proses CDMA lebih cepat dibandingkan dengan metode Algoritma Matematis karena dengan algoritma Matematis terjadi proses pembulatan yang tidak sama untuk setiap data masukan. Waktu proses 14.03 milidetik masih dibawah waktu delay yang diizinkan yaitu 125 milidetik. Meskipun demikian waktu proses tersebut masih ditambah dengan waktu proses Error correction dan proses pengkompesan suara (ADPCM atau QCELP).

Abstract

This research develops the software of DSP Processor type TMS320C54X for the CDMA circuitry. This research apply two algorithms namely Mathematic Algorithm and *table lookup* Algorithm. From both algorithms, the research examine what algorithm have the fastest processing time and the smallest memory occupation. From the examination results the Mathematic Algorithm have 20.62 milliseconds processing time and the *Table lookup* algorithm have 14.08 milliseconds processing time. The *Table lookup* algorithm have the fastest processing time because the Mathematic algorithm need the rounding process of the figure. Eventhough such processing time is not exceed from the 125 millisecond of delay time, in real condition, the total processing time are include the error correction processing time and voice compression processing time such as ADPCM or QCELP voice compression techniques.

Keywords: DSP, digital signal processing, CDMA

Pendahuluan

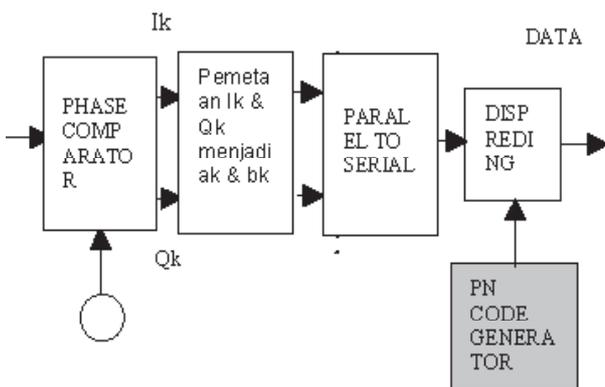
Dengan berkembangnya teknik radio selular yang tidak hanya digunakan sebagai komunikasi bergerak tetapi juga digunakan sebagai jaringan komunikasi tetap atau dikenal dengan istilah *Wireless Local Loop* (WLL) khususnya untuk daerah pedesaan, kota, [1] perlu dipikirkan kemungkinan dapat diproduksi di dalam negeri mengingat Indonesia terdiri dari ribuan kota serta desa. Perangkat komunikasi radio selular terdiri dari 2 bagian besar yaitu, bagian *baseband* dan bagian RF (*Radio Frequency*). Bagian RF terdiri dari subbagian pengolah suara (*Voice compression* dan *Error correction*), subbagian rangkaian Akses dan Modulator. Penggunaan rangkaian akses CDMA (*Code Division Multiple Access*) sangat cocok digunakan di Indonesia karena dapat menghindari adanya *jamming*,

gangguann gelombang pantul, pemakaian frekwensi yang berulang [2][3]. Di samping itu pula kemungkinan adanya integrasi dengan GSM sebagai Komunikasi bergerak [4] Dengan adanya teknologi prosesor DSP, sinyal suara dapat diproses langsung oleh prosesor tersebut, sehingga rangkaian *baseband* dapat diproses dengan perangkat lunak dari prosesor DSP[5]. Dengan mengembangkan perangkat lunak tersebut, rangkaian *baseband* yang biasanya menggunakan komponen chip dapat diganti dengan Prosesor DSP yang perangkat lunaknya dapat dikembangkan oleh pakar-pakar Indonesia. Prosesor DSP itu sendiri mudah dan murah diperoleh karena diproduksi oleh sejumlah produsen sehingga harganya pun bersaing. Masalah yang timbul pada saat mengembangkan perangkat lunak adalah algoritma yang digunakan, agar diperoleh waktu proses yang singkat serta memerlukan pemakaian

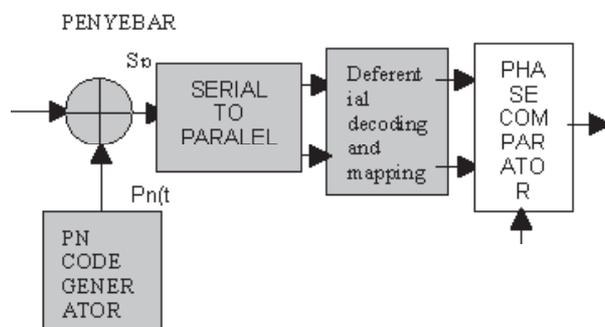
memori yang kecil. Terdapat 4 teknik algoritma yang digunakan yaitu : algoritma matematis, algoritma *table lookup*, algoritma stokastik dan algoritma cerdas buatan. Pada penelitian ini menggunakan algoritma matematis dan algoritma *table lookup*.

