MULTIPLEX DIGITAL POR DIVISÃO EM CÓDIGOS CÍCLICOS

P O R

WILLIAM FERREIRA GIOZZA

TESE DE MESTRADO

Apresentada à Coordenação Setorial de Pós-Graduação e Pesquisa da Pró-Reitoria para Assuntos do Interior da Universidade Federal da Paraíba, em cumprimento às exigências para obtenção do Grau de Mestre em Ciências.

Campina Grande, Maio de 1979

G494m Giozza, William Ferreira.
Multiplex digital por divisão em códigos cíclicos / William Ferreira Giozza. - Campina Grande, 1979.
129 f.

Dissertação (Mestrado em Ciências) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 1979.
"Orientação : Prof. Dr. Ivan Rocha Neto".
Referências.

1. Sistema de Comunicação Digital. 2. Sistemas de Multiplexação Digital Inteligente. 3. Multiplex Digital. 4. Códigos Cíclicos - Divisão. 5. Dissertação - Ciências. I. Rocha Neto, Ivan. II. Universidade Federal da Paraíba - Campina Grande (PB). III. Título

CDU 621.391(043)

A G R A D E C I M E N T O S

A todos, que direta ou indiretamente, permitiram a minha dedicação neste trabalho e, em especial ao Dr. Ivan Rocha Neto pela verdadeira orientação proporcionada.

R E S U M O

A economia de linhas ou canais de transmissão em comunicações em geral, aliada à necessidade crescente de confiabilidade dos sistemas de comunicações de dados atuais, geram o desenvolvimento de sistemas de multiplexação digital inteligentes.

Este trabalho desenvolve e implementa em "hardware" um sistema de multiplexação digital por divisão em códigos cíclicos que incorpora e troca, de uma maneira adaptativa com a atividade dos canais na entrada do sistema, capacidade de correção de erros aleatórios por capacidade do canal multiplexado. O sistema incorpora capacidade de correção de erros aleatórios, adaptativa ao número de canais ativos na entrada, através de códigos cíclicos corretores de erros aleatórios com redundância variável.

A teoria básica para a descrição do sistema é apresentada assim como os circuitos projetados e a discussão dos resultados.

A B S T R A C T

Cost savings of transmission lines or channels in general communication systems with the growing need of reliabitlity in the present data communication systems have stimulated the development of inteligent digital multiplexing schemes.

The implementation of a cyclic codes division multiplexing system is described. This system exchanges adaptively the excess of channel capacity for random error-correcting ability. This feature is achieved by using random error-correcting cyclic codes with variable redundancy.

The basic theoretical description of the system, the circuits used and a discussion of the results are presented.

ÍNDICE

PÁGINA

PÁGINA

INDICE DE FIGURAS

FIGURAS — PÁGINA

FIGURAS PÁGINA

LISTA DE ABREVIAÇÕES

CCA - Codificador Cíclico Adaptativo

CCP - Contador de Controle de Palavra

CCQ - Contador de Controle de Quadro

CD - Compressor de Dados

CLC - Circuito Lógico Combinacional

CPS - Conversor Paralelo-Série

CRSQ - Contador de Recuperação de Sincronismo de Quadro

CSB - Canal Simétrico Binário

CSP - Conversor Série-Paralelo

CSQ - Contador de Sincronismo de Quadro

DCA - Decodificador Cíclico Adaptativo

DD - Descompressor de Dados

FA - "Full-Adder"

FF - "Flip-Flop"

FFT - "Flip-Flop" tipo T

GF(2) - Campo de Galois binário

LIFO - "Last in-first out"

MDC - Multiplexação por Divisão em Códigos

MDDCC - Multiplex Digital por Divisão em Códigos Cíclicos

MDF - Multiplexação por Divisão em Frequência

MDO - Multiplexação por Divisão Ortogonal

MDS - Multiplexação por Divisão em Sequência

MDT - Multiplexação por Divisão em Tempo

MDTA - Multiplexação por Divisão em Tempo Assíncrono

MDTS - Multiplexação por Divisão em Tempo Síncrono

MONO - Circuito Monoestável

PC - Processador Central

PLL - "Phase-Lock Loop"

RA - Registrador de Armazenamento

RD - Registrador de Deslocamento

RS - Registrador de Síndrome

TTL - "Transistor-Transistor Logic"

UCR - Unidade de Controle do Receptor

UCT - Unidade de Controle do Transmissor

Vcc - Tensão de alimentação dos circuitos TTL

CAPÍTULO I

I N T R O D U Ç Ã O

Em comunicações de dados, de uma maneira geral, o custo das linhas ou canais de transmissão são fatores determinantes no custo total do sistema de comunicações. Dessa forma, o desenvolvimento ou a utilização de sistemas que minimizem o número de canais de transmissão, assim como, aproveitem melhor os canais já existentes, constituem problemas de grande importância na engenharia de comunicações. Sistemas de multiplexação digital, baseados em transformações que permitem a transmissão de sinais digitais de diversas fontes de informação em um único sinal, se apresentam satisfatoriamente como uma das soluções a esses problemas.

Por outro lado, a limitação de capacidade de transmissão de informação dos canais reais de comunicações devido ao ruído e a banda passante limitada (faixa de frequências do canal), assim como, a exigência de transmissão de da

dos, através desses canais, de uma maneira confiável, geram a necessidade de uma proteção contra erros na recepção das mensagens transmitidas. Uma maneira de se conseguir tal proteção é através da codificação de linha (ou de canal) que consiste na introdução sistemática de redundâncias à informação transmitida permitindo a deteção e mesmo a correção de erros na recepção das mensagens.

Dentro dessa ótica apresentada acima e, levando em conta a natureza intermitente das comunicações em geral, sistemas de multiplexação digital, adaptativos a atividade dos canais multiplexados, foram desenvolvidos. Em particular, o aproveitamento da inatividade dos canais na entrada de um sistema de multiplexação para a incorporação de um aumento na capacidade de controle de erros automático, consolidou-se num sistema de multiplexação não-convencional desenvolvido por Gordon e Barrett (1971) e, posteriormente, adaptado por Rocha Neto (1975). Esse sistema de multiplexação digital desenvolvido apresenta duas limitações inerentes ao processo de multiplexação utilizado. Uma delas, é a limitação no número de canais possíveis de serem multiplexados e, a outra, o aproveitamento real da inatividade dos canais na incorporação de uma maior capacidade de controle de erros na recepção do sinal multiplexado. Rocha Neto (1975) propôs soluções quanto a superação do limite do número de canais possíveis de serem multiplexados pelo sistema desenvolvido por Gordon e Barrett. Uma das soluções propostas foi a da utilização de códigos cíclicos com redundância variável na codificação da informação multiplexada.

O presente trabalho de tese teve como objetivos o desenvolvimento e a implementação prática de um sistema de multiplexação digital não-convencional para canais síncronos que incorporasse capacidade de controle de erros no canal multiplexado, através de códigos cíclicos com redundancia variável, de uma maneira adaptativa ao número de canais efetivamente ativos na entrada do sistema. A utilização de códigos cíclicos, para se aumentar a confiabilidade da transmissão de dados através de canais ruidosos, deve-se, principalmente,aos recentes avanços da tecnologia de circuitos integrados que aliados a relativa facilidade de tratamento matemático dos códigos cíclicos, tornam-nos competitivos em termos de custo e complexidade.

O segundo capítulo introduz o problema de codificação de linha, através de códigos blocos lineares em geral, para a seguir apresentar os códigos cíclicos. No terceiro capítulo é feito um estudo comparativo entre os dois sistemas de multiplexação digital mais utilizados na prática (MDF e MDT) para em seguida introduzir-se o sistema de multiplexação digital não-convencional proposto neste trabalho de tese. O quarto capítulo apresenta o Multiplex Digital por Divisão em Códigos Cíclicos (MDDCC) descrevendo em detalhes a implementação do sistema. Finalmente, o quinto capítulo sumariza e discute as principais conclusões do trabalho e, sugere pontos interessantes para estudos posteriores. No apêndice , são apresentados os conceitos matemáticos básicos para a compreensão dos códigos cíclicos.

CAPÍTULO II

CÓDIGOS CÍCLICOS

2.1 - INTRODUÇÃO

O crescente uso de processadores de dados automáticos em redes de comunicações de dados exige sistemas de transmissão digitais cada vez mais confiáveis. Um dos problemas mais sérios que os projetistas desses novos sistemas de transmissão de dados a alta velocidade enfrentam, é a ocorrência de erros na transmissão. O problema de como controlar esses erros é de básica importância na engenharia de comunicações atual.

A introdução sistemática de símbolos redundantes às mensagens a serem transmitidas é chamada CODIFICAÇÃO DE LINHA e, pode ser usada para decidir ambiguidades na recepção dessas mensagens quando corrompidas pelo ruído no processo de transmissão. Assim, os códigos de linha são introdu-

zidos para se obter um controle sistemático dos erros produzidos pelo ruído, controle êsse que permite a deteção e mesmo a correção dos erros.

O potencial desse método de controle de erros foi estabelecido por Shannon em 1948 no "Teorema Fundamental da Teoria da Informação" (Abramson, 1963). O teorema afirma o seguinte para um canal com ruído:

- todo canal tem uma capacidade máxima de transmissão de informação definida por C* e, para qualquer taxa de transmissão de informação R** menor que C, existem códigos de taxa R que, com decodificação por máxima semelhança***, tem uma probabilidade de erros arbitrariamente pequena (Lin, 1970).

O teorema acima prova a existência de códigos que podem tornar a probabilidade de erros muito pequena na recepção da informação. Todavia, o teorema não é construtivo e, com o objetivo de superar esse problema desenvolveu-se a teoria da codificação.

---

* *A expressão que determina a capacidade de canal C com ruído gaussiano é* $C = B \log_2(1 + S/N)$ *onde B representa a largura de faixa do canal e S/N a relação potência sinal-ruído (Carlson, 1968).*

** *A taxa de transmissão de informação R de uma fonte de informação com entropia H e duração de símbolo* τ *é* $R = R/\tau$ *bits/segundo (Carlson, 1968).*

*** *Definida mais adiante neste Capítulo.*

A partir do trabalho de Shannon (Shannon, 1948), a teoria da codificação desenvolveu-se bastante e, continua sendo, hoje em dia, uma extensa área de pesquisa. Particularmente, pela inerente estrutura matemática assim como pelo número de trabalhos e obras publicadas, os códigos lineares foram os mais desenvolvidos e estudados. Dentro dessa classe de códigos, uma sub-classe, a dos códigos cíclicos,concentrou grande parte dos esforços desenvolvidos pelos pesquisadores sensíveis ao problema da codificação. Essa concentração se deve sintomaticamente às facilidades de tratamento com tais códigos advindas da estrutura algébrica inerente que permite encontrar vários métodos de decodificação simples e eficientes, assim como do crescente avanço na tecnologia de circuitos integrados.

Neste capítulo, alguns conceitos básicos de códigos bloco lineares são apresentados para a seguir introduzirmos os códigos cíclicos. Por conveniência de entendimento e da aplicação ao sistema proposto neste trabalho (Capítulo IV), a apresentação se restringe aos códigos cíclicos binários e não pretende ser exaustiva. Os conceitos matemáticos envolvidos na compreensão deste capítulo são apresentados resumidamente no Apêndice I.

