99-01057

デジタル音声画像伝送処理用、超高速低消費電力VLSIアーキテクチャに関する研究

代表研究者　和田知久　琉球大学工学部情報工学科教授
共同研究者　金田喜共　琉球大学工学部理工学研究科M1
共同研究者　前原崇章　琉球大学工学部理工学研究科M1
共同研究者　山崎和美　琉球大学工学部理工学研究科M1

1. OFDM方式

(ア) OFDMの基本原理

OFDMは複数の搬送波を用いてデータを伝送するマルチキャリア変調方式の１種である。OFDMでは搬送波に直交する正弦波を用いることでスペクトルを重ねることができ、マルチキャリア変調の欠点を解決している。図1にマルチキャリア変調とOFDMの周波数スペクトルを示す。

OFDMは送信する情報によりディジタル変調された複数の直交する副搬送波を多重化して伝送する変調方式である。

① OFDM信号の構成

OFDM信号の基本要素は周波数nf_0(基本周波数f_0のn倍)の正弦波である。OFDMシステムではこの正弦波を副搬送波と呼ぶ。

$$a_n \cos(2\pi n f_0 t) - b_n \sin(2\pi n f_0 t) \quad (1)$$

OFDM信号$S_B(t)$は、式(1)を用いて、式(2)で表される。式(2)はOFDM信号の１シンボル分を表している。

$$S_B(t) = \sum_{n=0}^{N-1} \{a_n \cos(2\pi n f_0 t) - b_n \sin(2\pi n f_0 t)\} \quad (2)$$

式(2)の$S_B(t)$をベースバンドOFDM信号と呼ぶ。図１はN=8の時のベースバンドOFDM信号の例である。n=1～8の副搬送波を加え合わせることで$S_B(t)$を得る。個々の副搬送波はデータによりディジタル変調されているので各々振幅と位相が異なっている。

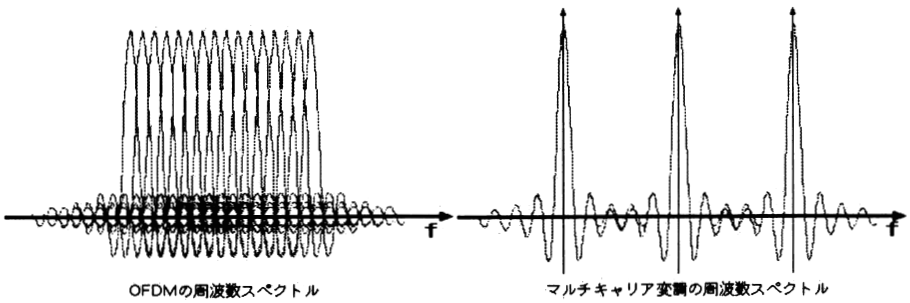


図1　OFDMとマルチキャリア変調の周波数スペクトル

② 副搬送波の変調

OFDMの副搬送波はデータシンボルによりディジタル変調されている。OFDMで用いるディジタル変調はPSK変調、QAM変調である。図２にPSK変調、QAM変調のコンスタレーションマップの例を示す。

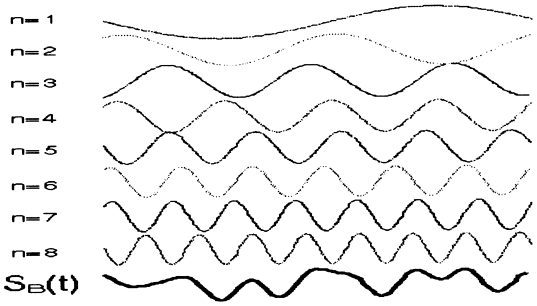


図2　OFDM信号の例

③ OFDM信号の生成と復調

マルチキャリア変調では副搬送波の数だけの変調器が必要であるが、ＯＦＤＭでは以下の方法により、DFT(Discrete Fourier Transform)アルゴリズムを用いることができる。よって、OFDMは数千個にもおよぶ副搬送波を用いた通信が可能である。

