

# 誤り訂正符号方式における訂正不能符号 の即時処理

豊田 準三・小西 憲一

岡山理科大学 電子理学科

(昭和60年9月26日 受理)

## まえがき

伝送中に生じる誤りについての自働訂正機能を有する符号系では受信側での復号結果には訂正機能を超えた誤りのデータの混入はさけられない。これを正常な復号結果と区別するには受信後に探表操作を含む厄介な作業を再処理として行うか、ARQ(自働再送要求)システムで対処するのが通常であるが、いずれにしても多数の受信データの中からの特殊データのとりだしということで、能率のわるさはさけられない。本文では1, 2誤訂正程度の低訂正能力の符号系について、訂正不能符号のとりだしを受信中での即時処理として行うことを検討した。ここではある符号系の符号群を論理関数として取扱い、復号過程を同期式順序回路として取扱っているが、訂正不能受信符号に関係した論理設計には図形論理演算による設計手法を導入することにより簡潔な回路構成がえられ、それにより即時処理が可能になることを示した。

## 1. ハミング符号の論理関数としての取扱い

ハミング符号は誤り訂正符号の元祖といえるものであるが、情報符号数  $h=4$ 、検査符号数  $m=3$  のものを引用して、ここでは前者には2値論理変数  $A, B, C$ 、および  $D$  を、後者には  $X, Y$ 、および  $Z$  をあたえ、符号語長  $n=k+m=7$  のものを7変数の論理関数として取扱うことにする。これは誤りがひとつしか生じないとした場合に自動的に訂正が可能な性質を有する符号であるが、検査方式が偶数パリティ方式と奇数パリティ方式のものについて表1に示してある。表中の等価数は各符号を2進数とみなした場合の10進数値表現である。これはまた、それぞれの表を7変数論理関数の積和形の真理値表として取扱い、その展開形を付図に示した区画番地を用いて、図1また図2の右上の最小項図(カルノ図)のように図形表示した場合の番地にも対応したものである。この等価数は偶数パリティの場合の最小項展開形を  $HE(m)$ 、奇数パリティの場合を  $HD(m)$  として、次のような展開で示した場合の各項に用いる添字数にも一致しているものでもある。(式中+は論理和記号)

$$HE(m) = m_0 + m_7 + m_{25} + m_{30} + m_{42} + m_{45} + m_{51} + m_{52} + m_{75} + m_{76} + m_{82} + m_{85} + m_{97} + m_{102}$$

$$+ m_{120} + m_{127} \tag{1}$$

$$HD(m) = m_{11} + m_{12} + m_{18} + m_{21} + m_{33} + m_{38} + m_{56} + m_{63} + m_{64} + m_{71} + m_{89} + m_{94} + m_{106} + m_{109} + m_{115} + m_{116} \tag{2}$$

それぞれの最小項図をみると、16区画単位毎にハミングの距離3が確保されていることがわかる。

符号名	A	B	C	X	D	Y	Z	等価数	符号名	A	B	C	X	D	Y	Z	等価数
HE0	0	0	0	0	0	0	0	00	HD0	0	0	0	1	0	1	1	11
HE1	0	0	0	0	1	1	1	07	HD1	0	0	0	1	1	0	0	12
HE2	0	0	1	1	0	0	1	25	HD2	0	0	1	0	0	1	0	18
HE3	0	0	1	1	1	1	0	30	HD3	0	0	1	0	1	0	1	21
HE4	0	1	0	1	0	1	0	42	HD4	0	1	0	0	0	0	1	33
HE5	0	1	0	1	1	0	1	45	HD5	0	1	0	0	1	1	0	38
HE6	0	1	1	0	0	1	1	51	HD6	0	1	1	1	0	0	0	56
HE7	0	1	1	0	1	0	0	52	HE7	0	1	1	1	1	1	1	63
HE8	1	0	0	1	0	1	1	75	HD8	1	0	0	0	0	0	0	64
HE9	1	0	0	1	1	0	0	76	HD9	1	0	0	0	1	1	1	71
HEA	1	0	1	0	0	1	0	82	HDA	1	0	1	1	0	0	1	89
HEB	1	0	1	0	1	0	1	85	HDB	1	0	1	1	1	1	0	94
HEC	1	1	0	0	0	0	1	97	HDC	1	1	0	1	0	1	0	106
HED	1	1	0	0	1	1	0	102	HDD	1	1	0	1	1	0	1	109
HEE	1	1	1	1	0	0	0	120	HDE	1	1	1	0	0	1	1	115
HEF	1	1	1	1	1	1	1	127	HDF	1	1	1	0	1	0	0	116

偶数パリティ ハミング符号

奇数パリティハミング符号

表1. ハミング符号

誤りが混入した受信符号は、これら16種の項による展開形以外の240種類の展開項のいづれかの形で受信されることになるが、それらは(1)または(2)の否定形で示すことができる。モルガンの定理による否定で和積形になった否定関数をそれぞれ  $\overline{HE}(M)$ ,  $\overline{HD}(M)$  で示すことにすると、それぞれの最大項展開項は次のようになる。

$$\overline{HE}(M) = M_{127}M_{120}M_{102}M_{97}M_{85}M_{82}M_{76}M_{75}M_{52}M_{51}M_{45}M_{42}M_{30}M_{25}M_7M_0 \tag{3}$$

$$\overline{HD}(M) = M_{116}M_{115}M_{109}M_{106}M_{94}M_{89}M_{71}M_{64}M_{63}M_{56}M_{38}M_{33}M_{21}M_{18}M_{12}M_{11} \tag{4}$$

番地を形成する変数域をとりかえ、区画間の論理結合を論理積結合にとりかえた最大項図上で、(3), (4)を図形表示するについては、位置はそのまゝにして最小項を示す記号・を最大項を示す記号+にとりかえるだけでよく、これも図1, 図2に示してある。<sup>1)</sup>