2.2 - CÓDIGOS BLOCO LINEARES

2.2.1 - DEFINIÇÕES

No processo de codificação de uma sequência de dígitos de informação binários, divide-se a sequência em blocos de dígitos de informação e adiciona-se, de acordo com certas regras, dígitos redundantes a cada bloco. Se a redundância adicionada ao bloco de dígitos de informação verifica (i.é., diz respeito) apenas os dígitos referentes a esse bloco, o código é chamado CÓDIGO BLOCO. Por outro lado, códigos em que a redundância em um bloco verifica dígitos de informação em mais de um bloco são chamados códigos convolucionais (Elias, 1955). Os códigos bloco binários cujos dígitos redundantes são calculados com somadores módulo-2 (operação linear no GF(2)) são chamados CÓDIGOS BLOCO LINEARES BINÁRIOS. Aqueles cujos dígitos redundantes são calculados com lógica não-linear (portas lógicas AND, NAND, etc...) são chamados de não-lineares.

O tratamento dos códigos bloco lineares usa o conceito de espaço vetorial (ver Apêndice I) sendo assim termos como vetor-código (ou, simplesmente vetor), espaço vetorial, etc., serão utilizados frequentemente neste capítulo em substituição aos termos palavra-código, alfabeto-código , etc... Esses códigos são normalmente representados por pares ordenados (n, k) onde n representa o número de dígitos em ca-

da palavra-código e é chamado de COMPRIMENTO DO CÓDIGO, k representa o número de dígitos de informação por bloco e (n - k) é o número de dígitos redundantes adicionados ao bloco de k dígitos de informação. No caso binário a EFICIÊNCIA DO CÓDIGO é definida pela relação k/n.

Um código bloco linear binário pode ser definido como sendo um conjunto de 2^k n-uplas que formam um sub-espaço do espaço vetorial de todas as n-uplas.

2.2.2 - MATRIZ GERADORA

Um código binário (n, k) possue 2^k palavras-código distintas. Para se usar esse código é necessário que o transmissor "conheça" todas as palavras-código e, esteja pronto para enviá-las de acordo com a informação a ser transmitida. No caso de n e k consideráveis torna-se proibitivo o armazenamento dos n x 2^k dígitos binários no transmissor. Todavia, quando as 2^k n-uplas formam um sub-espaço do espaço de todas as n-uplas (i.é., um código linear) é possível se obter um conjunto de k vetores linearmente independentes que, através de combinações lineares, geram todos os elementos do sub-espaço. Por exemplo, se

$$[V_1], [V_2], \ldots, [V_k]$$

são k n-uplas linearmente independentes no sub-espaço então qualquer outra n-upla $[U]$ nesse sub-espaço pode ser obtida na forma

$$[U] = m_1 [V_1] + m_2 [V_2] + \ldots + m_k [V_k]$$

onde $m_i \in \{0, 1\}$ e $1 \leq i \leq k$.

Então, tem-se que um código linear com 2^k vetores-código pode ser descrito por um conjunto de k vetores -código linearmente independentes. Esse resultado é melhor descrito em termos de uma matriz k x n onde as linhas são os k vetores linearmente independentes.

$$[G] = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_k \end{bmatrix}$$

A matriz $[G]$ é chamada matriz geradora do código. Qualquer palavra-código pode ser gerada como se segue. Seja $[M] = [m_1, m_2, \ldots, m_k]$ a sequência de dígitos de informação. A palavra-código correspondente ao bloco de k dígitos de informação $[M]$ resulta da matriz produto $[M].[G]$. Assim,

$$[U] = [M].[G] = m_1[V_1] + m_2 [V_2] + \ldots + m_k [V_k]$$

é a palavra-código associada com o bloco de k dígitos de informação $[M]$. O vetor $[U]$ é uma combinação linear das linhas da matriz geradora do código $[G]$. Dessa forma, desde

que um código bloco linear é completamente especificado por sua matriz geradora $[G]$, o problema de armazenamento no transmissor é reduzido significativamente. O codificador para um código bloco linear consiste basicamente em elementos de armazenamento das linhas da matriz $[G]$ e um circuito lógico para efetuar a combinação linear dessas linhas de acordo com a sequência de dígitos de informação. Como $[G]$ é uma matriz não-singular é possível escrever $[G] = [g : I_k]$ (Peterson, 1972) onde I_k é uma matriz unitária k x k e g é uma matriz k x (n-k). Nessa situação, as palavras-código geradas tem as k últimas posições ocupadas pelos dígitos de informação enquanto que os primeiros (n-k) dígitos são combinações lineares dos dígitos de informação. Um código com essa estrutura é chamado de SISTEMÁTICO. Os (n-k) dígitos da palavra-código são chamados de DÍGITOS DE PARIDADE e as funções lineares que geram os dígitos de paridade são chamados de EQUAÇÕES DE PARIDADE.

2.2.3 - MATRIZ DE PARIDADE

Dada uma matriz kxn $[G]$ de um código linear é possível se obter a matriz (n-k) x n $[H]$ de tal forma que o espaço vetorial gerado pelas linhas da matriz $[G]$ seja ortogonal a matriz $[H]$, i.é., se $[V_i]$ é um vetor no espaço vetorial gerado pelas linhas da matriz $[G]$ então $[V_i] \cdot [H]^T = 0$. A matriz $[H]$ é chamada matriz paridade do código e pode ser representada como $[H] = [I_{n-k} : h]$

onde h é uma matriz (n-k) x k e I_{n-k} é a matriz unitária (n-k) x (n-k). Pode ser mostrado que $[h] = [g]^T$ (Peterson, 1972) onde $[g]^T$ é a matriz transposta de $[g]$. Desde que as linhas da matriz $[H]$ são linearmente independentes, elas geram um código linear (n, n-k) que é conhecido como o dual do código linear (n,k) gerado por $[G]$.

2.2.4 - SÍNDROME

Supondo que uma palavra-código $[V]$ de um código bloco linear com matriz geradora $[G]$ e matriz paridade $[H]$ é transmitida em um canal ruidoso. No receptor é recebida uma n-upla $[R] = [r_0, r_1, \ldots, r_{n-1}]$ que pode ser diferente da n-upla $[V]$ transmitida devido ao ruído adicionado no canal durante a transmissão. É tarefa do decodificador recuperar $[V]$ de $[R]$. O primeiro passo consiste em verificar se $[R]$ é uma palavra-código. Esse passo pode ser representado pela equação

$$[R] \cdot [H]^T = [S]$$

onde $[S]$ é uma (n-k)-tupla chamada SÍNDROME de $[R]$. Se $[S] = [0]$ assume-se que não ocorreram erros e que $[R] = [V]$. Se $[S] \neq [0]$, então $[R]$ não é um vetor do espaço gerado pelas linhas da matriz $[G]$ e o decodificador usa essa síndrome para detetar e/ou corrigir erros. A n-upla recebida $[R]$ pode ser escrita como $[R] = [V] + [E]$ onde

$[E] = [e_0, e_1, \ldots, e_{n-1}]$ é uma n-upla que representa o padrão de erros introduzidos pelo canal durante a transmissão.

2.2.5 - DISTÂNCIA DE HAMMING

O número de componentes diferentes de zero em uma n-upla $[V]$ é chamado de PESO DE HAMMING de $[V]$ e é representado por $W_{(v)}$.

O número de posições em que duas n-uplas $[V_1]$ e $[V_2]$ diferem é chamado de DISTÂNCIA DE HAMMING entre $[V_1]$ e $[V_2]$ e é representado por $d_{(V_1V_2)}$.

A menor distância dentre qualquer par de palavras-código é chamada de DISTÂNCIA MÍNIMA do código e é representada simplesmente por d.

A soma de duas palavras-código de um código linear dá como resultado uma outra palavra-código. Esse fato se deve às propriedades de grupo (Ver Apêndice I) dos espaços vetoriais e pode ser representado por

$$[V_1] + [V_2] = [V_1 + V_2] = [V_3]$$

Daí tem-se que $W_{(V_1,V_2)} = W_{(V_3)}$ ou que $d_{(V_1,V_2)} = W_{(V_3)}$. Portanto, para códigos lineares a distância mínima é igual ao peso da palavra-código diferente de zero de peso mínimo. No caso linear existe uma propriedade importante que relaciona d com a matriz paridade $[H]$. A distân-

cia mínima de um código linear pode ser expressa em termos da matriz [H] como se segue.

Teorema 2.1 Um código linear cuja matriz de paridade [H] possue (d-1) colunas linearmente independentes tem uma distância mínima maior ou igual a d (Peterson, 1972).

2.2.6 - CANAL SIMÉTRICO BINÁRIO E DECODIFICAÇÃO POR MÁXIMA SEMELHANÇA

O conhecimento do comportamento estatístico do canal é de vital importância para o bom desempenho do codificador de linha. Na prática, essas estatísticas tornam-se de difícil obtenção e um modelo teórico é então usado. Um dos modelos de canais mais usados é o CANAL SIMÉTRICO BINÁRIO (CSB) onde se assume que os erros ocorrem independentemente (i.é., erros aleatórios) e que 0's e 1's tem a mesma probabilidade de estarem errados. Na Figura 2.1 representa-se esquematicamente o CSB.

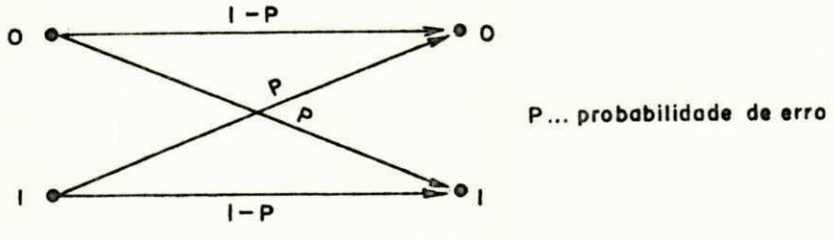


Fig. 2.1 - Canal Simétrico Binário

Se as palavras-código de um código (n,k) são selecionadas independentemente e todas tem a mesma probabilidade de serem transmitidas pelo canal, então um modo de decodificá-las pode ser o que se segue. Na recepção o decodificador compara a n-upla recebida $[R]$ com as 2^k n-uplas que compõem o código. Seleciona a n-upla mais próxima de $[R]$ em termos da distância de Hamming, i.é., a palavra-código que difere de $[R]$ num número mínimo de posições. Essa palavra é assumida na recepção como a palavra que foi transmitida. Esse procedimento de decodificação é chamado de DECODIFICAÇÃO POR MÁXIMA SEMELHANÇA.