式(2)はオイラーの公式を用いて式(3)のように変換することができる。

$$S_B(t) = \mathrm{Re}[u(t)]$$

$$u(t) = \sum_{n=0}^{N-1} d_n e^{j2\pi n f_0 t} \qquad \text{ただし} \quad d_n = a_n + jb_n \tag{3}$$

ここで、$u(t)$を$1/(Nf_0)$の標本化間隔でサンプリングすると式(4)を得る。

$$u\left(\frac{k}{Nf_0}\right) = \sum_{n=0}^{N-1} d_n \left(e^{j2\pi \frac{k}{Nf_0}}\right)^{nk} \qquad (k = 0,1,2,\ldots,N-1) \tag{4}$$

式(4)はN個の複素データシンボルd_nをIDFT(Inverse Discrete Fourier Transform)した形である。式(4)によって得られる系列を連続信号にすることで$u(t)$を生成でき、$u(t)$の実部をとることで$S_B(t)$を生成できる。

また、OFDM信号の復調は、逆にDFTを用いる。式(5)のようにをサンプリングした系列をDFTすることでデータシンボルd_lを復調することができる。

$$d_l = \frac{1}{N}\sum_{n\sim 0}^{N-1} u\left(\frac{k}{Nf_0}\right) e^{-j\frac{2\pi kt}{N}} \qquad (l = 1,2,3,\ldots,N-1) \tag{5}$$

このようにして、OFDMはDFTアルゴリズムを用いてOFDM信号の生成、データシンボルの復調を行うことができる。

④ ガードインターバルの付加

マルチパス伝送路でOFDM信号を受信すると、遅延波によりシンボル間干渉を引き起こし、データを正しく復調できなくなってしまう。OFDMは、送信シンボルにガードインターバルを付加することによってマルチパスの影響を軽減している。

図４にガードインターバルの概念を示す。ガードインターバルはシンボルの最後の部分をコピーしてシンボルの前部に付加したものであり、シンボルの長さを理論値よりも長くすることができる。OFDMでは各副搬送波の周波数が基本周波数の整数倍になっているためシンボルの最後の部分を先頭に付加しても接続部分に不連続のない信号としてシンボル長を伸ばすことができる。ガードインターバルを付加したOFDMシンボルはガードインターバル長以内の遅延波であれば、シンボル間干渉を防ぐことができる。その様子を図５に示す。

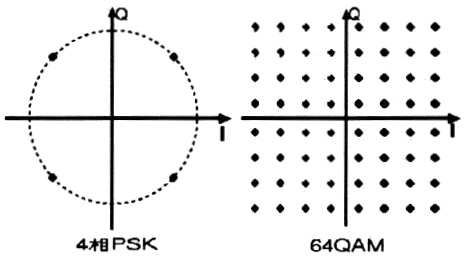


図3　副搬送波の変調方式の例

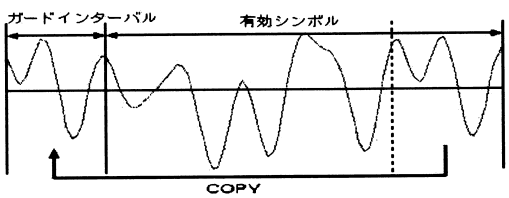


図4　ガードインターバルの概念

⑤ パイロットシンボルの挿入

ガードインターバルの効果によりシンボル間干渉は防ぐことができるが、受信シンボルは振幅と位相にひずみを受ける。よって、受信機では図６のように各信号点の位相が回転し、振幅が変動した信号が受信される。このひずみは

伝送路特性を推定することで容易に補正をすることができる。

OFDMでは伝送路特性の推定にパイロットシンボルを用いる。パイロットシンボルとは送信機、受信機で共に既知であるシンボルのことであり、データシンボルの間に挿入する。受信機ではまず、受信されたパイロットシンボルからパイロットシンボルに作用する伝送路特性を推定する。推定された伝送路特性を用いてデータシンボルに対する伝送路特性を推定し、データシンボルのひずみを補正する。