最大項図での論理関数の図形による論理和合成では、最小項図(カルノ図)の場合とは

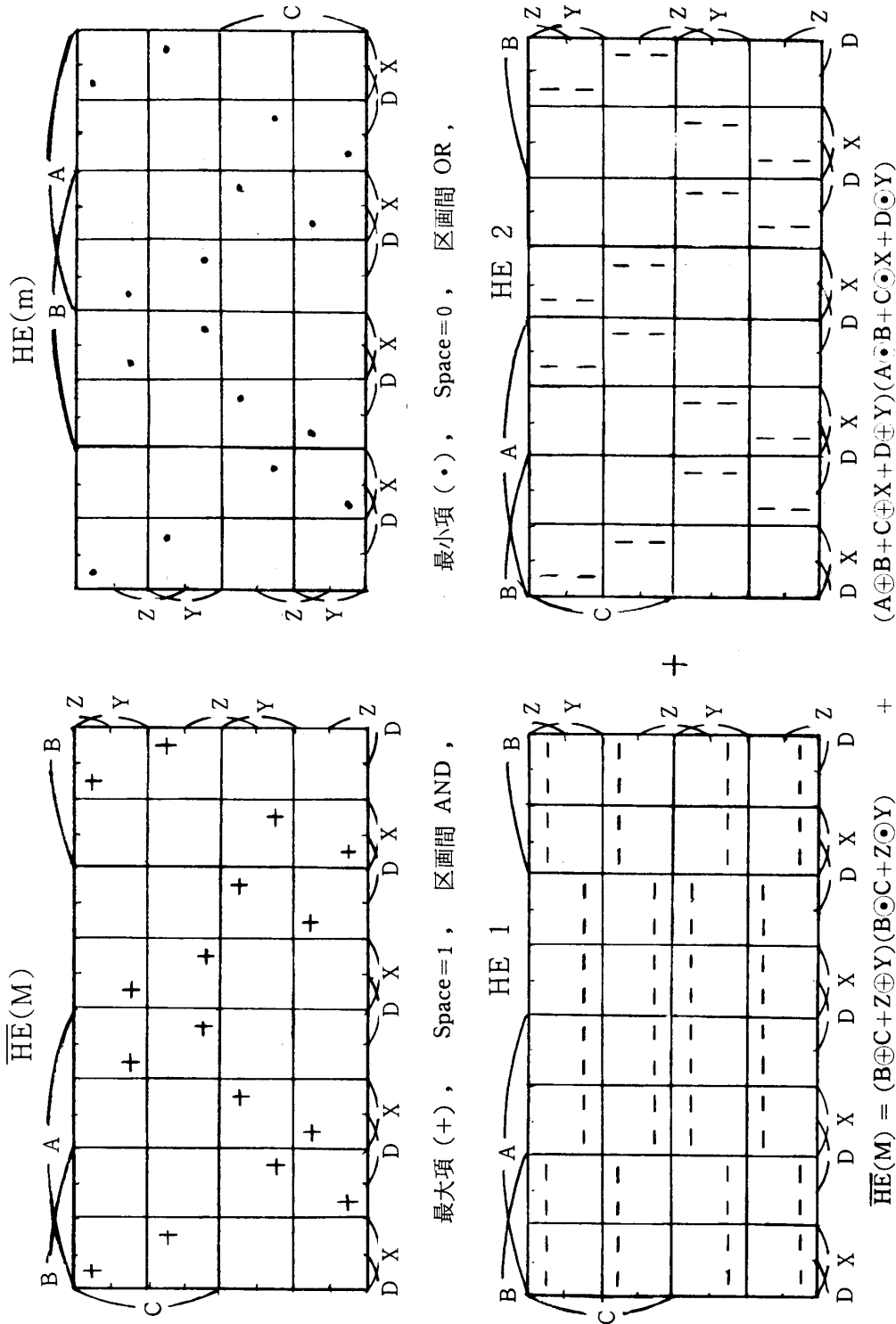


図1 偶数パリティハイミング符号の論理関数合成

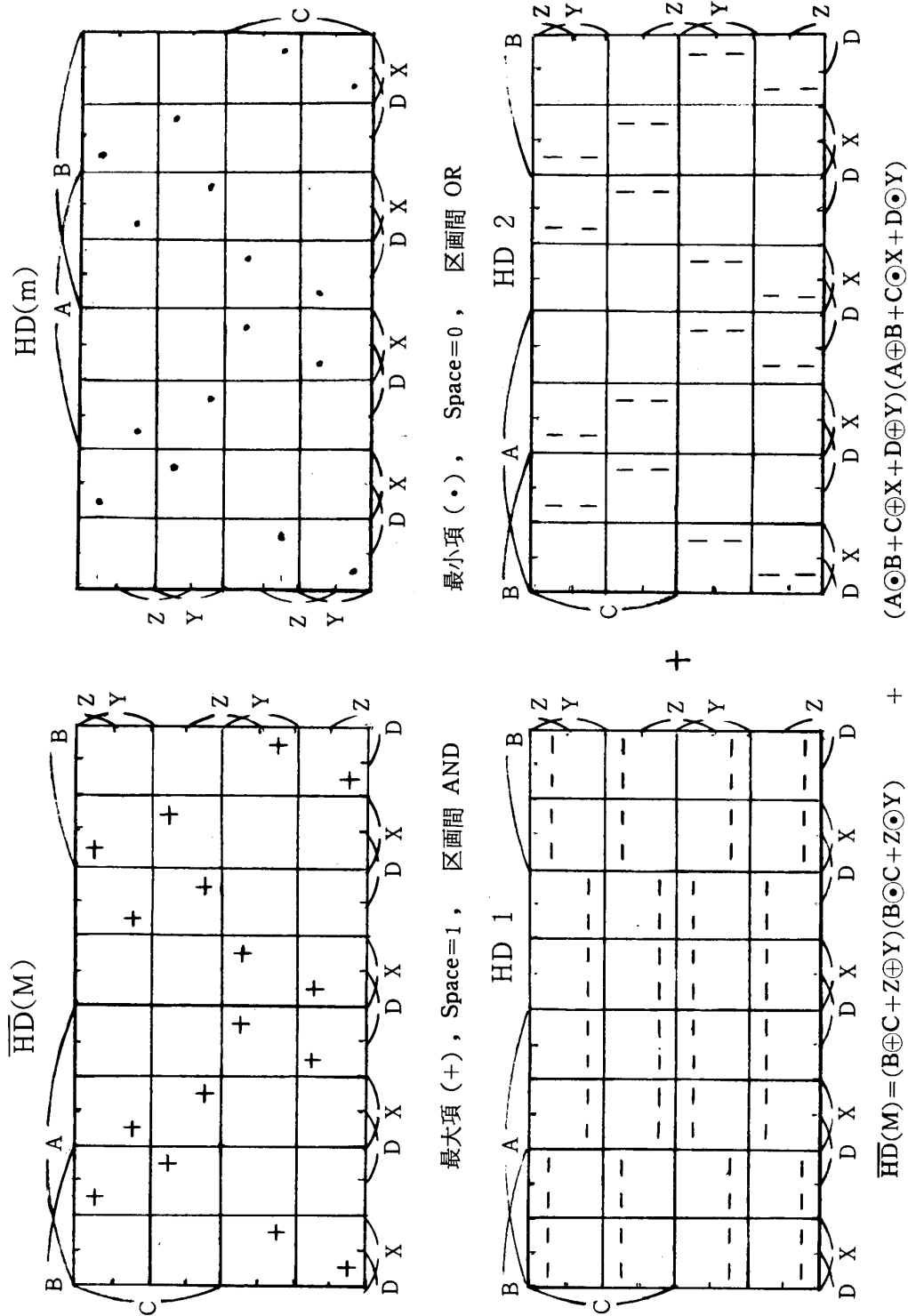


図2 奇数パリティハイミング符号の論理関数合成

双対的で、成分関数の+記号が一致しているところだけが残ることになるので、積和形では展開項数の多い関数については和積形をとりあげて最大項図で取扱うのが好都合である。<sup>2)</sup> 最大項図でのこの図形演算特性を起用すると、図1、図2のそれぞれに示したような和積形関数の分解と合成が成立する。この場合の成分関数はさらにまた排他論理和( $\oplus$ )と対等論理和( $\odot = \overline{\oplus}$ )の成分の組合せに帰着できることになるが、この場合の論理演算式は次のようになる。\*