Num CSB com decodificação por máxima semelhança tem-se que para um código com distância mínima d ser capaz de corrigir todos padrões de t ou menos erros aleatórios por palavra-código a seguinte desigualdade deve acontecer

$$d \geq 2t + 1 \qquad \text{(Peterson, 1972).}$$

Em outras palavras, a capacidade de correção de t erros aleatórios por palavra-código de um código linear é dada por

$$t = \left\lceil \frac{d-1}{2} \right\rceil \qquad \text{(Lin, 1970)}$$

onde $\left\lceil \frac{d-1}{2} \right\rceil$ significa o maior inteiro menor ou igual a $\frac{d-1}{2}$. A capacidade de deteção de D erros por palavra-código é dada por

$$D = d - 1 \qquad \text{(Lin, 1970)}$$

2.3 - CÓDIGOS CÍCLICOS

2.3.1 - DEFINIÇÕES

Um código bloco linear é chamado CÓDIGO CÍCLICO se o resultado de qualquer permutação cíclica de qualquer de suas palavras-código é uma outra palavra-código. Segue-se que se $[V] = [v_0, v_1, \ldots, v_{n-1}]$ é uma palavra-código de um código cíclico então o vetor $[V^{(i)}] = [v_{n-i}, v_{n-i+1}, \ldots, v_{n-1}, v_0, v_1, \ldots, v_{n-i-1}]$ obtido pelo deslocamento cíclico de i posições à direita do vetor $[V]$, é também uma palavra-código. Uma n-upla como $[V]$ acima pode ser representada na forma de um polinomio V(X) de grau no máximo igual a (n-1).

$$[V] = [v_0, v_1, \ldots, v_{n-1}] \leftrightarrow V(X) = v_0 + v_1 X + \ldots + v_{n-1} X^{n-1}$$

Assim a cada vetor corresponde um polinomio de grau (n-1) ou menor. Se $v_{n-1} = 1$, o grau de V(X) é (n-1); se $v_{n-1} = 0$ o grau é menor que (n-1). O polinomio V(X) é chamado polinomio código de $[V]$. O polinômio código correspondente ao vetor $[V^{(i)}]$ é

$$V^{(i)}(X) = v_{n-i} + v_{n-i+1} X + \ldots + v_{n-1} X^{i-1} + v_0 X^i + \ldots + v_{n-i-1} X^{n-1}$$

Pode ser mostrado (Lin, 1970) que $V^{(i)}(X)$ é o resto da divisão de $X^i V(X)$ por $X^n + 1$, i.é,

$$X^i V(X) = q(X)\ (X^n + 1) + V^{(i)}(X)$$

Teorema 2.2 Em um código cíclico binário (n,k) existe um e só um polinomio código G(X) de grau n-k

$$G(X) = 1 + g_1 X + g_2 X^2 + \ldots + g_{n-k-1} X^{n-k-1} + X^{n-k}$$

onde todo polinomio código é um múltiplo de G(X) e todo polinomio de grau (n-1) ou menor que é um múltiplo de G(X) deve ser um polinomio código (Lin, 1970).

Mostra-se (Lin, 1970) que um código cíclico (n,k) é especificado completamente pelo polinômio G(X) que é chamado de POLINÔMIO GERADOR do código. Todo polinômio código pode ser expresso na forma

$$V(X) = M(X) \cdot G(X)$$

onde $M(X) = m_0 + m_1 x + \ldots + m_{k-1} X^{k-1}$ é um polinômio cujos coeficientes são os k dígitos de informação a serem codificados.

Teorema 2.3 O polinômio gerador G(X) de um código cíclico binário (n,k) é um fator de $X^n + 1$, i.é,

$$X^n + 1 = H(X) \cdot G(X) \qquad \text{(Lin, 1970)}$$

Teorema 2.4 Se G(X) é um polinômio de grau (n-k) e é um fator de $X^n + 1$ então G(X) gera um código cíclico binário (n,k) (Lin, 1970).

Para exemplificar as teses apresentadas nos teoremas acima temos que o polinômio $X^{15} + 1$ pode ser fatorado da seguinte maneira

$$X^{15} + 1 = (X+1)\,(X^2+X+1)\,(X^4+X+1)\,(X^4+X^3+1)\,(X^4+X^3+X^2+X+1)$$

Alguns códigos de comprimento 15 gerados a partir da decomposição de $X^{15} + 1$ são listados a seguir

CÓDIGO	POLINÔMIO GERADOR
(15,11)	$G_5(X) = X^4+X+1$
(15,7)	$G_4(X) = (X^4+X+1)\,(X^4+X^3+X^2+X+1) = X^8+X^7+X^6+X^4+1$
(15,5)	$G_3(X) = (X^4+X+1)\,(X^4+X^3+X^2+X+1)\,(X^2+X+1) = X^{10}+X^8+X^5+X^4+X^2+X+1$
(15,2)	$G_2(X) = (X^4+X+1)\,(X^4+X^3+X^2+X+1)\,(X^4+X^3+1)\,(X+1) = X^{13}+X^{12}+X^{10}+X^9+X^7+X^6+X^4+X^3+X+1$
(15,1)	$G_1(X) = (X^4+X+1)\,(X^4+X^3+X^2+X+1)\,(X^4+X^3+1)\,(X^2+X+1) = X^{14}+X^{13}+X^{12}+X^{11}+X^{10}+X^9+X^8+X^7+X^6+X^5+X^4+X^3+X^2+X+1$

Uma definição equivalente para códigos cíclicos derivada de suas propriedades matemáticas é a que se segue.

Códigos cíclicos binários são ideais (ver

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 Tel (083) 324-7222-R 355
58.100 - Campina Grande - Paraíba

Apêndice I) na álgebra de polinômios módulo $X^n + 1$. Assim, as propriedades dos códigos cíclicos podem ser obtidas do estudo de ideais. Um resultado importante é que códigos cíclicos binários podem ser completamente especificados em termos das raízes do polinômio gerador G(X) em um campo extensão do GF(2). A fatorização de $X^n + 1$ dá como resultado

$$X^n + 1 = (X + \alpha_1)\ (X + \alpha_2)\ \ldots\ (X + \alpha_n)$$

onde as raízes $\alpha_1, \alpha_2, \ldots, \alpha_n$ são elementos do campo GF (n+1) extensão de GF(2) e n, no caso binário, é uma potência de 2 menos 1, i.é, $n = 2^m - 1$ (Peterson, 1972).

Um elemento α de um campo GF(q) é chamado elemento primitivo se todos os elementos diferentes de zero desse campo puderem ser expressos como potências de α. Assim, cada uma das raízes $\alpha_1, \alpha_2, \ldots, \alpha_n$ pode ser expressa como uma potência de uma raiz primitiva α, i.é., $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{n-1}, \alpha^n = \alpha^0 = 1$.

Um polinômio irredutível de grau m no GF(2) é chamado de primitivo se tiver um elemento primitivo do $GF(2^m)$ como raiz.

Um exemplo esclarece os conceitos apresentados. Seja $X^4 + X + 1$ um polinômio primitivo com coeficientes no GF(2). Seja α um elemento primitivo do $GF(2^4)$. Então

$$\alpha^4 + \alpha + 1 = 0$$

e os 2^4 elementos do $GF(2^4)$ podem ser formados pelo campo dos polinômios no GF(2) módulo X^4+X+1 como se segue

$$\alpha^0 = 1 = 0\ 0\ 0\ 1$$
$$\alpha^1 = \alpha = 0\ 0\ 1\ 0$$
$$\alpha^2 = \alpha^2 = 0\ 1\ 0\ 0$$
$$\alpha^3 = \alpha^3 = 1\ 0\ 0\ 0$$
$$\alpha^4 = \alpha + 1 = 0\ 0\ 1\ 1$$
$$\alpha^5 = \alpha\,\alpha^4 = \alpha^2 + \alpha = 0\ 1\ 1\ 0$$
$$\alpha^6 = \alpha^2\alpha^4 = \alpha^3 + \alpha^2 = 1\ 1\ 0\ 0$$
$$\alpha^7 = \alpha^3\alpha^4 = \alpha^3 + \alpha + 1 = 1\ 0\ 1\ 1$$
$$\alpha^8 = \alpha^4\alpha^4 = \alpha^2 + 1 = 0\ 1\ 0\ 1$$
$$\alpha^9 = \alpha\alpha^8 = \alpha^3 + \alpha = 1\ 0\ 1\ 0$$
$$\alpha^{10} = \alpha\alpha^9 = \alpha^2 + \alpha + 1 = 0\ 1\ 1\ 1$$
$$\alpha^{11} = \alpha\alpha^{10} = \alpha^3 + \alpha^2 + \alpha = 1\ 1\ 1\ 0$$
$$\alpha^{12} = \alpha\alpha^{11} = \alpha^3 + \alpha^2 + \alpha + 1 = 1\ 1\ 1\ 1$$
$$\alpha^{13} = \alpha\alpha^{12} = \alpha^3 + \alpha^2 + 1 = 1\ 1\ 0\ 1$$
$$\alpha^{14} = \alpha\alpha^{13} = \alpha^3 + 1 = 1\ 0\ 0\ 1$$
$$\alpha^{15} = \alpha\alpha^{14} = 1 = 0\ 0\ 0\ 1$$

O polinomio de menor grau com coeficientes no GF(2) que é um fator de $X^n + 1$ e contém α_i como raiz é chamado polinômio mínimo de α_i e denota-se por $m_{\alpha_i}(X)$.

Se $n = 2^m - 1$ pode ser mostrado que o máximo grau de um polinômio mínimo $m_{\alpha_i}(X)$ é m (Peterson, 1972).No caso binário $m^2_{\alpha_i}(X) = m_{\alpha_i}(X^2)$, então se β é uma raiz de

$m_{\alpha_i}(X)$, β^2, β^4, β^8, ..., também o são. Se G(X), um fator de $X^n + 1$, possue raízes distintas α_1, α_2, ..., α_{n-k} então qualquer polinômio V(X) pertence ao código gerado por G(X) se e só se $V(\alpha_i) = 0$ para $1 \leq i \leq n-k$. No caso binário, a condição de $X^n + 1$ só possuir raizes distintas é de que n seja ímpar.

Teorema 2.5 Se β é uma raiz de um polinômio V(X) então V(X) é divisível por $m_\beta(X)$ o polinômio mínimo de β (Peterson, 1972).

Como consequência do Teorema 2.5, se o polinômio mínimo de α_i é $m_{\alpha_i}(X)$ então V(X) é uma palavra - código em um código cíclico se e só se, V(X) é divisível por $m_{\alpha_1}(X)$, $m_{\alpha_2}(X)$, ..., $m_{\alpha_{n-k}}(X)$, i.é., V(X) deve dividir o mínimo múltiplo comum de $m_{\alpha_1}(X)$, $m_{\alpha_2}(X)$, ..., $m_{\alpha_{n-k}}(X)$. Assim, o polinômio gerador do código G(X) pode ser escrito como

$$G(X) = \text{MMC}\{m_{\alpha_1}(X), m_{\alpha_2}(X), \ldots, m_{\alpha_{n-k}}(X)\}$$

onde α_i, $(1 \leq i \leq n-k)$, é raiz de G(X).

Os códigos apresentados num exemplo anterior podem ser expressos em termos das raizes do polinômio gerador (Peterson, 1972).