Metode Penelitian

Rangkaian CDMA terdiri dari 2 bagian yaitu bagian Pembangkit dan bagian Pendeteksi. Diagram blok bagian Pembangkit dapat dilihat pada Gambar 1., sedangkan diagram blok bagian Pendeteksi dapat dilihat pada Gambar 2. Bagian Pembangkit terdiri dari PN Code Generator, rangkaian Penyebar, dan Modulator DQPSK atau $\pi/4$ DQPSK [6]. Sedangkan rangkaian Deteksi CDMA terdiri dari rangkaian Dispreeding dan PN Code Generator [7]. (Lihat Gambar 1 dan 2).



Gambar 1 Rangkaian pendeteksi CDMA



Gambar 2. Rangkaian pembang

Output dari rangkaian Penyebar adalah :

$$St(t) = D(t) \text{ XOR } Pn(t) \tag{1}$$

Output dari Rangkaian *Differential* dan *Decoding* adalah [6] :

$$I_k = I_{k-1} \cdot \cos [\Delta\phi(ak,bk)] - Q_{k-1} \cdot \sin [\Delta\phi(ak,bk)] \tag{2}$$

$$Q_k = I_{k-1} \cdot \sin[\Delta\phi(ak,bk)] + Q_{k-1} \cdot \cos[\Delta\phi(ak,bk)] \tag{3}$$

Dari persamaan di atas terlihat nilai I_k dan Q_k ditentukan dari nilai sebelumnya (I_{k-1} dan Q_{k-1}). Perubahan fasa $\Delta\phi$ (ak , bk) tergantung dari (ak , bk) dengan besar fasa dapat dilihat pada Tabel 1 di bawah.

Besarnya Fasa yang dihasilkan pada proses pemetaan I_k dan Q_k menjadi ak dan bk dari **Rangkaian Pendeteksi** diperoleh dari persamaan di bawah ini [7] :

$$\begin{aligned} \text{Cos}(\Delta\phi) &= \frac{I_k \cdot I_{k-1} + Q_k \cdot Q_{k-1}}{(Q_{k-1})^2 + (I_{k-1})^2} \\ \text{Sin}(\Delta\phi) &= \frac{Q_k \cdot I_{k-1} - I_k \cdot Q_{k-1}}{(Q_{k-1})^2 + (I_{k-1})^2} \end{aligned} \tag{4, 5}$$

nilai ak dan bk dapat ditentukan dengan mencari tanda (+/-) dari Persamaan (4) dan (5).

- Jika $\text{Cos}(\Delta\phi) > 0$ maka $ak = 1$ dan jika $\text{Cos}(\Delta\phi) < 0$ maka $ak = 0$
- Jika $\text{Sin}(\Delta\phi) > 0$ maka $bk = 1$ dan jika $\text{Sin}(\Delta\phi) < 0$ maka $bk = 0$

Untuk mendapatkan tanda (+/-) tersebut maka hanya diperlukan perhitungan dari pembilangnya saja yaitu :

$$\text{Sign} (\text{Cos}(\Delta\phi)) = I_k \cdot I_{k-1} + Q_k \cdot Q_{k-1} \tag{6}$$

Tabel 1. Penggeseran fasa sebagai fungsi informasi

ak	bk	$\Delta\phi$	$\text{Cos } \Delta\phi$	$\text{Sin } \Delta\phi$
0	0	$5\pi/4$	-0.707	-0.707
0	1	$3\pi/4$	-0.707	+0.707
1	0	$7\pi/4$	+0.707	-0.707
1	1	$\pi/4$	+0.707	+0.707

$$\text{Sign}(\sin(\Delta\phi)) = Q_k \cdot I_{k-1} + I_k \cdot Q_{k-1} \quad (7)$$

Keluaran dari *Despreading* menjadi sinyal aslinya adalah:

$$b(t) = b(t) \cdot c(t) \cdot c(t) \quad (8)$$

dimana nilai $b(t) \cdot c(t)$ adalah masukan ke rangkaian *Despreading*. (lihat Gambar 3).