$\overline{HE}(M) = HE1 + HE2$  とすると

$$\begin{aligned} HE1 &= (A+B+C+Y\oplus Z)(A+B+\overline{C}+Y\odot Z)(A+\overline{B}+C+Y\odot Z) \\ &\quad (A+\overline{B}+\overline{C}+Y\oplus Z)(\overline{A}+\overline{B}+C+Y\odot Z)(\overline{A}+\overline{B}+\overline{C}+Y\oplus Z) \\ &\quad (\overline{A}+B+C+Y\oplus Z)(\overline{A}+B+\overline{C}+Y\odot Z) \\ &= (B+C+Y\oplus Z)(\overline{B}+\overline{C}+Y\oplus Z)(\overline{B}+C+Y\odot Z)(B+\overline{C}+Y\odot Z) \\ &= (B\oplus C+Y\oplus Z)(B\odot C+Y\odot Z) \end{aligned} \quad (5)$$

$$\begin{aligned} HE2 &= (A+B+C+X+D\oplus Y)(A+B+\overline{C}+\overline{X}+D\oplus Y) \\ &\quad (A+\overline{B}+C+\overline{X}+D\odot Y)(A+\overline{B}+\overline{C}+X+D\odot Y) \\ &\quad (\overline{A}+\overline{B}+C+X+D\oplus Y)(\overline{A}+\overline{B}+\overline{C}+\overline{X}+D\oplus Y) \\ &\quad (\overline{A}+B+C+\overline{X}+D\odot Y)(\overline{A}+B+\overline{C}+X+D\odot Y) \\ &= (A\oplus B+C+X+D\oplus Y)(A\oplus B+\overline{C}+\overline{X}+D\oplus Y) \\ &\quad (A\odot B+C+\overline{X}+D\odot Y)(A\odot B+\overline{C}+X+D\odot Y) \\ &= (A\oplus B+C\oplus X+D\oplus Y)(A\odot B+C\odot X+D\odot Y) \end{aligned} \quad (6)$$

$$\begin{aligned} \text{従って } \overline{HE}(M) &= (B\oplus C+Y\oplus Z)(B\odot C+Y\odot Z) \\ &\quad + (A\oplus B+C\oplus X+D\oplus Y)(A\odot B+C\odot X+D\odot Y) \end{aligned} \quad (7)$$

がえられる。従ってこれを否定して

$$\overline{\overline{HE}(M)} = HE(m) \quad (8)$$

がえられる。また  $\overline{HD}(M)$  については

$\overline{HD}(M) = HD1 + HD2$  として次のようにえられる。

$$\begin{aligned} HD1 &= (C+B+Y\oplus Z)(\overline{C}+B+Y\odot Z)(C+\overline{B}+Y\odot Z)(\overline{C}+\overline{B}+Y\oplus Z) \\ &= (B\oplus C+Y\oplus Z)(B\odot C+Y\odot Z) \end{aligned} \quad (9)$$

$$\begin{aligned} HD2 &= (A+B+C+\overline{X}+Y\odot D)(A+B+\overline{C}+X+Y\odot D) \\ &\quad (A+\overline{B}+C+X+Y\oplus D)(A+\overline{B}+\overline{C}+\overline{X}+Y\oplus D) \\ &\quad (\overline{A}+\overline{B}+C+\overline{X}+Y\odot D)(\overline{A}+\overline{B}+\overline{C}+X+Y\odot D) \\ &\quad (\overline{A}+B+C+X+Y\oplus D)(\overline{A}+B+\overline{C}+\overline{X}+Y\oplus D) \\ &= (A+B+C\odot X+Y\odot D)(A+\overline{B}+C\oplus X+Y\oplus D) \end{aligned}$$

\* 成分関数にも+記号を用いるべきところであるが、縦と横との重なっているところが合成関数であることがわかりやすい記号を用いた。また重ならないところは、無定義項を簡単化のために起用したことになる。

$$\begin{aligned} & (\bar{A} + \bar{B} + C \odot X + Y \odot D)(\bar{A} + B + C \oplus X + Y \oplus D) \\ & = (A \oplus B + C \odot X + Y \odot D)(A \odot B + C \oplus X + Y \oplus D) \end{aligned} \quad (10)$$

$$\begin{aligned} \overline{HD}(M) & = (C \oplus B + Y \oplus Z)(C \odot B + Y \odot Z) \\ & + (A \oplus B + C \odot X + Y \odot D)(A \odot B + C \oplus X + Y \oplus D) \end{aligned} \quad (11)$$

このようにいずれについても排他論理和関数を基調とした成分関数の合成としてえられる。 $\overline{HE}(M)$ については図4中に示したように排他論理和素子5個に否定素子5, NOR素子6, およびOR素子1個が付加された簡潔な論理回路構成としてえられることになる。一般に積和形では多数の展開項となる関数(上記の否定を積和形で求めると240項の最小項展開となる。)は和積形として最大項図上での図形演算で処理すると簡潔な論理設計が実現できることが多い。このようにしてえられた否定形 $\overline{HE}(m)$ を否定すると元の積和形関数 $HE(m)$ がえられるが,これを直接積和形として構成すると,多数個の前置素子を含む7入力のNAND素子16個と16入力のNAND素子1個が必要になることを注意しておきたい。