CÓDIGO	RAÍZES DO POLINÔMIO GERADOR	d	t
(15,11)	α	3	1
(15,7)	α, α^3	5	2
(15,5)	α, α^3, α^5	7	3
(15,2)	1, α, α^3, α^7	10	4
(15,1)	α, α^3, α^5, α^7	15	7

Os polinômios geradores são obtidos como se segue

a - Código (15,11)

$G_5(X) = MMC \{m_\alpha(X)\}$

$m_\alpha(X) = X^4 + X + 1$

$G_5(X) = X^4 + X + 1$

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 - Tel (083) 321-7222-R 355
58.100 - Campina Grande - Paraíba

b - Código (15,7)

$G_4(X) = MMC \{m_\alpha(X), m_{\alpha 3}(X)\}$

$m_\alpha(X) = X^4+X+1$

$m_{\alpha 3}(X) = (X+\beta)(X+\beta^2)(X+\beta^4)(X+\beta^8)$, $\beta = \alpha^3$

$= (X+\alpha^3)(X+\alpha^6)(X+\alpha^{12})(X+\alpha^9)$

$= (X+\alpha^3)(X+\alpha^3+\alpha^2)(X+\alpha^3+\alpha^2+\alpha+1)(X+\alpha^3+\alpha)$

$= X^4+X^3+X^2+X+1$

$G_4(X) = (X^4+X+1)(X^4+X^3+X^2+X+1)$

$= X^8+X^7+X^6+X^4+1$

c - Código (15,5)

$$G_3(X) = MMC\{m_\alpha(X),\ m_{\alpha^3}(X),\ m_{\alpha^5}(X)\}$$

$$m_\alpha(X) = X^4 + X + 1$$

$$m_{\alpha^3}(X) = X^4 + X^3 + X^2 + X + 1$$

$$m_{\alpha^5}(X) = X^2 + X + 1$$

$$G_3(X) = X^{10} + X^8 + X^5 + X^4 + X^2 + X + 1$$

d - Código (15,2)

$$G_2(X) = MMC\{m_1(X),\ m_\alpha(X),\ m_{\alpha^3}(X),\ m_{\alpha^7}(X)\}$$

$$m_1(X) = X + 1$$

$$m_\alpha(X) = X^4 + X + 1$$

$$m_{\alpha^3}(X) = X^4 + X^3 + X^2 + X + 1$$

$$m_{\alpha^7}(X) = X^4 + X^3 + 1$$

$$G_2(X) = X^{13} + X^{12} + X^{10} + X^9 + X^7 + X^6 + X^4 + X^3 + X + 1$$

c - Código (15,1)

$$G_1(X) = MMC\{m(X),\ m_{\alpha^3}(X),\ m_{\alpha^5}(X),\ m_{\alpha^7}(X)\}$$

$$G_1(X) = X^{14} + X^{13} + X^{12} + X^{11} + X^{10} + X^9 + X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X + 1$$

2.3.2 - REPRESENTAÇÃO MATRICIAL

Seja G(X) o polinômio gerador de um código cíclico binário (n,k). Seja $M(X) = m_o + m_1 k + \ldots m_{k-1} X^{k-1}$ o

polinômio correspondente a k-tupla de dígitos de informação $(m_0, m_1, \ldots, m_{k-1})$. Multiplicando-se M(X) por X^{n-k} e dividindo-se o resultado por G(X) obtêm-se.

$$X^{n-k} M(X) = Q(X) G(X) + C(X)$$

onde Q(X) é o cociente da divisão e $C(X) = c_0 + c_1 X + \ldots + c_{n-k-1} X^{n-k-1}$ é o resto da divisão. Esse resultado pode ser escrito assim,

$$C(X) + X^{n-k} M(X) = Q(X) G(X)$$

donde

$$C(X) + X^{n-k} M(X) = c_0 + c_1 X + \ldots + c_{n-k-1} X^{n-k-1} + m_0 X^{n-k} + \ldots + m_{k-1} X^{n-1}$$

é um polinômio código do código cíclico gerado por G(X) e corresponde à palavra-código $(c_0, c_1, \ldots, c_{n-k-1}, m_0, m_1, \ldots, m_{k-1})$. Dessa forma, a palavra-código consiste do bloco de k dígitos de informação seguido dos (n-k) dígitos de paridade e o código é dito estar numa forma sistemática.

A matriz geradora do código cíclico (n,k) pode ser formada numa forma sistemática da seguinte maneira. Os polinômios $V_i(X) = X^{n-k+i} + C_i(X)$, (i=0,1,2,...,k-1), são polinômios códigos linearmentes independentes que formam uma base do espaço-código. Assim, arranjando-se os k polinômios códigos como as k linhas de uma matriz k x n obtem-se

$$[G] = \begin{bmatrix} V_0 \\ V_1 \\ . \\ . \\ . \\ V_{k-1} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} & \cdots & c_{0,n-k-1} & 1\ 0\ \ldots\ 0 \\ c_{10} & c_{11} & \cdots & c_{1,n-k-1} & 0\ 1\ \ldots\ 0 \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ c_{k-1,0} & c_{k-1,1} & \cdots & c_{k-1,n-k-1} & 0\ 0\ \ldots\ 1 \end{bmatrix}$$

que é a matriz geradora do código cíclico. Se $(m_0, m_1, \ldots, m_{k-1})$ são os k dígitos de informação a serem codificados então o vetor código correspondente é

$$[V] = (m_0, m_1, \ldots, m_{k-1})\ [G]$$
$$= m_0[V_0] + m_1[V_1] + \ldots + m_{k-1}[V_{k-1}]$$

A matriz paridade do código é

$$[H] = \begin{bmatrix} 1\ 0\ \ldots\ 0 & c_{00} & c_{10} & \cdots & c_{k-1,0} \\ 0\ 1\ \ldots\ 0 & c_{01} & c_{11} & \cdots & c_{k-1,1} \\ . & & & & \\ . & & & & \\ . & & & & \\ 0\ 0\ \ldots\ 1 & c_{0,n-k-1} & c_{1,n-k-1} & \cdots & c_{k-1,n-k-1} \end{bmatrix}$$

e pode ser escrita e sistematizada, em termos das raizes α_1, α_2, ..., α_{n-k} do polinômio gerador

$$[H] = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{n-1} \\ \vdots & & & & \\ 1 & \alpha_{n-k} & \alpha_{n-k}^2 & \cdots & \alpha_{n-k}^{n-1} \end{bmatrix}$$

A obtenção das matrizes geradora e de paridade dos códigos apresentados anteriormente exemplificam o procedimento descrito acima.

Código (15,11)

$G_5(X) = X^4 + X + 1$

$X^{4+i} = Q_i(X)\, G_5(X) + C_i(X) \qquad i = 0,1,2, \ldots, 10$

$C_0(X) = 1 + X$

$C_1(X) = X + X^2$

$C_2(X) = X^2 + X^3$

$C_3(X) = 1 + X + X^3$

$C_4(X) = 1 + X^2$

$C_5(X) = X + X^3$

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 Tel (083) 321 7222-R 355
58.100 - Campina Grande - Paraíba

$$C_6(X) = 1 + X + X^2$$

$$C_7(X) = X + X^2 + X^3$$

$$C_8(X) = 1 + X + X^2 + X^3$$

$$C_9(X) = 1 + X^2 + X^3$$

$$C_{10}(X) = 1 + X^3$$

$$[G_5] = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[H_5] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Código (15,7)

$$G_4(X) = X^8 + X^7 + X^6 + X^4 + 1$$

$$X^{8+i} = Q_i(X)\, G_4(X) + C_i(X) \qquad i = 0,1,2,\ldots,6$$

$$[G_4] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[H_4] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Código (15,5)

$$G_3(X) = X^{10} + X^8 + X^5 + X^4 + X^2 + X + 1$$

$$X^{10+i} = Q_i(X)\, G_3(X) + C_i(X) \qquad i = 0,1,2,3,4.$$

$$[G_3] = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[H_3] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Código (15,2)

$$G_2(X) = X^{13} + X^{12} + X^{10} + X^9 + X^7 + X^6 + X^4 + X^3 + X + 1$$

$$X^{13+i} = Q_i(X)\ G_2(X) + C_i(X) \qquad i = 0,1$$

$$[G_2] = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$[H_2] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Código (15,1)

$$G_1(X) = X^{14} + X^{13} + X^{12} + X^{11} + X^{10} + X^9 + X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X + 1$$

$$[G_1] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$[H_1] = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}$$

2.3.3 - CODIFICAÇÃO COM (n-k) ESTÁGIOS

A propriedade cíclica permite uma implementação sequencial da matriz [G] através de registradores de deslocamento ("shift registers"). A codificação dos k dígitos de informação representados por M(X) é equivalente ao cálculo de C(X), o resto da divisão de X^{n-k} M(X) por G(X). Essa operação pode ser feita por um circuito divisor (Figura 2.2) que é um RD (registrador de deslocamento) de (n-k) estágios com lógica de realimentação de acordo com o polinômio gerador $G(X) = 1 + g_1X + g_2X^2 + \ldots + g_{n-k-1}X^{n-k-1} + X^{n-k}$.

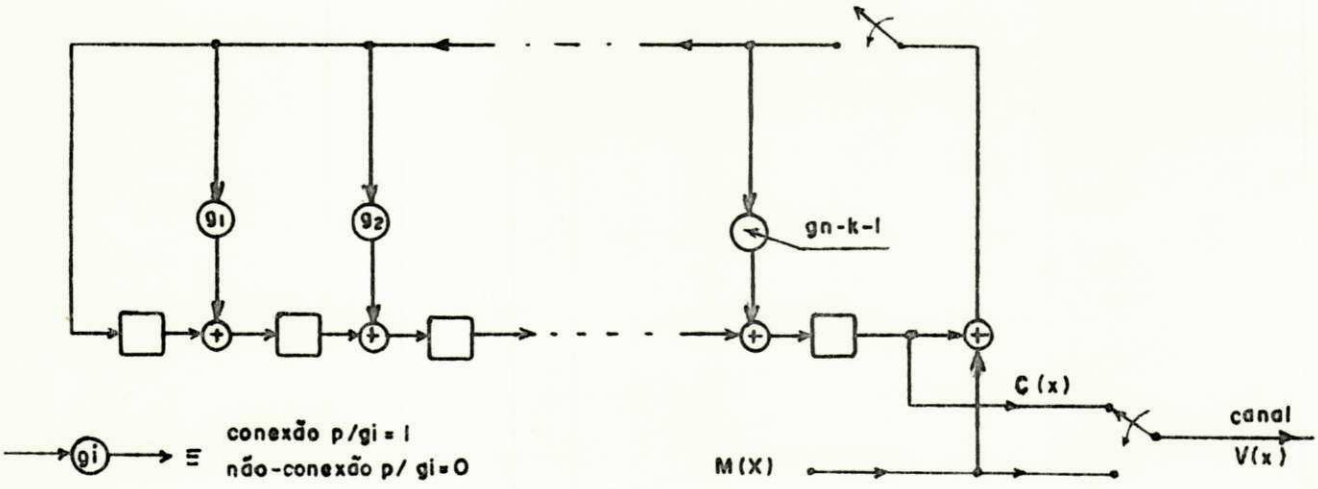


Fig. 2.2 - Codificador com RD de (n-k) estágios

O procedimento na codificação pode ser descrito como se segue. Com a chave fechada, os k dígitos de

informação $M(X) = m_0 + m_1 X + \ldots + m_{k-1} X^{k-1}$ são deslocados para o RD e, simultaneamente, para o canal de transmissão. Assim que os k dígitos de informação tenham entrado no RD, o conteúdo dos (n-k) estágios do RD representa os (n-k) dígitos de paridade. Então, a chave é aberta, cortando-se a realimentação e o conteúdo do RD é deslocado para fora e enviado para o canal. Os (n-k) dígitos de paridade $C(X) = c_0 + c_1 X + \ldots + c_{n-k-1} X^{n-k-1}$ com os k dígitos de informação $X^{n-k} M(X)$ formam a palavra-código a ser enviada $V(X) = C(X) + X^{n-k} M(X)$.