Algoritma Pemrograman untuk pembangkit adalah sebagai berikut :

- Algoritma Penyebaran (Spreading)

Mulai

Pengulangan $i < 8$

Mulai

Baca data bit ke l

Pengulangan $j < 128$

Mulai

Baca kode PN bit ke j

spread bit ke $j = \text{data bit ke } l \text{ XOR dengan kode PN bit ke } j$

Jika spread bit ke $j = 1$ maka

spreadoutput = +1

Jika spread bit ke $j = 0$

maka spreadoutput = - 1

Simpan nilai spreadoutput bit ke l, j

selesai

Selesai

selesai

- Serial to Paralel Converter

Mulai

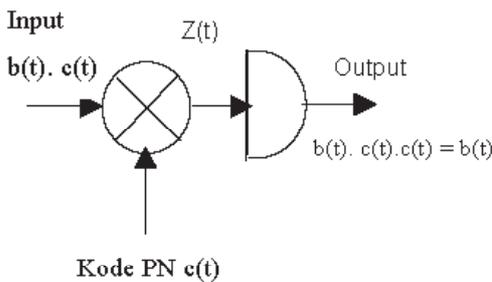
Pengulangan $l < 1024$

Mulai

Baca urutan spreadoutput ke l

Jika l genap maka $a_k = \text{spreadoutput ke } l$

$Bk [i] = bk [i-1]$



Gambar 4. Diagram blok *despreading*

Jika l ganjil maka $b_k = \text{spreadoutput ke } i$
 $Ak [i] = ak [i-1]$

Proses differential dan Mapping

Selesai

selesai

- Algoritma Differential dan Mapping

Mulai

Pengulangan $l < 512$

Mulai

Baca nilai $ak [l]$ dan $bk [l]$

Jika $ak [l] = 1$ dan $bk [l] = 1$ maka

$I_k = I_{k-1} * \cos(\pi/4) - q_{k-1} * \sin(\pi/4)$

$q_k = I_{k-1} * \sin(\pi/4) + q_{k-1} * \cos(\pi/4)$

Jika $ak [l] = 1$ dan $bk [l] = 0$ maka

$I_k = I_{k-1} * \cos(7\pi/4) - q_{k-1} * \sin(7\pi/4)$

$q_k = I_{k-1} * \sin(7\pi/4) + q_{k-1} * \cos(7\pi/4)$

Jika $ak [l] = 0$ dan $bk [l] = 1$ maka

$I_k = I_{k-1} * \cos(3\pi/4) - q_{k-1} * \sin(3\pi/4)$

$q_k = I_{k-1} * \sin(3\pi/4) + q_{k-1} * \cos(3\pi/4)$

Jika $ak [l] = 0$ dan $bk [l] = 0$ maka

$I_k = I_{k-1} * \cos(5\pi/4) - q_{k-1} * \sin(5\pi/4)$

$q_k = I_{k-1} * \sin(5\pi/4) + q_{k-1} * \cos(5\pi/4)$

kirim I_k dan a_k ke port keluaran

selesai

selesai

- Algoritma pemetaan I_k dan Q_k

Mulai

Pengulangan $l < 512$

Mulai

Baca nilai I_k dan Q_k ke l

Jika $I_k > -1.35$ atau $Q_k < -0.8$ maka I_k dan $Q_k = -1$

Jika $I_k > 0.8$ atau $Q_k < -0.35$ maka I_k dan $Q_k = -0.7$

Jika $I_k > -0.35$ atau $Q_k < 0.35$ maka I_k dan $Q_k = 0$

Jika $I_k > 0.35$ atau $Q_k < 0.8$

maka I_k dan $Q_k = 0.7$

Jika $I_k > 0.8$ atau $Q_k < 1.35$ maka I_k dan $Q_k = 1$

Hitung $\text{sign} \sin(\Delta\phi) = I_k \cdot I_{k-1} + Q_k \cdot Q_{k-1}$

Jika $\text{sign} \sin(\Delta\phi) > 0$ maka $ak [l] = 1$

dan jika $\text{sign} \sin(\Delta\phi) \leq 0$

maka $ak [l] = 0$

Hitung $\text{sign} \sin(\Delta\phi) = Q_k \cdot I_{k-1} + I_k \cdot Q_{k-1}$

Jika $\text{sign} \sin(\Delta\phi) > 0$ maka $bk [l] = 1$

dan jika $\text{sign} \sin(\Delta\phi) \leq 0$

maka $bk [l] = 0$

Jika l ganjil maka $\text{spin} [l] = ak [l]$

Jika l genap maka $\text{spin} [l] = bk [l]$

proses **Despreading**

selesai

selesai

- Algoritma Despreading

```

Mulai
  Pengulangan l<8
  Mulai
    Bit data [ l ] = spin [ l ] XOR dengan kode
    PN bit ke j
  Jika spin [ l ] = 1
    maka bit data ke l = +1
  Jika spin [ l ] = 0
    maka bit data ke l = - 1
  Kirim bit data ke l ke port keluaran
  Selesai
Selesai
    
```

Uji coba dilakukan dengan menggunakan Simulator Prosesor DSP TMS329C54X. Pemrograman ditulis dalam bahasa pemrograman C++ dan oleh simulator dikonversi ke Bahasa Pemrograman Assembler TMS [7].