このようにしてえられた論理関数は元のハミング符号に誤りの混入したもののすべて(240種)を含むものであるが,単一誤りの訂正動作が正常であれば,  $7 \times 16 = 112$ 種は訂正可能のものである。誤りがなかったもの16種とあわせて取除いた残りの128種のものである。この符号系が有している自働訂正機能を越えた誤りの結果として生じるものである。そのなかには検査に合格して誤りが訂正できたと誤認されるようなものもある。例えば表1の符号HECが伝送される途中で

$$HEC = 1100001 \rightarrow 1110000 \quad (12)$$

のように右端のものが0になり,5番目ものが1となるような逆転した2重誤りは,この符号系の正常な訂正動作では第4番目を逆転して1111000がえられることになる。(次節に記述)これは表1ではHEEであり,このように誤認による復号結果としてえられる誤りも含まれている。このような誤りの受信符号を復号時点で即時分離を行うについては関数 $\overline{HE}(M)$ の論理回路出力が起用される。

## 2. ハミング符号における訂正可能な誤りと訂正不能の誤り

表1のように変数割り当てされた偶数パリティのハミング符号がZをビット番号1として,これを先頭にして送出され,この順番で受信されたとすると,誤り訂正のための論理式(検査方程式)は次のように示される。

$$(Z' =) \quad Z \oplus D \oplus C \oplus A = 0 \quad (13)$$

$$(Y' =) \quad Y \oplus D \oplus B \oplus A = 0 \quad (14)$$

$$(X' =) \quad X \oplus C \oplus B \oplus A = 0 \quad (15)$$

ひとつの誤りがある場合はこの論理演算の結果の $X'$ ,  $Y'$ , または $Z'$ のどれかが1となるのであるが,それに与えられる重みを $Z'$ は1,  $Y'$ は2,  $X'$ は4としておくとそれら

の代数和で誤りのビット番号がわかることになる。例えば  $0100010 (=abcxdyz)^*$  (16) が右端を先頭に受信された場合を考えると、その検査演算は次のようになる。

$$z' = 0 \oplus 0 \oplus 0 \oplus 0 = 0$$

$$y' = 1 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$x' = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$x'=1$  の重みを考慮して第4番目が誤っていることがわかり、ここを逆転させて正常な符号がえられる。これは4番目が誤った符号 HE4 についての正常な訂正復号動作である。またこの場合にえられる(6), (7)で示される選別のための関数値を検討すると次のようになる。

$$\begin{aligned} \overline{HE}(M) &= (1 \oplus 0 + 1 \oplus 0)(0 \odot 0 + 0 \odot 0) \\ &\quad + (0 \oplus 1 + 0 \oplus 0 + 0 \oplus 1)(0 \odot 1 + 0 \odot 0 + 0 \odot 1) \\ &= (1+1)(1+1) + (1+0+1)(0+1+0) = 1 \\ HE(m) &= \overline{HE}(M) = 0 \end{aligned} \quad \left. \vphantom{\overline{HE}(M)} \right\} \quad (17)$$

また誤りのない HE4 については

$$\begin{aligned} \overline{HE}(M) &= (1+1)(0+0) + (1+1+1)(0+0+0) = 0 \\ HE(m) &= 1 \end{aligned} \quad \left. \vphantom{\overline{HE}(M)} \right\} \quad (18)$$

また検査出力関数を CHK とすると無誤りについては

$$CHK = x' + y' + z' = 0$$

これらをまとめると次のようになる。

$$\text{無誤りで訂正動作がない場合} \quad \overline{CHK} \cdot HE(m) = 1 \quad (19)$$

$$1 \text{ 誤りで正常訂正の場合} \quad CHK \cdot \overline{HE}(M) = 1 \quad (20)$$

次に受信符号 1001000 についての検査出力は

$$z' = 0 \oplus 0 \oplus 0 \oplus 1 = 1$$

$$y' = 0 \oplus 0 \oplus 0 \oplus 1 = 1$$

$$x' = 1 \oplus 0 \oplus 0 \oplus 1 = 0$$

これにより3番目を訂正する動作が生じる。この場合の選別出力  $\overline{HE}(M)$  と  $HE(m)$  は次のようになる。

$$\begin{aligned} \overline{HE}(M) &= (0+0)(1+1) + (1+1+0)(0+0+1) \\ &= 0+1 = 1; \quad HE(m) = 0 \end{aligned} \quad (21)$$

訂正された結査は 1001100 で表1の HE9 に1誤りが生じた場合の正常訂正動作である。しかし上記の受信符号 1001000 は符号 HEE の5番目と6番目が2重に誤ったものとする、この符号系の訂正能力を超えたものであり誤認による訂正ということになる。また(12)に示したものでは次のようになる。

\* 誤りの混入があると考えられるので送出側に対応した英小字を用いた。

$$\text{CHK} = x' + y' + z' = 1 + 0 + 0$$

これにより4番目を逆転して 1111000 がえられる。これは送出符号が  $\text{HEC} = 1100001$  の場合に1番目と5番目が逆方向に2重誤りの結果であるにも拘らず1誤りとしての動作が生じて  $\text{HEE} = 1111000$  と誤認される場合である。これらの論理的出力の組合せを訂正動作終了直後について検討すると

$$\text{正常な復号結果でない場合} \quad \text{CHK} \cdot \overline{\text{HE}}(M) = 1 \quad (22)$$

$$\text{正常な復号と誤認される場合} \quad \text{CHK} \cdot \text{HE}(m) = 1 \quad (23)$$

(19), (20), (22)および(23)は訂正動作についての判定のための論理項として起用できる。(20)と(22)とは同一形であるが、これは論理項のとりだしのタイミングで区別でき、図4に関係して後述する。

上記では偶数パリティ方式について記述したが、奇数パリティ方式については検査回路にあらかじめ検査ビットの初期値を1にしておけば検査回路は偶数パリティのものがそのまま起用できる。その検査ビットの変数を  $p$  とすると検査方程式は

$$\left. \begin{array}{l} (z'' =) \quad p \oplus z \oplus d \oplus c \oplus a = 0 \\ (y'' =) \quad p \oplus y \oplus d \oplus b \oplus a = 0 \\ (x'' =) \quad p \oplus x \oplus c \oplus b \oplus a = 0 \end{array} \right\} \quad (24)$$

である。例として 11111011 (=  $abcdyzp$ ) の場合の検査演算は

$$z'' = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$y'' = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0$$

$$x'' = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1$$

であり、5番目を逆転して 1101101 がえられる。これは表2のHDDの1誤りについての正常な復号動作になっている。すなわち検査ビット数は5になるが、初期値設定だけのちがいでレジスタ回路は偶数パリティの場合と共用できることになる。(図4については、初期値  $prs = 0$  を初期値  $prs = 1$  とするだけでよい。)

なお奇数パリティ方式の場合の選別のための論理設計も最大項図での図形演算で簡潔な結果がえられるが、その分解合成演算についても図2に示してある。

### 3. 巡回符号における訂正可能誤りと訂正不能誤り

前記ではハミング符号の原形についての誤りの訂正をブール代数的に論理関数としての取扱いで示したが、この種の符号については一般的な符号理論体系のもとに線形符号として統一的に取扱われるようになり、色々な符号系が論じられるようになった。そのひとつに巡回符号がある。これは符号生成の容易さと、バースト誤りに対する強みのあることからよく用いられるので、この誤り訂正機能についてもブール代数的に取扱い、その復号系についての図形演算による論理設計についても下記する。