2.3.4 - CODIFICAÇÃO COM k ESTÁGIOS

Desde que G(X) divide $X^n + 1$ (Teorema 2.3) tem-se

$$X^n + 1 = H(X)\, G(X)$$

onde $H(X) = h_0 + h_1 X + \ldots + h_k X^k$.

Pode ser mostrado que o polinômio H(X) especifica completamente o código cíclico (n,k) com polinomio gerador G(X). Seja V(X) uma palavra-código, i.é.,

$$V(X) = P(X)\, G(X)$$

onde $V(X) = v_0 + v_1 X + \ldots + v_{n-1} X^{n-1}$. Multiplicando-se V(X) por H(X) tem-se

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 - Tel (083) 321 7222-R 355
58.100 - Campina Grande - Paraíba

$$V(X)\,H(X) = P(X)\,G(X)\,H(X)$$
$$= P(X)\,(X^n + 1)$$
$$= X^n P(X) + P(X)$$

Como o grau de P(X) é no máximo k-1, tem-se que os termos X^k, X^{k+1}, ..., X^{n-1} não aparecem em $X^n P(X) + P(X)$. Assim, os coeficientes de X^k, X^{k+1}, ..., X^{n-1} na expressão de V(X)·H(X) devem ser zeros, i.é.,

$$\sum_{i=0}^{k} h_i\, v_{n-i-j} = 0 \qquad \text{para } 1 \leq j \leq n-k$$

onde $h_0 = 1$ e $h_k = 1$. Resulta daí que

$$v_{n-k-j} = \sum_{i=0}^{k-1} h_i\, v_{n-i-j} \qquad \text{para } 1 \leq j \leq n-k$$

Essa última expressão representa uma equação diferença (Lin, 1970) e provê uma regra para o cálculo dos dígitos de paridade v_{n-k-1}, v_{n-k-2}, ..., v_0 dados os k dígitos de informação v_{n-1}, v_{n-2}, ..., v_{n-k}. Então, tem-se que o código cíclico (n,k) gerado pelo polinômio G(X) é completamente especificado pelo polinômio $H(X) = \frac{X^n + 1}{G(X)}$. Esse polinômio é conhecido como POLINÔMIO PARIDADE do código cíclico gerado por G(X).

O diagrama de circuito apresentado na Figura 2.3 mostra um codificador com RD de k estágios baseado na equação diferença deduzida acima.

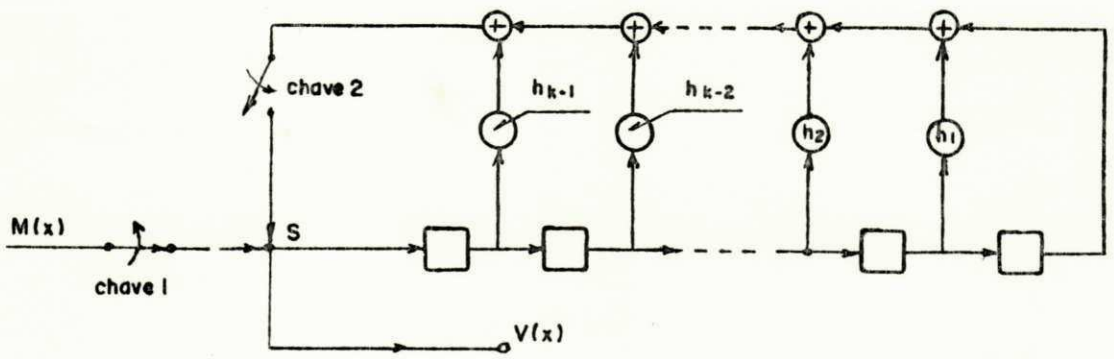


Fig. 2.3 - Codificador com RD de k estágios

O procedimento na codificação pode ser descrita da seguinte forma. Com a chave 1 fechada e a chave 2 aberta, os k dígitos de informação $M(X) = m_0 + m_1 X + \ldots + m_{k-1} X^{k-1}$ são deslocados simultaneamenteo para o RD e para o canal. Assim que os k dígitos de informação tenham entrado no RD, a chave 1 é aberta e a chave 2 é fechada. O primeiro dígito de paridade.

$$v_{n-k-1} = h_0 v_{n-1} + h_1 v_{n-2} + \ldots + h_{k-1} v_{n-k}$$

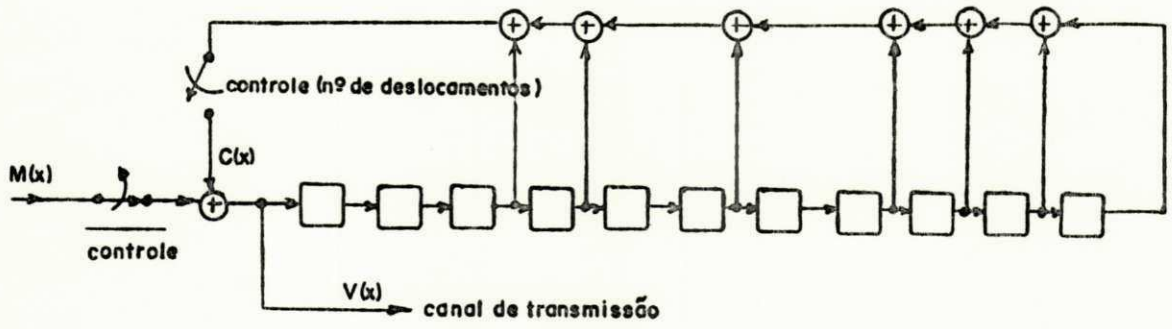
$$= m_{k-1} + h_1 m_{k-2} + \ldots + h_{k-1} m_0$$

é formado e aparece em S. O RD é então deslocado de uma posição. O primeiro dígito de paridade é deslocado simultaneamente para o canal e para o primeiro estágio do RD e o segundo dígito de paridade

$$v_{n-k-2} = h_0 v_{n-2} + h_1 v_{n-3} + \ldots + h_{k-1} v_{n-k-1}$$
$$= m_{k-2} + h_1 m_{k-3} + \ldots + h_{k-2} m_0 + h_{k-1} v_{n-k-1}$$

aparece em S. Esse procedimento se repete até que os (n-k) dígitos de paridade tenham sido deslocados para o canal. Após, a chave 1 é fechada e a chave 2 é aberta deixando o codificador pronto para receber o próximo bloco de informação.

Na figura 2.4 é apresentado o codificador com k estágios para o código (15,11) gerado por $G_5(X)=X^4+X+1$.



$$H_5(X) = 1 + X + X^2 + X^3 + X^5 + X^7 + X^8 + X^{11}$$

Fig. 2.4 - Codificador com 11 estágios para o código-cíclico (15,11) gerado por $G_5(X) = X^4+X+1$.

De uma maneira geral, comparando-se os tipos de codificadores apresentados, pode-se dizer que para (n-k) > k, o codificador com RD de k estágios é mais econômico enquanto que para (n-k) < k o codificador com RD de (n-k) estágios torna-se preferível. Todavia, na implementação práti

ca com circuitos de pequena e média integração, observa-se que o codificador com RD de k estágios é mais vantajoso, mesmo quando (n-k) for menor que k, pelo fato de não requerer portas lógicas externas entre os estágios do registrador.

2.3.5 - CÁLCULO DA SÍNDROME

O decodificador, na recepção de uma n-upla R(X), tem por função verificar se o vetor recebido é ou não uma palavra-código e recuperar o vetor transmitido V(X). No caso de códigos cíclicos tanto o cálculo da síndrome como a subsequente correção de erros são relativamente bem mais simples do que no caso dos códigos lineares em geral.

Como num código cíclico todas palavras-códigos são múltiplas do polinômio gerador do código, o primeiro passo do decodificador é verificar se a n-upla R(X) é divisível por G(X). O resto dessa divisão é a síndrome e, se ela for zero assume-se que não ocorreram erros na transmissão. Uma síndrome diferente de zero indica que o decodificador detetou erros e a correção dos erros pode então ser procedida.

A n-upla recebida R(X) pode ser escrita como

$$R(X) = V(X) + E(X)$$

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso 882 Tel (083) 321 7222-R 355
58.100 - Campina Grande - Paraíba

onde V(X) é o vetor transmitido e $E(X) = e_0 + e_1 X + \ldots + e_{n-1} X^{n-1}$ é o padrão de erros introduzido pelo ruído no canal.

A síndrome $S(X) = s_0 + s_1 X + \ldots + s_{n-k-1} X^{n-k-1}$, resultado da divisão de R(X) por G(X)

$$R(X) = P(X)\ G(X) + S(X)$$

é um polinômio de grau (n-k-1) ou menor. Essa última expressão pode ser escrita como

$$V(X) + E(X) = P(X)\ G(X) + S(X)$$

ou

$$M(X)\ G(X) + E(X) = P(X)\ G(X) + S(X)$$

ou ainda

$$E(X) = \left[P(X) + M(X) \right]\ G(X) + S(X)$$

Decorre desse último resultado uma relação bem definida entre a síndrome e o padrão de erros, i.é.,

$$S(X) = \text{resto de } \left\{ \frac{E(X)}{G(X)} \right\}$$

Diferentes polinômios E(X) podem resultar numa mesma síndrome S(X), mas assume-se que E(X) é o polinômio de menor peso satisfazendo a relação

$$S(X) = \text{resto de } \{\frac{E(X)}{G(X)}\}$$

Os circuitos usados no cálculo da síndrome são semelhantes àqueles da codificação de códicos cíclicos. Assim, registradores de cálculo de síndrome de k ou (n-k) estágios podem ser utilizados. A Figura 2.5 mostra um diagrama de circuito para o cálculo da síndrome com RD de (n-k) estágios.