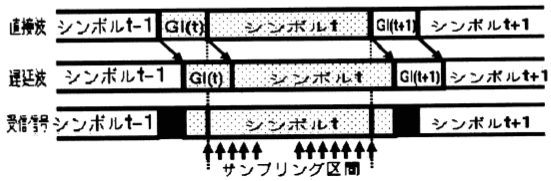


図5　ガードインターバルの効果

(イ) OFDMシステムの構成

OFDMシステムの構成を図７に示す。図７の構成は主要な要素のみの簡略化したものであり、実際のシステムはその他にも構成要素がある。

送信機では、何らかのデータが入力されるとまず、「マッピング」により各副搬送波をディジタル変調するための複素シンボル列に変換する。得られた複素シンボル列は「直並列変換機」でDFTサイズのパラレル信号に変換され「IDFT」でIDFTされる。IDFT後はパラレル信号なので「並直列変換機」でシリアル信号に変換し、複素ベースバンドOFDM信号を得る。「GI挿入」で各シンボルにガードインターバルを挿入する。アップコンバージョン処理を行って、伝送路へ送信する。

受信機では、まず、「フロントエンド」で受信した信号をダウンコンバージョン処理する。「A/D」で離散時間信号を生成し、「GI削除」でガードインターバルを削除する。その後、「直並列変換器」で各シンボルごとのパラレルデータを生成し、「DFT」でDFT処理を行う。「並直列変換器」でシリアル信号に変換し、「デマッピング」により得られたデータの判別を行う。

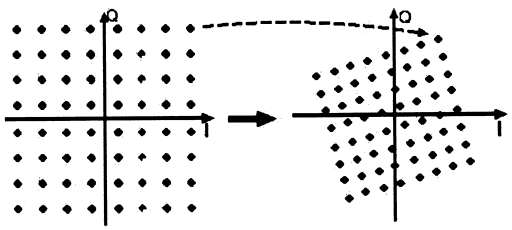


図6　受信信号のひずみ（64QAM）

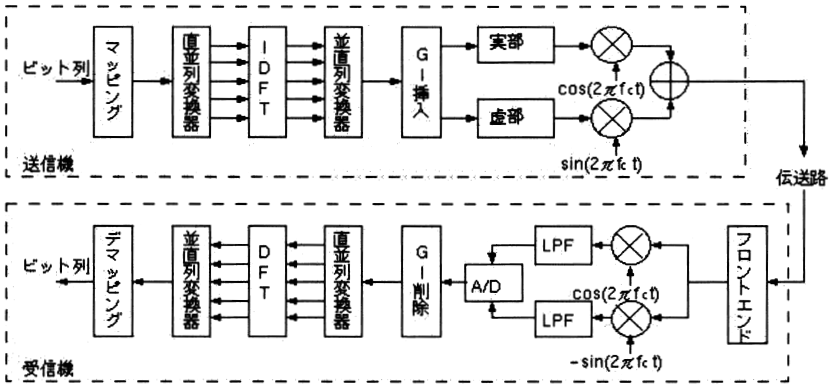
2. FFTアルゴリズム

FFTは離散フーリエ変換（DFT: Discrete Fourier Transform）における計算のむだを省くため三角関数の周期性を利用した計算技法である。FFTはDFTと比べて大幅に計算回数を減らすことができるため、OFDMではDFT演算を行う部分でFFTアルゴリズムを用いる。FFTアルゴリズムには現在Radix-2,4,8の３つが検討されている。ここではそれぞれの構成を説明し、性能の比較を行う。

(ア) Radix-2

Radix-2はDFTを２入力の計算に分解して演算を行う。DFTの式を式（６）のように変形する。

$$
\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(n) W_N^{nk} + \sum_{n=0}^{\frac{2}{2}N-1} x(n) W_N^{nk} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(n) W_N^{nk} + W_N^{\frac{N}{2}k} \sum_{n=0}^{\frac{N}{2}-1} x\left(n+\frac{N}{2}\right) W_N^{nk}
\end{aligned}
\qquad (6)
$$