巡回符号は、 $n$  個の符号からなる符号語を



$$(a_0, a_1, a_2, \dots, a_{n-1}) \quad (25)$$

としたとき、これを  $(n-1)$  次の多項式 (符号多項式)

$$F(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} \quad (26)$$

の係数に対応させ、より低い次数  $m$  の多項式  $G(x)$  で桁上げ無視の除算を実施して割り切れるものだけを取りあげて、その係数により符号語を作るものである。その場合の商  $Q(x)$  も低次数でえられ符号多項式は

$$F(x) = G(x)Q(x) \quad (27)$$

なる桁上げ無視の積の形で示すことができる。

(27)で  $G(x)$  が  $F(x)$  を割切ってしまうと

$$G(x) = g_0 + g_1x + g_2x^2 + \dots + g_mx^m; (m < n-1) \quad (28)$$

で示される時に、これは生成項式といわれる。

符号化しようとする情報は全体から  $m$  個を取除いた符号点に適当な配置で割当てることができるが、これも次のような情報多項式で示すことができる。<sup>3)</sup>

$$P(x) = p_0 + p_1x + p_2x^2 + \dots + p_kx^k; (k < n-1) \quad (29)$$

これを上記の  $Q(x)$  の多項式に対応させて符号語を生成する符号多項式がえられる。 $F(x)$  と  $G(x)$  とを上記のように設定すると符号の生成は簡単な巡回シフトレジスタで可能となるので巡回符号といはれる。

巡回シフトレジスタの起用には乗算器として起用される場合と除算器として起用される場合とがあるが、後者の方が符号化しようとする情報との対応がつく形でえられ、誤り訂正の復号回路についても線形符号のひとつとしてハミング符号と共通的に取扱いやすいので、ここでは除算形生成のものを取りあげることにする。

(26)で  $x=2$  として  $a_0 \sim a_{n-1}$  のすべてに2値変数を割当てることになると2進整数形になる。従って符号多項式の係数を取りあげて作った符号語は、論理変数で作られた論理関数としての取扱いができることになる。線形符号の基本的なものとしてまたハミング符号との対応で、ここでは長さ  $n=7$  のものを取りあげることが、7個の符号点に対応した論理変数として、I, J, K, と A, B, C および D を用いることにする。I, J および K は検査符号点、A, B, C および D は情報符号点に対応づけられるものであるが、ハミング符号では両者が互に入り込んだ姿で作られるのに対して、除算形巡回符号では符号多項式に対応して位置として高次側に元の情報語の配列を維持した形で、検査符号部はその残りの部分にまとめられた姿でえられるのが特長である。これはまた線形符号の統一理論のひとつとしての検査行列理論からのひとつの帰結でもある。

符号語のこのような姿での生成は、符号多項式の取扱いでは情報多項式  $P(x)$  の次数を  $x^{n-k}$  だけ高次にした  $x^{n-k}P(x)$  を  $G(x)$  による桁上げ無視の除算をしてえられる剰余に対応した多項式を作り、その係数を次数の降り順にして対応した検査符号部 K, J, I に割当てるということになる。

長さ7の符号化については  $x^7-1$  を割り切る多項式  $G(x)=x^3+x+1^*$  が起用される。これによる16進数値情報の基本的な符号化について下記する。ここでは(/)は桁上げ無視の除算記号で、情報点は情報語の1に対応した論理変数で示す。

16進数 (0001) 情報点 D ; 情報多項式  $x^3+0+0+0$

$$x^{n-k}P(x)=x^6 ; x^6(/)(x^3+x+1) \text{ の剰余}=1+x^2$$

従って符号語 CE1=1010001 (=IJKABCD)

16進数 (0010) 情報点 C ; 情報多項式  $x^2+0+0$

$$x^{n-k}P(x)=x^5 ; x^5(/)(x^3+x+1) \text{ の剰余}=1+x+x^2$$

従って符号語 CE2=1110010 (=IJKABCD)

同様にして 符号語 CE4=0110100 (=IJKABCD)

符号語 CE8=1101000 (=IJKABCD)

がえられる。上記の4個の基本符号語がえられると残りの符号語は、各符号点毎の縦方向の排他論理和演算として求められる。例えば符号語 CED は下記のようにえられる。

	I	J	K	A	B	C	D
CE1	1	0	1	0	0	0	1
	⊕	⊕	⊕	⊕	⊕	⊕	⊕
CE4	0	1	1	0	1	0	0
	⊕	⊕	⊕	⊕	⊕	⊕	⊕
CE8	1	1	0	1	0	0	0
CED	0	0	0	1	1	0	1

えられた符号語のすべては表2に示してある。

符号語は  $G(x)$  で割り切れるように作られており、割り切れなかった場合には何らかの誤りのある符号受信であるということになる。そのための検査には I, J, および K の値が関係するが、他の符号点とどのような論理関係にあるかを求めることで検査方程式を作ることができる。上記の基本的符号語で、I が値1となるような情報点は D, C, および A で J が値1となる情報点は C, B, および A であり、K が値1となる情報点は D, C, および B であることから検査方程式は次のようにえられる。(ここでも送出側に対応した英小文字を用いた。)

$$\left. \begin{array}{l} (i'=) \quad i \oplus a \oplus c \oplus d = 0 \\ (j'=) \quad j \oplus a \oplus b \oplus c = 0 \\ (k'=) \quad k \oplus b \oplus c \oplus d = 0 \end{array} \right\} \quad (30)$$

(30)に対応した検査を受信側で適合したタイミングで実施して、 $i'$ ,  $j'$  および  $k'$  のどれかに値1が生じておれば偶数パリティ検査にひつかかったことになる。このような値1についての重みは、 $i$  検査に4,  $j$  検査に2,  $k$  検査に1を与えてその代数和で示される番