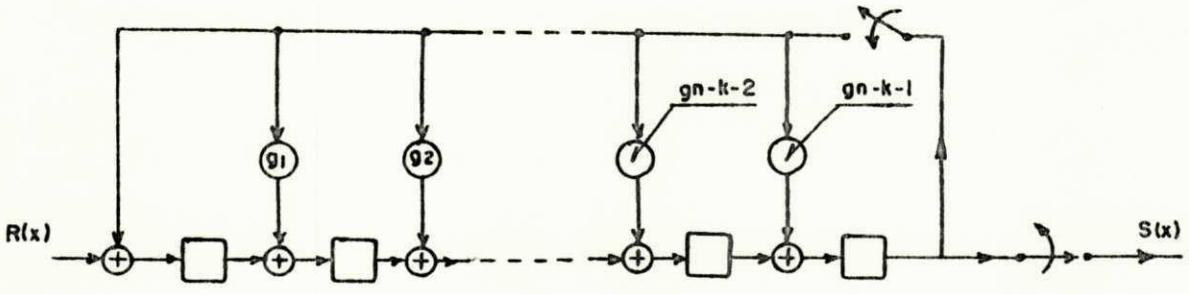


Fig. 2.5 - Cálculo da Síndrome com RD de (n-k) estágios

Inicialmente o conteúdo de RD é zero. A n-upla recebida é deslocada para o RD e, após n deslocamentos ("clock"), o registrador tem como conteúdo a síndrome. Antes de receber a próxima n-upla o RD precisa apagar o conteúdo de seus estágios, i.é., torná-los zero.

Um RD de k estágios para o cálculo da síndrome é mostrado na Figura 2.6. Nesse circuito a n-upla recebida é deslocada para o registrador com as chaves 1 fechada e 2, 3 e 4 abertas. Após k deslocamentos as chaves 2, 3 e 4

são fechadas e a chave 1 é aberta. Então os dígitos de paridade recalculados são somados módulo-2 aos recebidos formando os dígitos da síndrome na saída.

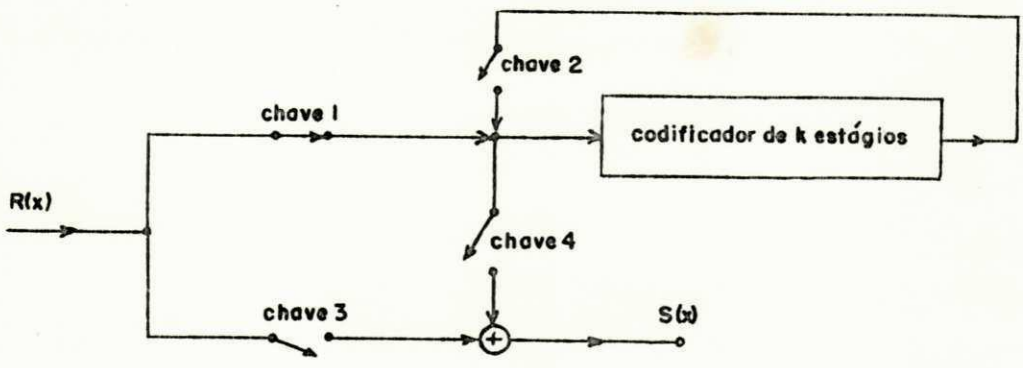


Fig. 2.6 - Cálculo da Síndrome com RD de k estágios

A decodificação de códigos corretores de erros aleatórios é, em geral, um problema difícil e, na maioria das situações práticas a complexidade do decodificador é uma limitação na escolha dos códigos a serem utilizados. Nas seções seguintes alguns dos algoritmos mais importantes para a decodificação de códigos cíclicos corretores de erros aleatórios são apresentados com comentários sobre suas vantagens e limitações.

2.3.6 - DECODIFICADOR DE MEGGITT

Um decodificador geral de um código cíclico (n,k) é mostrado na Figura 2.7 (Lin, 1970).

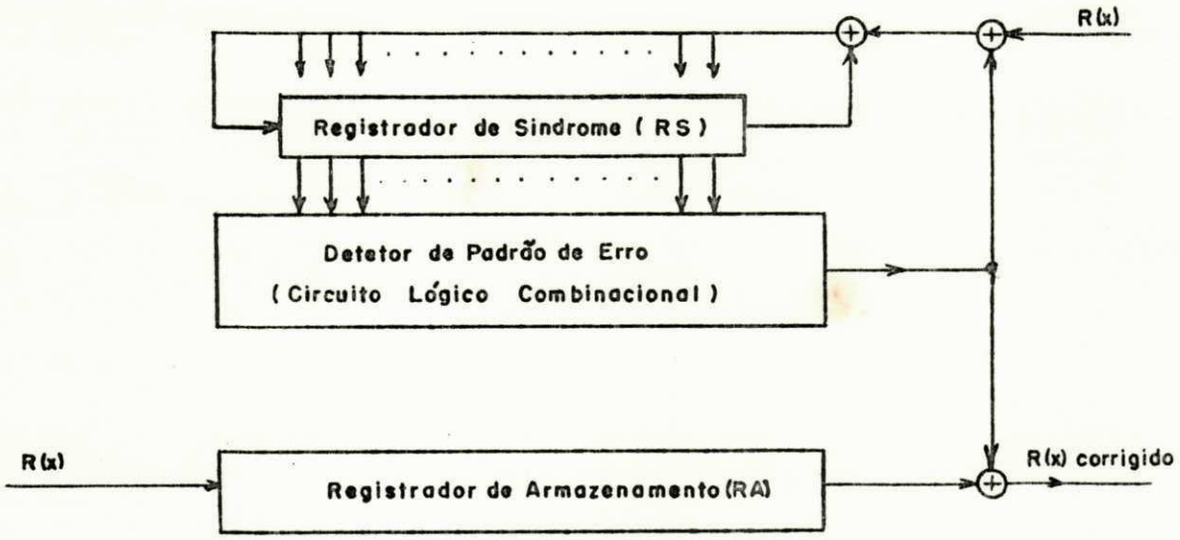


Fig. 2.7 - Decodificador de Meggitt

O procedimento na correção de erros pode ser descrito como se segue. A síndrome é formada pelo deslocamento completo da n-upla recebida no registrador de síndrome. Ao mesmo tempo, a n-upla recebida é armazenada no registrador de armazenamento RA. A síndrome formada é, então, lida pelo detetor de padrão de erros que é um circuito lógico combinacional projetado de modo que sua saída seja "1" se e só se a síndrome no registrador de síndrome RS corresponder a um padrão de erros E(X), corrigível, com um erro na posição de maior ordem. Assim, se um "1" aparece na saída do detetor, assume-se que o dígito do estágio mais à direita do RA, i.é., o primeiro dígito recebido, está errado e deve ser corrigido. Caso um "0" apareça na saída do detetor, assume-se que o primei

ro dígito recebido está correto e nenhuma correção se faz necessária. Dessa forma, a saída do detetor é um valor binário estimativa do erro para o dígito mais à direita do RA. Se o primeiro dígito recebido foi detetado estar errado ele é, então, corrigido pela saída do detetor através do somador módulo-2 na saída do RA. A saída do detetor realimenta, também, o RS para remover o efeito do erro no cálculo da síndrome. Esse procedimento se repete até que o vetor recebido tenha sido completamente lido do RA.

Após o valor recebido ter sido inteiramente lido do RA, os erros que, por ventura, tenham ocorrido na transmissão terão sido corrigidos se estiverem dentro da capacidade de correção do código.

Esse decodificador, em princípio, se aplica a qualquer código cíclico todavia, sua aplicação prática é inteiramente dependente da implementação prática do circuito lógico combinacional detetor de padrão de erros, i.é., da capacidade de armazenamento dos padrões de erro corrigíveis. A utilização de circuitos lógicos tais como PROM's pode reduzir significativamente o problema de armazenamento (Rocha, 1976).

2.3.7 - DECODIFICAÇÃO POR "ERROR-TRAPPING"

A decodificação por "error-trapping", que é uma variação prática da decodificação de Meggitt apresentada anteriormente, é descrita resumidamente nesta seção.

Seja V(X) o vetor-código de um código cíclico binário (n,k) com capacidade de correção de t erros aleatórios e R(X) o vetor recebido quando da transmissão de V(X). O padrão de erros causados pelo ruído no canal é E(X) = R(X) + V(X). A síndrome S(X) de R(X) é o resultado da divisão do padrão de erros E(X) pelo polinômio gerador G(X), i.é.,

$$E(X) = P(X)\ G(X) + S(X)$$

Se os erros de E(X) estão concentrados nas (n-k) posições, $1, X, \ldots, X^{n-k-1}$ de R(X), então E(X) é um polinômio de grau no máximo igual a (n-k-1). Dessa forma tem-se que P(X) = 0 e E(X) = S(X), i.é., se os erros em R(X) estão concentrados nas (n-k) posições de paridade, então a síndrome de R(X) é identica ao padrão de erros E(X). Assim, a correção dos erros pode ser feita pela simples soma módulo-2 da síndrome com os (n-k) dígitos de paridade recebidos.

Suponha agora que os erros não estejam concentrados nas (n-k) posições de paridade de R(X) mas estejam concentradas em (n-k) posições consecutivas (incluindo os casos extremos) X^i, X^{i+1}, ..., $X^{(n-k)+i-1}$. Devido a natureza cíclica do código, tais padrões de erros podem ser deslocados inteiramente para a seção de paridade da n-upla recebida. Após (n-i) deslocamentos cíclicos de R(X), os erros estarão deslocados para as (n-k) posições de paridade do vetor recebido, ciclicamente deslocado, $R^{(n-i)}(X)$. Então a síndrome de $R^{(n-i)}(X)$ é idêntica aos erros concentrados nas posições X^i, X^{i+1}, ..., $X^{(n-k)+i-1}$ de R(X) o que permite a correção de erros como no

caso anterior.

A decodificação por "error-trapping" é um dos modos mais efetivos para se decodificar códigos cíclicos de baixa eficiência e/ou de baixa capacidade de correção de erros aleatórios. Todavia, para códigos longos com alta eficiência e grande capacidade de correção de erros aleatórios, esse modo de decodificação torna-se ineficiente por causa de sua inerente incapacidade de correção de erros que não estejam concentrados em (n-k) posições consecutivas (Lin, 1970). Em geral, para um código cíclico ser decodificado eficientemente por "error-trapping", a seguinte condição deve ser satisfeita:

$n/k > t$ (Peterson, 1972)

2.3.8 - DECODIFICAÇÃO POR FUNÇÃO DE MAIORIA

A decodificação por função de maioria é um outro modo bastante eficiente para a decodificação de certas classes de códigos cíclicos. Historicamente, Reed (1954) foi o primeiro a sugerir a idéia de decodificação por função de maioria para uma classe de códigos chamada códigos de Reed-Muller (Muller, 1954). Mais tarde extensões e generalizações do trabalho de Reed foram feitas por muitos teóricos da codificação. A primeira formulação unificada da decodificação por função de maioria deve-se a Massey (1963), (Lin, 1970). Importante também, mencionar a contribuição de Rudolph (1967) na

construção de códigos decodificáveis por funções de maioria.

a - Somas de Paridade

A síndrome $[S]$ de um vetor recebido em um código cíclico (n,k) com matriz $[H]$ pode ser escrita como

$$[S] = [s_0, s_1, s_2, \ldots, s_{n-k-1}] = [E]\,[H]^T$$

onde $[E] = [e_0, e_1, e_2, \ldots, e_{n-1}]$ representa um padrão de erros. Expandindo-se a expressão acima obtém-se o seguinte conjunto de equações

$$\begin{aligned}
s_0 &= e_0 + c_{00}e_{n-k} + \ldots + c_{0,k-1}\,e_{n-1} \\
s_1 &= e_1 + c_{10}e_{n-k} + \ldots + c_{1,k-1}\,e_{n-1} \\
s_2 &= e_2 + c_{20}e_{n-k} + \ldots + c_{2,k-1}\,e_{n-1} \\
&\vdots \\
s_{n-k-1} &= e_{n-k-1} + c_{n-k-1,0}\,e_{n-k} + \ldots + c_{n-k-1,k-1}\,e_{n-1}
\end{aligned}$$

Considere agora uma combinação linear dos dígitos da síndrome

$$A = a_0 s_0 + a_1 s_1 + \ldots + a_{n-k-1}\,s_{n-k-1}$$

onde $a_i \in \{0, 1\}$. Do conjunto de equações resultantes da expansão do produto $[E]\,[H]^T$ com a expressão de A acima tem-se

$$A = b_0e_0 + b_1e_1 + \ldots + b_{n-1}e_{n-1}$$

onde $b_i \in \{0, 1\}$. Essa equação acima é chamada SOMA DE PARIDADE. Um dígito de erro e_i é dito ser verificado por A se o coeficiente b_i de A for "1".