ここで、三角関数の周期性より$W_N^{\frac{N}{2}} = e^{-j(2\pi/N)\frac{N}{2}}$ =-1となるので式（６）は式（７）となる。

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} \left\{ x(n) + (-1)^k x\left(n + \frac{N}{2}\right) \right\} W_N^{nk} \qquad (7)$$

導出された式（７）により図8のようなRadix-2演算器が構成される。

Radix-2での分解回数は、$\log_2 N$回となり、これはFFTの演算回数に相当する。

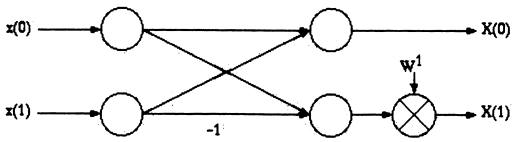


図8　Radix-2演算器

(イ) Radix - 4

Radix-4はRadix-2の拡張で、４入力での演算を行う。離散フーリエ変換の式より式（８）のRadix-4演算が導き出すことができる。

$$X(k) = \sum_{n=0}^{\frac{N}{4}-1} \left\{ x(n) + (-j)^k x\left(n + \frac{N}{4}\right) + (-1)^k x\left(n + \frac{2}{4}N\right) + (j)^k x\left(n + \frac{3}{4}N\right) \right\} W_N^{nk} \qquad (8)$$

式(8)により、図9のようなRadix-4演算器が構成される。

Radix-4の分解回数は、$\log_4 N$となる。

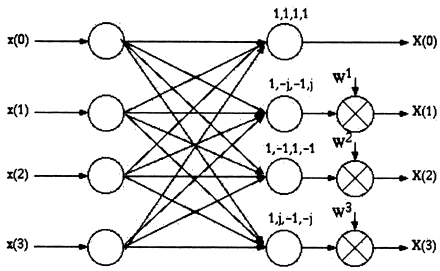


図9　Radix-4演算器

(ウ) Radix- 8

Radix-2、Radix-4と同じように離散フーリエ変換の式より、式（９）のRadix-8演算を導き出すことができる。

$$X(k)=\sum_{n=0}^{\frac{N}{8}-1}\left\{\begin{array}{l} x(n)+(\frac{1}{\sqrt{2}}-j\frac{1}{\sqrt{2}})^k x\left(n+\frac{N}{8}\right) \\ +(-j)^k x\left(n+\frac{2}{8}N\right)+(-\frac{1}{\sqrt{2}}-j\frac{1}{\sqrt{2}})^k x\left(n+\frac{3}{8}N\right) \\ +(-1)^k x\left(n+\frac{4}{8}N\right)+(-\frac{1}{\sqrt{2}}+j\frac{1}{\sqrt{2}})^k x\left(n+\frac{5}{8}N\right) \\ +(j)^k x\left(n+\frac{6}{8}N\right)+(-\frac{1}{\sqrt{2}}+j\frac{1}{\sqrt{2}})^k x\left(n+\frac{3}{8}N\right) \end{array}\right\} W_N^{nk} \qquad (9)$$