\* これは2進整数としての 1111111 (/) 1011=1101 で割りきれることに対応している。

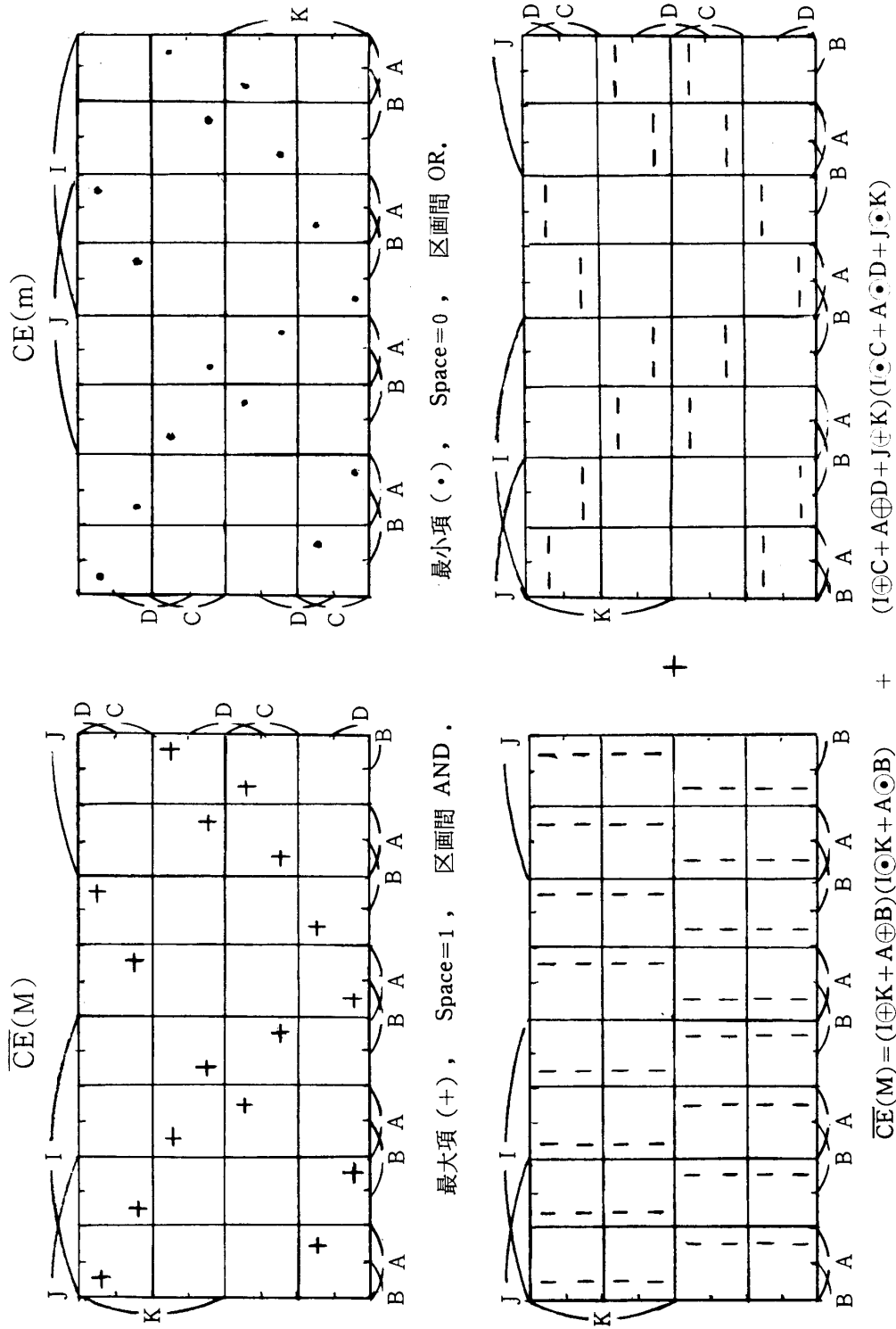


図3 偶数パリティ巡回符号の論理関数合成

符号名	I	J	K	A	B	C	D	等価数
CE0	0	0	0	0	0	0	0	00
CE1	1	0	1	0	0	0	1	81
CE2	1	1	1	0	0	1	0	114
CE3	0	1	0	0	0	1	1	35
CE4	0	1	1	0	1	0	0	52
CE5	1	1	0	0	1	0	1	101
CE6	1	0	0	0	1	1	0	70
CE7	0	0	1	0	1	1	1	23
CE8	1	1	0	1	0	0	0	104
CE9	0	1	1	1	0	0	1	57
CEA	0	0	1	1	0	1	0	26
CEB	1	0	0	1	0	1	1	75
CFC	1	0	1	1	1	0	0	92
CED	0	0	0	1	1	0	1	13
CEE	0	1	0	1	1	1	0	46
CEF	1	1	1	1	1	1	1	127

偶数パリティ巡回符号

表2. 巡回符号

号の符号点の誤りの指摘と対応ができることはハミング符号と同様であり、あらかじめ  $prs$  "1" を用意して奇数パリティ検査と兼用可能なことも同様である。

つまり誤り訂正についての選別回路の論理設計も前記同様に図形演算により簡潔な結果がえられ、それらについては図3にまとめて示してある。

#### 4. 訂正不可能符号の即時分離復号回路

上記では誤り訂正符号系の基本的な2種について訂正不能な誤りを含んだ受信情報を即時的に分離するについての論理的検討を行ったが、共通的に取扱いができることが多いので、偶数パリティ方式のハミング符号(7, 4)を代表的にえらんで論理回路構成を図4\*に示した。主要部分は、復号レジスタ(F)、分離レジスタ(S)、計数器(C1, C2, C3)および2個の分離論理回路(L-1, L-2)で構成されている。符号はZを先頭にして送出されるが、先行した1ビットタイム、後続した2ビットタイムの合計3個の空白パルスを含めて10ビットタイムの繰返して連続的に行はれるものとする。復号器では対応して  $t_0 \sim t_9$  で示される計10個のタイミング信号が用意されている。これらは単独の制御信号又は計数パルスとして使用されるほかに、 $t_1 \sim t_8$  の間にシフト動作を行うための SHIFT 信号を作るためにも用いられる。

動作開始は  $t_0$  のタイミングで生じるが、このパルスタイムは先行して動作していた  $t$

\* フリップフロップエのクロック入力省略してある。

形フリップフロップによる計数器 C1~C3 にプレセット値  $pqs$  "0" を導入すると同時に SHIFT 信号を起働させるタイミングである。このタイミングは次の  $t1$  のパルスタイムでは先頭の Z が両シフトレジスタの入力側に現出するように同期がとられているものとする。

SHIFT 信号で示される  $t1$ ~ $t8$  の間に受信符号情報はゲート I を通って後尾の空白 1 ビットが付加された 8 ビット語として F, S 両レジスタにとりいれられる。それと並行してパリティチェックのための適合したタイミングでゲートされた入力が計数器 C1~C3 にとりいれられる。C1 には  $z$  系列のチェック, C2 には  $y$  系列のチェック, C3 には  $x$  系列のチェックのために適合した計数入力がとりいれられることになる。 $t7$  で計数動作が終るが、この結果として 3 者共に 0 の値を示しておれば、いずれも偶数パリティチェックに合格したことを示しており、受信符号には誤りがなかったことを示すゲート番号 0 の出力だけが値 1 を示す。計数器のどれかに値 1 が生じておればゲート番号 1~7 のどれかひとつの出力が値 1 となるように計数器出力組合せ回路がつくられている。例えば 3 個共に値 1 の計数結果が生じておれば 7 番ゲート出力だけが値 1 を示すようになっており、7 番目の符号に誤りがあることを示し、この出力はその誤り訂正を行う信号として起用される。