Hasil dan Pembahasan

Dengan melakukan 5 percobaan untuk setiap algoritma (algoritma *table lookup* dan algoritma matematis), akan diperoleh Hasil uji coba unjuk kerja rangkaian CDMA dan untuk waktu proses dari masing-masing algoritma. Setiap percobaan mempunyai data masukan yang berbeda masing-masing data 555H, 123H, aaaH, abcH dan xyzH. Dari hasil uji coba, algoritma *table lookup* untuk setiap percobaan menghasilkan keluaran yang sama sesuai dengan teori.

Hasil ujicoba unjuk kerja rangkaian CDMA dengan algoritma matematis, untuk 5 percobaan hasilnya berbeda. Dilihat dari hasil uji coba bagian Pembangkit, keluaran dari rangkaian Differential dan Pemetaan tidak mengalami ~~Tabel 1. Hasil uji coba keluaran rangkaian Difren Qk~~ dan ~~tidak bernilai~~ harga ak dan bk sesuai dengan teori yang dapat dilihat pada Tabel 1 dimana nilai 0.999 dibulatkan menjadi nilai 1.

Bit l	a	b	D	Q
488	0	1	0.000	-0.999
489	1	1	0.707	0.707
490	1	1	0.000	0.999
491	1	1	-0.707	0.707
492	0	0	-0.999	0.000
493	0	0	0.707	0.707
494	0	0	-0.999	-0.999
495	0	0	-0.707	0.707
496	0	0	-0.999	0.000
497	0	0	-0.707	-0.707

Tabel 3. Keluaran rangkaian pemetaan Ik dan Qk bagian pendeteksi

Bit l	I	II	III	IV
488	-135	-135	135	-45
489	-135	45	-45	135
490	-45	45	135	45
491	-45	135	-45	-135
492	45	135	135	45
493	-135	-45	45	-45
494	135	-135	-135	-135
495	45	-135	45	45
496	135	135	135	135
497	45	45	-45	-135

Tabel 4. Keluaran rangkaian *despreading*

Bit l	Penc. I		Penc. II		Penc. III	
	a	b	a	b	a	b
488	1	1	1	1	1	0
489	1	1	0	0	0	1
490	0	1	0	0	1	0
491	0	1	1	0	0	1
492	0	0	1	0	1	0
493	1	1	0	1	0	0
494	1	0	1	1	1	1
495	0	0	1	1	0	0
496	1	0	1	0	1	0
497	0	0	0	0	0	1

Dari hasil uji coba keluaran rangkaian Pemetaan Ik dan Qk bagian Pendeteksi (Tabel 3) untuk 4 percobaan dari 5 percobaan dengan data masukan yang sama yaitu 555H menghasilkan keluaran yang berbeda. Demikian pula hasil keluaran rangkaian Despreading, hasilnya pun berbeda (Tabel 4). Hal ini disebabkan adanya pembulatan dari perhitungan $\sin(\Delta\phi)$ (lihat algoritma pemetaan Ik dan Qk). Jadi dengan menggunakan algoritma matematis menghasilkan data keluaran yang berbeda dari data masukan pembangkit (Pengirim).

Untuk mengetahui waktu proses yang diperlukan baik proses pengiriman maupun proses penerimaan, perlu diketahui jumlah *cycle* yang dihasilkan untuk setiap proses. Waktu proses diperlukan untuk mengetahui delay yang dihasilkan melebihi 250 milidetik. Prosedur yang dilakukan untuk menghitung *cycle*, pertama kali ditentukan terlebih dahulu titik *Breakpoint* yaitu tempat di mana program berhenti untuk diproses. Penentuan titik Breakpoint ini dilakukan dengan cara menekan OpCode pada Program bahasa assembler di jendela DIASSEMBLY atau dapat juga menentukannya di Program bahasa C di jendela FILE. Untuk menghitung jumlah *cycle* yang dikerjakan sampai

selesai maka penempatan *breakpoint* haruslah pada bagian akhir dari proses baik itu pada *assembler* atau C. Kemudian program dijalankan (*run*) dengan menekan tombol F% atau dapat juga dengan membuat perintah '*Runb*' yaitu perintah untuk menjalankan proses *benchmarking* pada jendela COMMAND [8]. Proses yang telah selesai dikerjakan ditandai dengan garis terang yang akan muncul pada titik *breakpoint*.