式(8)により、図10のようなRadix-8演算器が構成される。

Radix-8の分解回数は、$\log_8 N$となる。

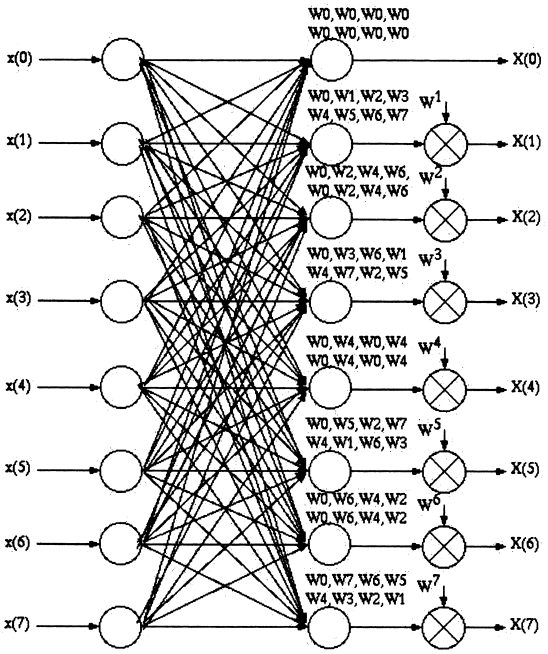


図10　Radix-8演算器

(エ) 回路構成

FFT演算は主に複素数を扱う。実際のハードウェアでは複素数をと表現し二項式として扱うことにより演算を実行している。

$$\begin{aligned}(a+jb)+(c+jd)&=(a+c)+j(b+d)\\(a+jb)\times(c+jd)&=(a\times c-b\times d)+j(a\times d+b\times c)\end{aligned} \qquad (10)$$

式(10)により、複素数加算は２つの実数加算によって構成され、複素数乗算は４つの実数乗算と２つの実数加算で構成される。

また、複素数乗算回路は式(11)に書き換えることが可能である。

$$(a+jb)\times(c+jd)=[a\times(c+d)-(a+b)\times d]+j[b\times(c-d)+(a+b)\times d] \qquad (11)$$

式(11)の複素数乗算では３つの実数乗算と３つの実数加算で構成することができる。式(11)の回路構成を図11に示す。

乗算はハードウェア規模が大きく、計算時間もかかる。図11(a)は図11(b)よりも実数乗算が１つ少なくなっているためハードウェア規模と計算時間の縮小が実現されている。

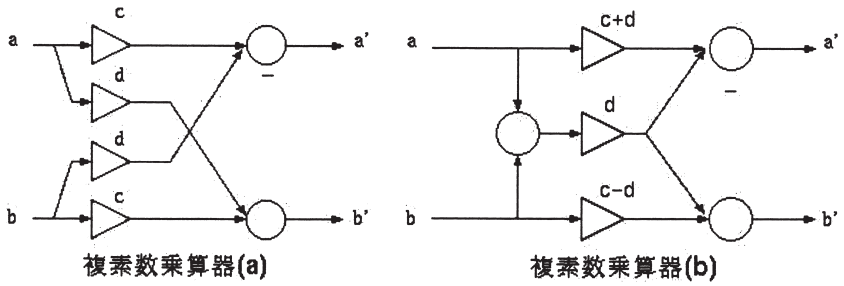


図11複素数乗算器の構成

(オ) 回路規模

表1にRadix-2,4,8における回路規模をまとめた。表1におけるαは実数乗算器の回路規模に対する$1/\sqrt{2}$-乗算器の回路規模比率であり、表2にαに対するRadix-8の乗算器数を示す。Radix-2,4,8の分解回数はそれぞれ$\log_2 N$, $\log_4 N$, $\log_8 N$となるため、総PE数は　Radix-8が最小となる。PEとは図8,9,10の演算器を構成するハードウェアのことをいう。

しかし、１つのPEで処理する入力データ数が多くなるにつれ、その演算がより複雑になり、PEの構成が複雑になる。さらに、Radix-8では$1/\sqrt{2}$-乗算器を考慮に入れなければならない。表1,2により、α=0.1を達成できればRadix-8が最も効率のよいものとなる。

表1　Radix-2,4,8回路規模比較

Radix		2	4	8
Real-Add		7	33	197
Real-Mul		3	9	21
$1/\sqrt{2}$-Mul		0	0	32
PE 数	64FFT	192	48	16
	4kFFT	24k	6k	2k
Mul 数, (PE×Real-Mul)+(PE×$1/\sqrt{2}$-Mul)	64FFT	576	432	330+512×α
	4kFFT	72k	54k	42k+64k×α