復号レジスタの各段はそれぞれの次段えは正常な論理値を伝達することも、逆転した論理値を伝達することもできるようになっている。分離レジスタでは最後の S8 だけに正常な論理値または逆転した論理値がとりいれられるようになっている。計数動作が終了した次のタイミング  $t8$  では、両レジスタが 7 個の符号の F2~F8 えのとりいれを終了し、送出側で末尾に付加された空白の 1 ビットが両レジスタの入力側に現出している。復号レジスタでは各段毎に上記の番目 0~7 のゲート出力が次段えの伝達制御信号として導入されているので、このタイミングで必要な逆転論理値を次段に導入することにより、1 誤りの訂正された結果が F1~F7 にとりいれられる。

一方復号レジスタの各段の出力は L-1 回路と L-2 回路にみちびかれている。前者はタイミング  $t8$  で対応したレジスタ出力が所定のパターンになっているかどうかをチェックするものであり、後者は誤り訂正結果の出力パターンを  $t9$  のタイミングでチェックするためのものである。いずれも図 1 でえられた結果を論理回路化したものである。 $t8$  では F8~F2 には受信符号 ( $abcxdyz$ ) が収納されているので、これが選出された (ABCXDYZ) と同じで誤りがなければ回路出力  $\overline{HE}(M)=0$  であるが、ちがっておれば  $\overline{HE}(M)=1$  で、これは計数器出力の CHK と対応して生じる、L-1 回路の  $\overline{HE}(M)=1$  の出力は 8 番ゲートからタイミング  $t8$  で出力(8)としてとりだされ、これは S8 の入力を制御して逆転論理値をとりいれるようになっている、これは送出時に末尾に付加された空白ビットであるから逆転で生じた値 1 がとりいれられることになる。これは後程の印字動作では負符号として取扱われるためのものであり、分離レジスタでは訂正動作はないが負符号を付加する動作が  $t8$  で生じるのである。これに対して復号レジスタ内では  $t8$  では前記の訂正働

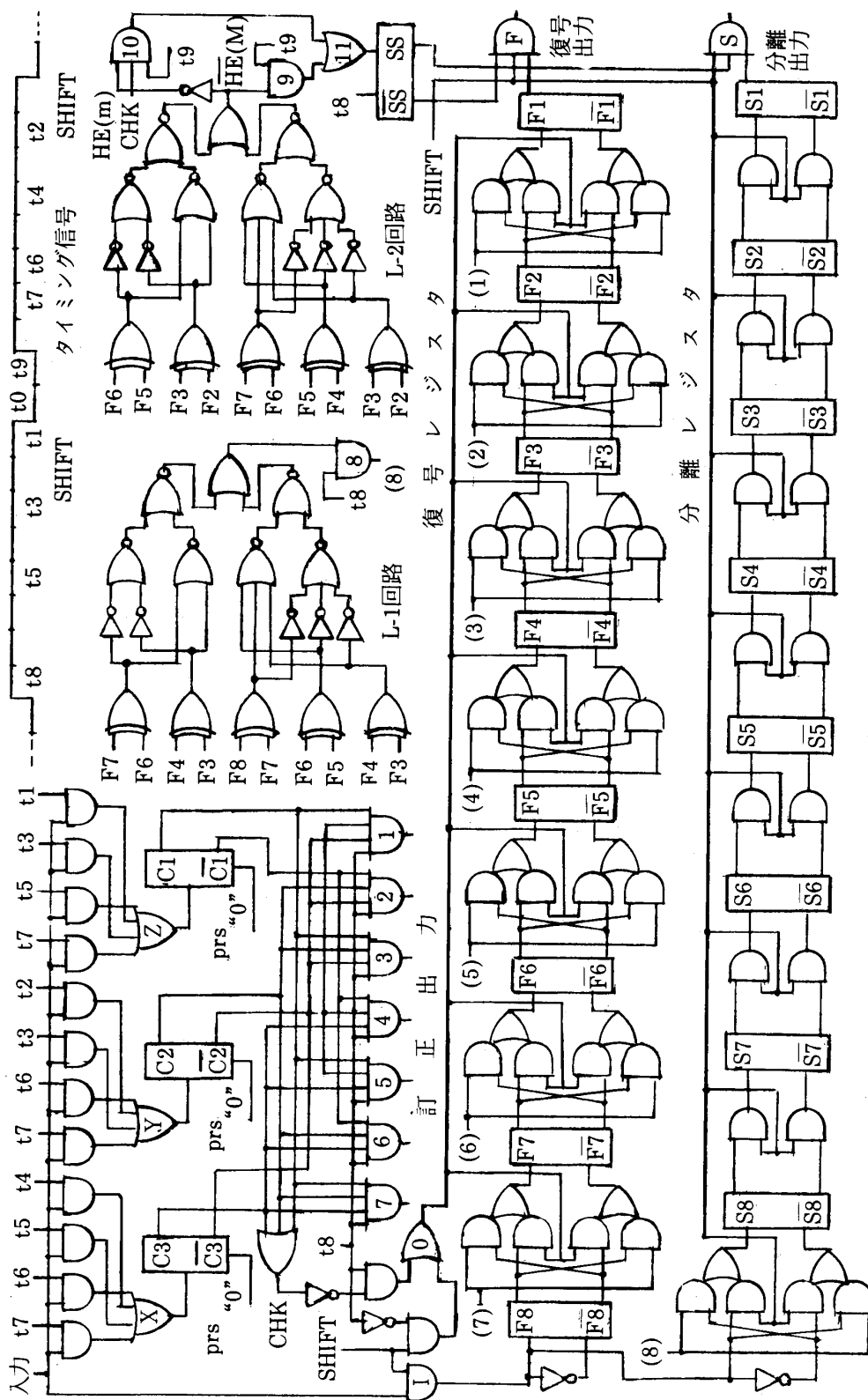


図4 訂正不能受信号の即時分離復号回路 (偶数パリティハイミング符号)