Um conjunto de J somas de paridade A_1, A_2, ... A_j é dito ser ORTOGONAL a um dígito de erro e_ℓ se e_ℓ é verificado por cada uma das somas A_j do conjunto e nenhum outro dígito de erro é verificado por mais de uma soma de paridade.

Se, ao invés, de se ter e_ℓ comum a todas as equações do conjunto, tem-se um conjunto $E = \{e_i, e_k, \ldots e_\ell\}$ de dígitos de erro comum a todas as equações então a seguinte definição é feita.

Um conjunto de J somas de paridade A_1, A_2, ..., A_j é dito ser ortogonal ao conjunto de dígitos de erro $E = \{e_i, e_k, \ldots, e_\ell\}$. Se e só se o conjunto E é verificado por todas os J somas de paridade e nenhuma posição de erro fora do conjunto é verificada por mais de uma soma de paridade.

Observe que

$$[R][H]^T = [V + E][H]^T = [V][H]^T + [E][H]^T$$

Mas $[V][H]^T = 0$, então

$$[R][H]^T = [E][H]^T$$

Resulta daí, que a formação de somas de paridade ortogonal a um dígito de erro e_ℓ (ou conjunto $E = \{e_i, e_k, \ldots, e_\ell\}$) é equivalente a formação de somas de paridade ortogonal ao dígito r_ℓ (ou conjunto de dígitos $\{r_i, r_k, \ldots, r_\ell\}$) do vetor recebido.

As somas de paridade ortogonal a um dígito de erro formam a base para a DECODIFICAÇÃO POR FUNÇÃO DE MAIORIA EM UM PASSO enquanto que as somas de paridade ortogonal a um conjunto de dígitos de erro constituem a base para a DECODIFICAÇÃO POR FUNÇÃO DE MAIORIA EM L-PASSOS.

b - Decodificação por Função de Maioria em Um Passo

Dado um código cíclico binário (n,k), suponha que pode-se encontrar J somas de paridade ortogonal ao dígito da posição de maior ordem da n-upla recebida e, portanto, ao dígito de maior ordem, e_{n-1}, do vetor erro. Se o vetor erro tiver peso $W_{(E)} \leq J/2$ então o vetor recebido pode ser corrigido da seguinte maneira. Se o dígito de maior ordem está correto, i.é., $e_{n-1} = 0$ então os dígitos de erro diferentes de zero estão distribuídos no máximo em J/2 somas de paridade. Assim, pelo menos J/2 somas de paridade ortogonal a e_{n-1} são iguais a $e_{n-1} = 0$. Dessa forma, se uma maioria das somas de paridade é zero, ou se resultar num empate, tem-se que o dígito de maior ordem r_{n-1} do vetor recebido está correto. Por outro lado, se $e_{n-1} = 1$, então os J/2 - 1 erros restantes estão espalhados por no máximo J/2 - 1 somas de paridade. Isso

significa que pelo menos J-(J/2 - 1) somas de paridade possuem apenas $e_{n-1} = 1$. Dessa forma, uma maioria clara das somas de paridade é igual a 1, e o dígito de maior ordem do vetor recebido, pode ser corrigido pelo decodificador.

No caso dos códigos cíclicos, a segunda posição de maior ordem da n-upla recebida (correspondente a e_{n-2}) pode ser corrigida de maneira idêntica à descrita acima porque após um deslocamento cíclico, e_{n-2} ocupa a posição de e_{n-1}. Uma vez que o efeito do erro e_{n-1} tenha sido removido das somas de paridade podemos proceder a estimativa de e_{n-2}. Esse processo continua até que a palavra-código tenha sido decodificada completamente.

O processo de decodificação por função de maioria é eficiente quando J é igual ou muito próximo a d-1 (d é a distância mínima do código) pois $t = \left\lceil \frac{d-1}{2} \right\rceil \geq J/2$ é a capacidade de correção de erros aleatórios do código. Códigos onde J = d-1 são ditos serem COMPLETAMENTE ORTOGONALIZÁVEIS EM UM PASSO.

Dependendo do modo como as somas de paridades são construídas, existem dois tipos de implementação basicamente diferentes. São conhecidos por decodificadores por função de maioria Tipo I e Tipo II (Lin, 1970). O decodificador Tipo I utiliza o conjunto de somas de paridade em termos dos dígitos da síndrome enquanto que o Tipo II usa o fato de que o conjunto de J somas de paridade ortogonal a e_{n-1} é equivalente a J vetores no espaço das linhas da matriz paridade $[H]$ ortogonal ao componente de maior ordem.

Exemplo. Considere o código cíclico (15,7) com polinomio gerador $G_4(X) = X^8 + X^7 + X^6 + X^4 + 1$ e matriz de paridade $[H_4]$ (ver exemplo da seção 2.3.2). Seja $[E] = [e_0, e_1, e_2, \ldots, e_{14}]$ o vetor erro e $[R] = [r_0, r_1, r_2, \ldots, r_{14}]$ o vetor recebido. A síndrome correspondente ao vetor $[E]$ é

$$[S] = [s_0,s_1,s_2,s_3,s_4,s_5,s_6,s_7] = [E][H_4]^T = [R][H_4]^T$$

Daí resultam as seguintes equações

$$s_0 = e_0 + e_8 + e_9 + e_{11}$$

$$s_1 = e_1 + e_9 + e_{10} + e_{12}$$

$$s_2 = e_2 + e_{10} + e_{11} + e_{13}$$

$$s_3 = e_3 + e_{11} + e_{12} + e_{14}$$

$$s_4 = e_4 + e_8 + e_9 + e_{11} + e_{12} + e_{13}$$

$$s_5 = e_5 + e_9 + e_{10} + e_{12} + e_{13} + e_{14}$$

$$s_6 = e_6 + e_8 + e_9 + e_{10} + e_{13} + e_{14}$$

$$s_7 = e_7 + e_8 + e_{10} + e_{14}$$

Quatro somas de paridade ortogonal a e_{14} podem ser formadas como se segue

$$A_1 = s_3 = e_3 + e_{11} + e_{12} + e_{14}$$

$$A_2 = s_1 + s_5 = e_1 + e_5 + e_{13} + e_{14}$$

$$A_3 = s_0 + s_2 + s_6 = e_0 + e_2 + e_6 + e_{14}$$

$$A_4 = s_7 = e_7 + e_8 + e_{10} + e_{14}$$

As somas de paridade ortogonal a e_{14} expressas em termos dos dígitos da síndrome são usados para se construir o decodificador por função de maioria em um passo Tipo I mostrado na Figura 2.8 enquanto que as somas de paridade expressas em termos dos componentes do vetor erro são usadas na construção do decodificador Tipo II mostrado na Figura 2.9.

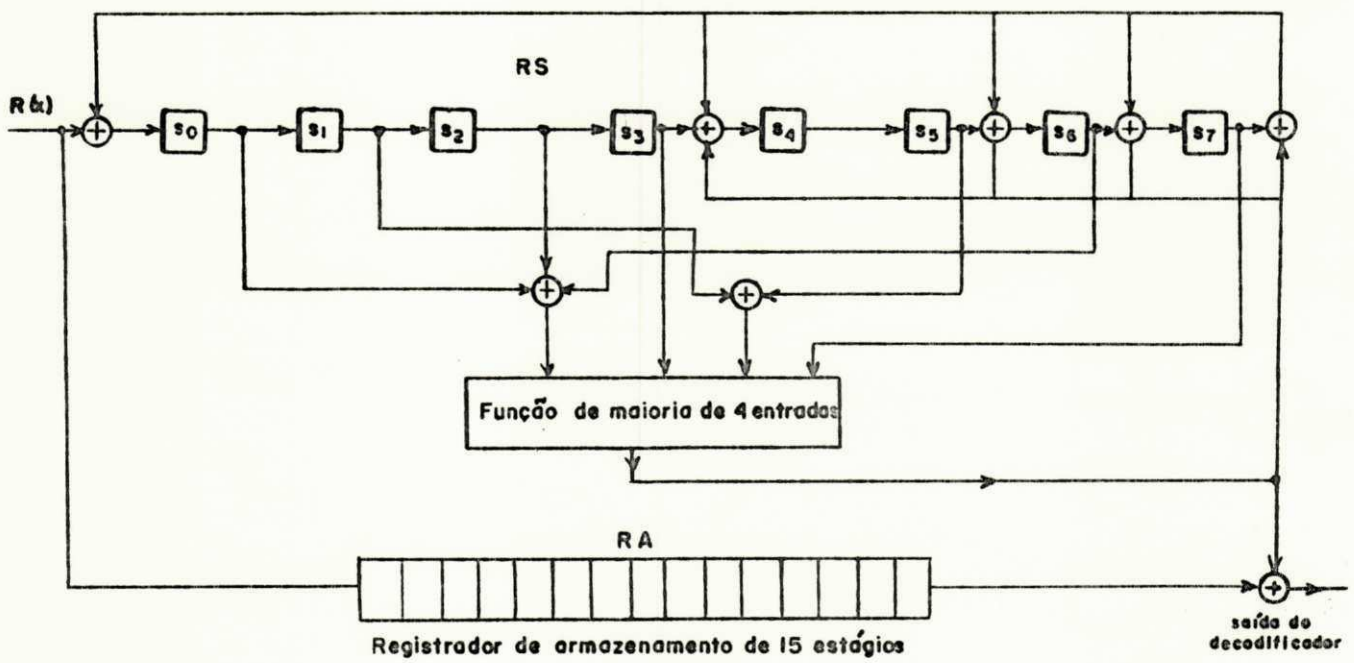


Fig. 2.8 - Decodificador por função de maioria em um passo do Tipo I para o código cíclico (15,7) com $G_4(X) = X^8 + X^7 + X^6 + X^4 + 1$.