表2　Radix-8乗算回路数

α			0.5	0.1
Mul 数	64FFT	330+512×α	586	381
	4kFFT	42k+64k×α	74k	48k

3. RS符号化アルゴリズム

RS符号は何ビットかをひとつの単位として扱う。そのひとつの単位をα^iと表現し、これをシンボルと呼ぶ。RS符号の能力は(n,k)と表し、nは符号全体のシンボル数であり、kは情報シンボルである。また、この二つの差n-kはエンコードによって付加される冗長シンボル数となる。

(ア) RS符号の生成

G(z)を生成多項式とすると、G(z)は式(12)のように定義される。

$$G(z)=(z-\alpha)(z-\alpha^2)(z-\alpha^3)\cdots(z-\alpha^{2t}) \tag{12}$$

tはエラー訂正可能な最大シンボル数であり、符号の最小距離dがd=2t+1の場合、t=(n-k)/2となる。

生成多項式G(z)を用いて式(13)のようにして冗長シンボルを生成する。I(z)を情報多項式、C(z)を冗長シンボルとする。

$$C(z)=\{I(z)\cdot z^{2t}\}\bmod G(z) \tag{13}$$

RS符号化したシンボルを符号多項式と呼び、W(z)とすると、W(z)は式(14)で生成する。W(z)はG(z)で割り切れる

形になる。(Q(z)はW(z)をG(z)で割った商とする。)

$$W(z)=\{I(z)\cdot z^{2t}\}-C(z)=G(z)Q(z) \tag{14}$$

符号多項式W(z)を図で表すと図12のようになる。情報シンボルをそのままシフトさせ、そこへ冗長シンボルを詰めた形になることがわかる。

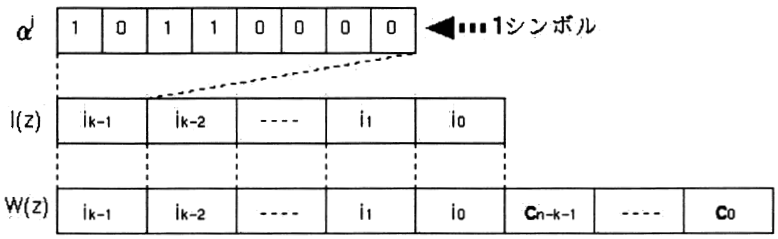


図12　RS符号

(イ) RS符号の復号

RS符号の復号は以下のような手順で行われる。

1．受信信号R(z)から、シンドローム多項式S(z)を求める。

2．シンドローム多項式S(z)から誤り位置多項式σ(z)と誤り数値多項式ω(z)を求める。

3．誤り位置多項式σ(z)から誤り位置を求める。

4．誤り位置多項式σ(z)、誤り数値多項式ω(z)および3で求めた誤り位置から誤り数値を求める。

① シンドローム多項式の生成

式(14)で生成された符号多項式W(z)は伝送路で誤りE(z)が加わり、受信語R(z)を得る。RS復号器はこの受信語R(z)(=W(z)+E(z))を入力とする。

復号器では受信語R(z)にG(z)=0の解を代入すると、式(15)となり、伝送路で加わった誤りのみに依存する値が得られる。

$$s_{i-1}=R(\alpha^i)=W(\alpha^i)+E(\alpha^i)=E(\alpha^i)\qquad (i=1,2,3,\cdots,2t) \tag{15}$$