作が生じ、末尾の空白ビットはそのまゝとりいれられ、これは後程の印字には正符号としての“Space”が印字されるための1ビットとなり、合計8ビットとなったものが収納される。 $t_9$ ではL-2回路が復号レジスタのこのタイミングでの出力で動作するが、その回路出力 $\overline{HE}(M)=1$ は2誤り以上の誤りが生じて訂正結果が間違っただけであることを示し(22)式の出力を作ってゲート番号9からとりだされるようになっている。 $t_9$ で $\overline{HE}(M)=0$ 、従って $HE(m)=1$ の場合には、無誤りであった場合、正常な訂正が行われた場合のほかに、(23)式で示される誤認訂正が行われた場合が含まれる。これは $CHK=1$ となっていながら $HE(m)=1$ となっていることで特性づけられ、ゲート番号10より出力がとり出される。これら両者により $t_9$ では11番ゲート出力でフリップフロップSSがセットされるようになっているので、このセットされた出力により、訂正失敗の結果( $\overline{HE}(M)=1$ )、または誤認訂正の結果( $CHK \cdot HE(m)=1$ )が特性づけられることになる。

分離レジスタのS1の出力はセットされた $SS=1$ でSゲートよりとりだされるようになっており、次に生じるSHIFT信号により、訂正失敗となる前のデータまたは誤認訂正となる前のデータが負符号付きでシフト出力としてSゲートよりとりだされる。この時の信号レジスタ内の対応した情報は自動的にSHIFT信号により消滅する。SSがセットされないで $\overline{SS}=1$ の信号では復号レジスタのF1の出力がとりだせるようになっているので、これにより無誤りのものまたは正常な訂正が行われた結果はSHIFT信号によりFゲートからシフト出力として正符号付きでとりだされる。これらのSHIFT動作は後続の受信入力のとりいれ動作ともなっているため、両レジスタのどちらかが分離された結果として新入力データにより押しだされる形でとりだされるのである。またSSはタイミング $t_8$ で自動的にリセットされ、次のデータの処理に移行するようになっている。

このようにあるひとつの符号データが復号されるごとにその時点で正常なものとそうでないものとの分離が即時的に逐行されるのはフリップフロップSSのセット、リセットによるのである。このような即時処理ができない場合には、数千にもなる一連の符号データの受信終了後、改めての両処理として探表操作を含む作業で分離せざるを得ない。それについては何らかの形でソフトウェアに依存せざるを得ず、多大な処理時間が必要になる。そのような厄介さを回避できる点で即時処理は有効であろうと思はれる。

## あとがき

誤り訂正符号系の低訂正機能の基本的なものについて訂正不能データの即時分離を検討した。バースト誤りをも含めた高訂正機能のものについては別の観点からの検討が必要であろう。しかしその場合にも機能を超えた誤り訂正についての宿題が残る。符号系として低訂正機能であっても訂正不能データについての即時処理が可能であれば簡便法として再処理の厄介さを軽減できることでそれなりの意味があろう。即時処理をハードウェアで実現するについては図形演算による簡潔な論理設計が寄与しているところも多い。本文は双

対図による論理設計の応用例でもある。

検討に参加して頂いた各位に深謝する。

文 献

- 1) 豊田準三 双対図法による多変数論理設計 岡山大学紀要 第19号A pp. 47-66, 1984
- 2) 豊田準三 小西憲一 双対図法による論理関数の合成と無定義項 岡山大学紀要 第20号A pp. 163-173, 1985
- 3) 宮川 洋 著 情報理論 (コロナ社, 昭和59年3月, 頁199)

付 録 双対図による論理設計 要約

論理関数の図形表示のためにカルノ図は、 $2^n$ 個のセルに関数の最小項展開形を示す1を記入して関数の積和形の図形表示とし、記入のないところは否定関数の最小項展開形の図

	A															
	B							B								
	127	123	115	119	95	91	83	87	31	27	19	23	63	59	51	55
	126	122	114	118	94	90	82	86	30	26	18	22	62	58	50	54
C	124	120	112	116	92	88	80	84	28	24	16	20	60	56	48	52
	125	121	113	117	93	89	81	85	29	25	17	21	61	57	49	53
	111	107	99	103	79	75	67	71	15	11	03	07	47	43	35	39
	110	106	98	102	78	74	66	70	14	10	02	06	46	42	34	38
	108	104	96	100	76	72	64	68	12	08	00	04	44	40	32	36
	109	105	97	101	77	73	65	69	13	09	01	05	45	41	33	37
	E D							E D							E	
															G	
															F	

最大項番地, 区画間結合 AND, 使用記号 (+)

	A															
	B							B								
	00	04	12	08	32	36	44	40	96	100	108	104	64	68	76	72
	01	05	13	09	33	37	45	41	97	101	109	105	65	69	77	73
	03	07	15	11	35	39	47	43	99	103	111	107	67	71	79	75
	02	06	14	10	34	38	46	42	98	102	110	106	66	70	78	74
C	16	20	28	24	48	52	60	56	112	116	124	120	80	84	92	88
	17	21	29	25	49	53	61	57	113	117	125	121	81	85	93	89
	19	23	31	27	51	55	63	59	115	119	127	123	83	87	95	91
	18	22	30	26	50	54	62	58	114	118	126	122	82	86	94	90
	E D							E D							E	
															G	
															F	

最小項番地, 区画間結合 OR, 使用記号 (•)

付図 7変数の最小項番地と最大項番地



形表示としたものである。これが積和形の図形表示となるについてはセル相互間に論理和結合があることが前提となっている。これに対してセル相互間には論理積結合があるものとして、各セルを最大項に対応させ最大項展開形に存在している最大項を0で記入し、記入のないところは1とした上で0と最大項との間に論理和（従って0の代わりに+記号を使用する。）を作り、それにセル間の論理積結合特性を適用したものは関数の和積形の図形表示となる。この場合各セルの番地を両図形について双対的になるように領域の設定をすると両関数形がひとつの図上で表示できることになる。それぞれの図形の番地を他方の番地によみかえして図形をよみとると、それぞれの否定関数がよみとれる。

このような双対図によるとある論理関数の関数形の変換が容易になり、また複数個の関数の合成、あるいは複数個の成分関数えの分解が容易になる。

和積形関数だけを対象としたものを最大項図と称する。

## Online extracting of uncorrected data on automatic error correcting code systems

Junso TOYODA and Ken-ichi KONISHI

*Department of Electronic Science, Okayama University of Science  
Ridai-cho-1-1, Okayama 700, Japan*

### Abstract

In this paper it is discussed to extract uncorrected data, on real time processing, from received data of error correcting code systems. Applications are developed on Hamming Code System and Cyclic Code Systems. It is expected to avoid troublesome table look up processing, which, otherwise, needed to extract the data at the end of receiving. Dual Karnaugh Mapping is utilized on logic design of extracting circuits which seemed to lead the realization of the systems.