Hasil perhitungan *cycle* dan waktu proses dari bagian Pembangkit (Pengirim) dapat dilihat pada Tabel 5. Sedangkan hasil perhitungan *cycle* dan waktu Proses bagian Penditeksi (Penerima) dapat dilihat pada Tabel 6. Total Waktu tunda CDMA dengan algoritma matematis 19.62 milidetik.

Untuk melihat perbandingan waktu proses untuk kedua algoritma dapat dilihat pada Tabel 7 dan Tabel 8.

Tabel 5. Waktu proses dan Jumlah *cycle* pembangkit

Rangkaian	Jumlah Cycle	Waktu Proses (milidetik)
Pengobar (Spreading)	472254	9.41
Serial to Paralel Converter	44656	0.89
Def. Encoding & Mapping	3330	0.06
Total	520240	10.36

Tabel 6. Jumlah *cycle* & waktu proses blok program rangkaian penditeksi

Rangkaian	Jumlah Cycle	Waktu Proses (milidetik)
Pemetaan Di dan Qk	339207	6.78
Paralel to Serial Converter	44656	0.89
Komansi -1,1 kode PN	806	0.016
Decomading	78818	1.57
Total	433467	9.26

Tabel 7. Perbedaan waktu proses algoritma matematis dengan *table lookup* pembangkit

Rangkaian	Table lookup	Matematis
Pengobar (Spreading)	9.41	9.41
Serial to Paralel Converter	0.89	0.89
Def. Encoding & Mapping	0.03	0.06
Total	10.33	10.36

Tabel 8. Jumlah *cycle* & waktu proses blok program rangkaian Penditeksi

Rangkaian	Table lookup	Matematis
Pemetaan Di dan Qk	1.28	6.78
Paralel to Serial Converter	0.89	0.89
Komansi -1,1 kode PN	0.016	0.016
Decomading	1.57	1.57
Total	3.756	9.266

Pada Tabel 8, perbedaan terletak pada bagian pemetaan Ik dan Qk bagian Penerima dimana untuk algoritma matematis memerlukan waktu proses 6.78 milidetik sedangkan algoritma *table lookup* hanya memerlukan waktu 1.28 milidetik sehingga waktu proses waktu tunda CDMA dengan algoritma *table lookup* adalah 14.08 milidetik dan angka tersebut masih di bawah waktu tunda yang diizinkan yaitu sebesar 125 milidetik [1].

Kesimpulan

Dari hasil pembahasan di atas, penggunaan Prosesor DSP TMS320C54X pada rangkaian CDMA dapat disimpulkan sebagai berikut :

1. Unjuk kerja yang baik (tidak terjadi kesalahan) menggunakan algoritma *table lookup*.
2. Waktu tunda yang ditimbulkan karena waktu proses yang terbaik menggunakan algoritma *table lookup* yaitu 14.08 milidetik dimana dengan menggunakan algoritma Matematis memerlukan waktu 19.62 milidetik.
3. Waktu tunda 14.08 milidetik masih di bawah waktu tunda yang diizinkan yaitu 125 milidetik.

Daftar Acuan

1. Pusrenbangti PT TELKOM, Strategic Development Plan 1994, PT TELKOM, 1995.
2. W.C.Y. Lee, IEEE Trans. Commun. 40 (1991) 291.
3. Ridhania, Skripsi Sarjana, Jurusan Teknik Elektro Fakultas Teknik Universitas Indonesia, 1995.
4. A. Djohan, Prosiding Cellular Group Seminar (1998).
5. A. Mansoor, Christie, Telecommunication. Texas Instrument Inc., Texas, 1995.
6. S. Purnawan, Skripsi Sarjana, Fakultas Teknik Elektro, Fakultas Teknik Universitas Indonesia, 1997.

7. A. Shahab, Skripsi Sarjana, Fakultas Teknik Elektro, Fakultas Teknik Universitas Indonesia,
8. Microprocessor Development System. TMS320C5x Evaluation Modul Technical Sheet. Texas Instrumens Inc., Texas, 1994.