これをシンドロームと呼び、S_{i-1}を係数として持つ多項式をシンドローム多項式と呼ぶ。シンドローム多項式S(z)は式(16)のようになる。

$$S(z)=s_0+s_1z+s_2z^2+\cdots+s_{2t-1}z^{2t-1} \tag{16}$$

② 誤り位置多項式と誤り数値多項式の生成

伝送路で加わったエラー多項式E(z)を式(17)のように定義する。

$$E(z)=e_{j_k}z^{j_k}\qquad (0\le j_k<n-1,k=1,2,\cdots u) \tag{17}$$

この式はnシンボル中に個のシンボルに誤りが生じたとして、その誤り位置Z^{jk}と、それに対する誤り数値e_{jk}を多項式で表したものである。

誤り位置多項式σ(z)、誤り数値多項式ω(z)は式(18)，(19)のように定義される。

$$\sigma(z)=\prod_{k=1}^{u}(1-\alpha^{i_k}z) \tag{18}$$

$$\omega(z)=\sum_{l=1}^{u}e_{i_l}\alpha^{i_l}\prod_{k=1}^{u}(1-\alpha^{i_k}z) \tag{19}$$

誤り位置多項式と誤り位置多項式はシンドローム多項式と式(20)のような関係を持つ。

$$\sigma(z)\cdot S(z)=\omega(z)\bmod z^{2t} \tag{20}$$

式(20)にS(z)が与えられたとき、式(18)，(19)のσ(z)、ω(z)を満たす多項式は一組しかない。このような関係を満たす多項式の組み合わせは拡張ユークリッド互除法で求められる。

③ 誤り位置と誤り数値の計算

誤り位置は誤り位置多項式を用いて計算する。誤り位置多項式σ(z)は定義より誤り位置を解として持つため、$\sigma(\alpha^j)=0$となるような解を求めることで誤り位置を得ることができる。

誤り位置がわかると、$\sigma(\alpha^{-i'})=0$となる$\alpha^{i'}$に対して式(21)を実行すると、誤り数値が得られる。

$$e_{i'}=-\frac{\omega(\alpha^{-i'})}{\sigma'(\alpha^{i'})} \tag{21}$$

(ウ) 実装回路の構成

① RSエンコーダ

RSエンコーダは生成多項式G(z)の係数を用いて図13のように構成される。

まず、全てのレジスタをリセットし、MUXは ’0’ をセレクトする。次に、k個の情報シンボルを一つずつレジスタに入力し、同時に出力していく。k個入力し終わった段階ではレジスタには入力された情報シンボルをG(z)で割った剰余つまりC(z)が蓄えられている。ここで、MUXを ’1’ に切り替えて順次出力すると、情報シンボルに冗長シンボルを付加した符号化したシンボルが得られる。

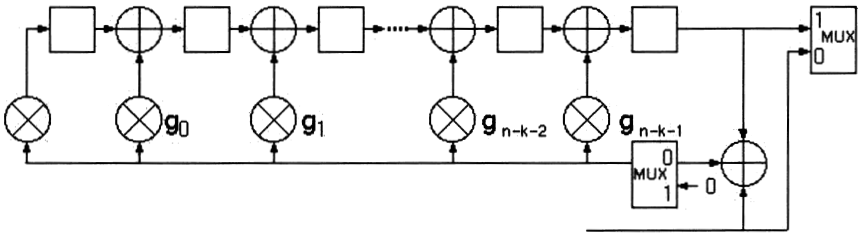


図13　RSエンコーダ

② RSデコーダ

1. シンドローム生成回路

シンドロームは受信語R(z)に、G(z)=0となる解を代入して得られる。この計算は式(22)で求められる。

$$R(\alpha^i) = r_0 + r_1(\alpha^i) + r_2(\alpha^i)^2 + \cdots + r_{n-1}(\alpha^i)^{n-1} + r_n(\alpha^i)^n$$

$$= (((\cdots(r_n\alpha^i + r_{n-1})\alpha^i \cdots + r_2)\alpha^i + r_1)\alpha^i + r_0$$

$$(i = 1,2,3,\cdots,2t) \qquad (22)$$

式(22)の動作は図14の回路で実現でき、この回路を2t個用意することで全てのiに対してのシンドロームを求めることができる。

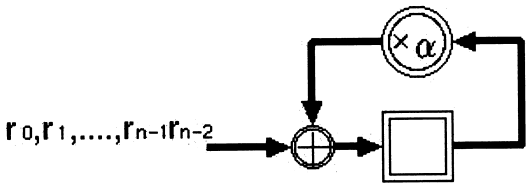


図14　シンドローム回路

2. ユークリッド演算回路

ユークリッド演算回路では、式(20)の関係を全て調べ、誤り位置多項式と誤り数値多項式を求める。この回路では2t個のシンドロームからt個の誤り位置情報とt個の誤り数値情報が得られる。

3. 誤り位置演算回路

誤り位置の計算は$\sigma(\alpha^{-i})$の剰余を求める計算に等しい。

まず、初期値として図15の各レジスタに誤り位置多項式の係数をセットする。この状態から計算を実行すると$(\sigma(\alpha^{-i})-\sigma_0)$が順次計算される。　これは、チェンサーチ回路と呼ばれている。

4. 誤り数値演算回路

誤り数値の計算は誤りのある個所$\alpha^{i'}$に対して式(21)を行う。ここで、$w(\alpha^{-i'})$は図15のチェンサーチ回路を用いる。$\sigma(\alpha^{-i})$は図16の回路を用いる。

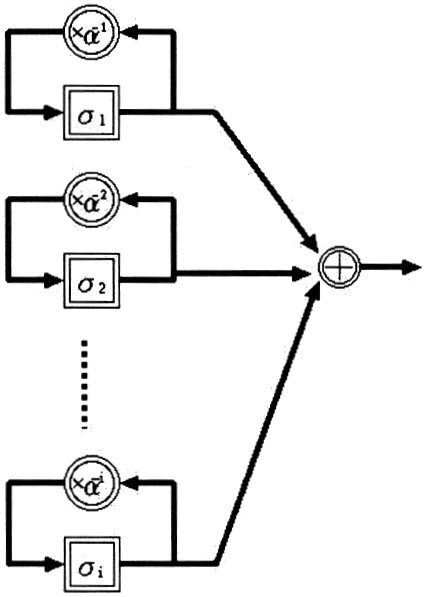


図15　チェンサーチ回路

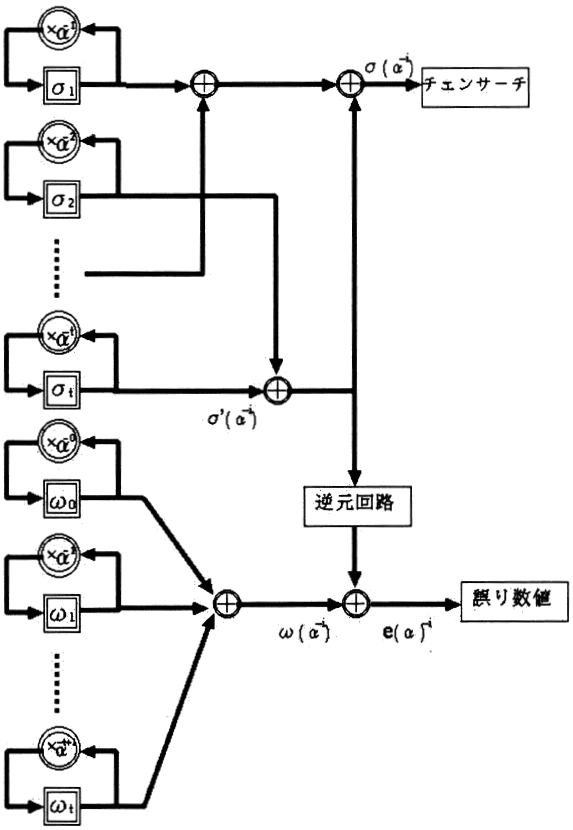


図16　誤り数値演算回路

＜ 発　表　資　料 ＞

題　　名	掲載誌・学会名等	発表年月
（案）OFDM receiver LSI for Mobil Rayleigh Fading Application	IEEE International Solid-State Circuit Conference 2002 投稿予定	2002年２月