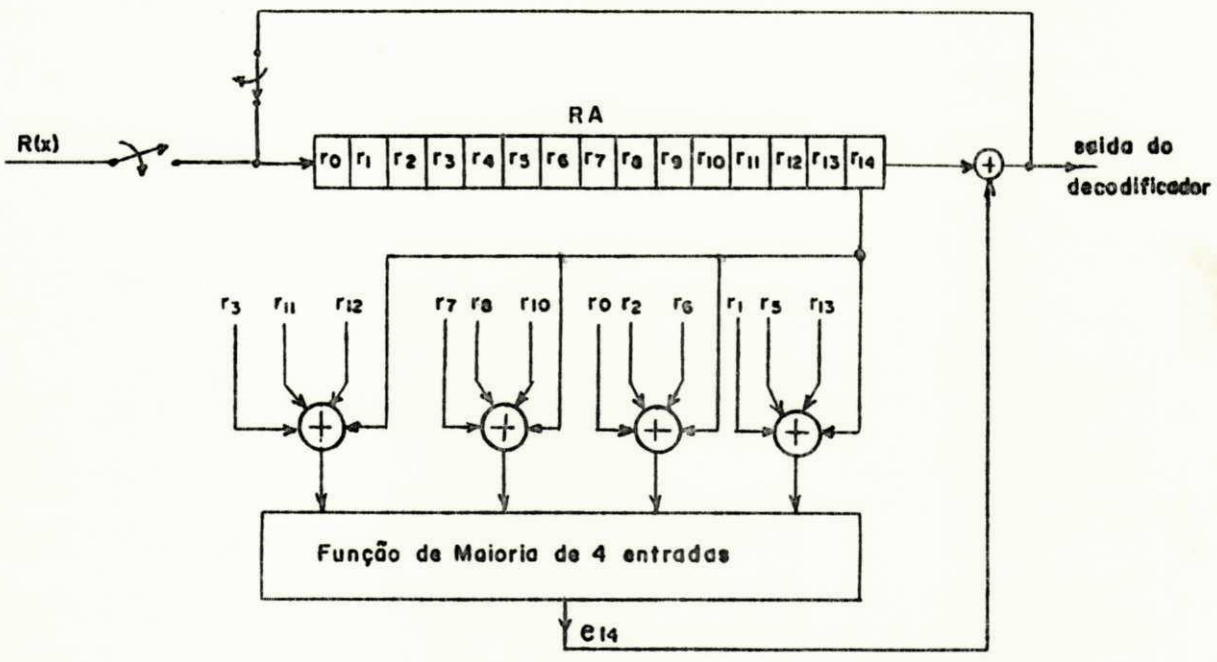


Fig. 2.9 - Decodificador por função de maioria em um passo do Tipo II para o código cíclico (15,7) com $G_4(X) = X^8 + X^7 + X^6 + X^4 + 1$.

Funcionamento do decodificador Tipo I. A síndrome da n-upla recebida é calculada da maneira usual. Após o cálculo da síndrome, a saída do circuito função de maioria é usada para corrigir os dígitos que forem sendo lidos do RA (registrador de armazenamento) e, ao mesmo tempo, realimenta o RS (registrador de cálculo da síndrome) para remover

os efeitos dos erros corrigidos. Após o último dígito da palavra-código armazenada no RA ter sido lido, o conteúdo do RS deve ser zero. Caso contrário, um padrão de erros não-corrigível foi detetado.

Funcionamento do decodificador Tipo II. A n-upla recebida é armazenada no RA. Nos próximos n deslocamentos, a saída do circuito função de maioria é adicionada módulo-2 aos dígitos da n-upla recebida para a correção dos possíveis erros. O resultado dessa soma é realimentado para o RA afim de eliminar os efeitos dos erros corrigidos. No final de n deslocamentos, o RA contém a palavra-código corrigida e as entradas do circuito função de maioria são zeros. Caso contrário, um padrão de erros não-corrigível foi detetado.

c - Decodificação por Função de Maioria em L-Passos

A decodificação por função de maioria em um passo é de fácil implementação porém, são poucas as classes de códigos cíclicos ortogonalizáveis em um passo. A implementação com conjuntos de somas de paridade ortogonal a um conjunto de dígitos de erro, que generaliza a idéia de ortogonalidade, juntamente com o procedimento descrito acima para a decodificação em um passo, permite um número maior de códigos cíclicos serem decodificáveis por função de maioria. Vários níveis de circuitos função de maioria são usados. A cada nível uma soma de dígitos ortogonal é estimada. Esse processo continua até que um conjunto de J e, não mais, somas de paridade

UFPb / BIBLIOTECA / BRAS

ortogonal a um simples dígito de erro é obtido. Daí o valor do dígito de erro pode ser estimado como no caso de um passo.

Exemplo. Considere o código (15,11) como polinômio gerador $G_5(X) = X^4 + X + 1$ e matriz de paridade $[H_5]$ (ver exemplo da seção 2.3.2). Seja $[E] = [e_0, e_1, \ldots, e_{14}]$ o vetor erro. Pode-se formar os seguintes conjuntos de somas de paridade

$$A_1^{(1)} = e_0 + e_4 + e_8 + e_{10} + e_7 + e_{12} + e_{13} + e_{14}$$

$$A_2^{(1)} = e_3 + e_6 + e_9 + e_{11} + e_7 + e_{12} + e_{13} + e_{14}$$

$$A_1^{(2)} = e_0 + e_1 + e_{11} + e_{13} + e_5 + e_8 + e_9 + e_{14}$$

$$A_2^{(2)} = e_2 + e_3 + e_7 + e_{10} + e_5 + e_8 + e_9 + e_{14}$$

onde o conjunto de somas de paridade $A_1^{(1)}$, $(1 \leq i \leq 2)$, é ortogonal ao conjunto de dígitos de erro $E^{(1)} = \{e_7, e_{12}, e_{13}, e_{14}\}$ enquanto que o conjunto de somas de paridade $A_1^{(2)}$, $(1 \leq i \leq 2)$, é ortogonal ao conjunto $E^{(2)} = \{e_5, e_8, e_9, e_{14}\}$. As somas de paridade relativas aos conjuntos de dígitos de erro $E^{(j)}$ $(1 \leq j \leq 2)$

$$A_1^{(3)} = e_7 + e_{12} + e_{13} + e_{14}$$

$$A_2^{(3)} = e_5 + e_8 + e_9 + e_{14}$$

formam um conjunto de somas de paridade ortogonal ao dígito

de erro e_{14}. Tem-se então, que esse código pode ser decodificado por Função de maioria em dois passos. O decodificador do Tipo II é mostrado na Figura 2.10.

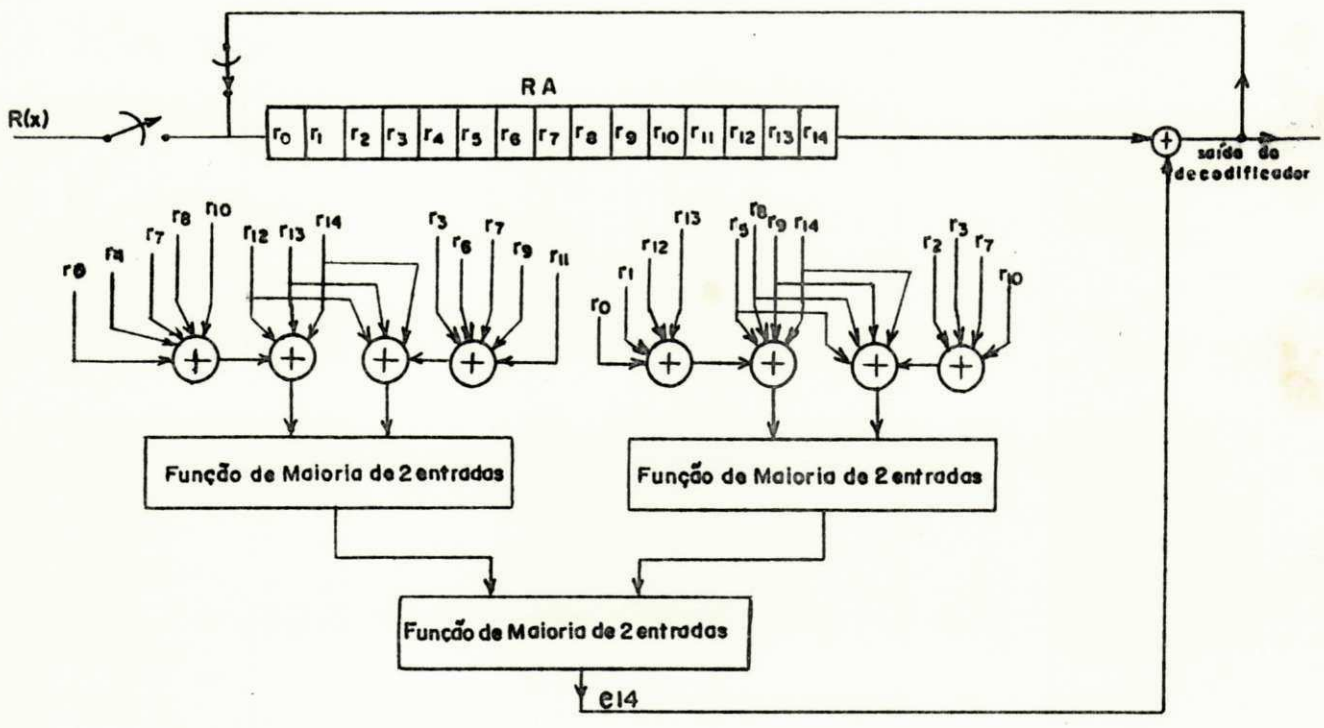


Fig. 2.10 - Decodificador por função de maioria em dois passos do Tipo II para o código cíclico (15,11) com polinômio gerador $G_5(X) = X^4 + X + 1$.

O uso de realimentação dos dígitos corrigidos, como nos exemplos dados acima, permite uma capacidade ex

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso 882 Tel (083) 321 7222-R 355
58.100 - Campina Grande - Paraíba

tra de correção para o código (Lucky et al, 1968). Por exemplo, se t + 1 erros ocorrerem e um erro no primeiro dígito recebido for corrigido satisfatoriamente então os t remanescentes podem, também, ser corrigidos. Argumentos semelhantes valem para padrões de erro de maior peso . Assim, em muitas situações, a decodificação por função de maioria permite a correção de padrões de erro com peso ligeiramente superior a t. A decodificação por função de maioria de uma maneira sequencial (Rudolph e Hartmann, 1973) pode reduzir substancialmente a complexidade do decodificador sob o compromisso de um aumento do tempo de decodificação. Outras alternativas do processo de decodificação por função de maioria foram desenvolvidas com o objetivo de aumentar a capacidade de correção de erros aleatórios (Townsend e Weldon, 1967).

2.3.9 - OUTRAS CLASSES DE CÓDIGOS CÍCLICOS

Até agora, a apresentação dos códigos cíclicos foi feita com ênfase especial aos códigos corretores de erros aleatórios, i.é., erros estatisticamente independentes. Todavia, existem códigos que foram desenvolvidos para utilização em canais onde os erros introduzidos não ocorrem independentemente, i.é., a ocorrência de um erro em um dígito aumenta a probabilidade de que o próximo dígito esteja, também, errado. Canais desse tipo são ditos terem memória e os erros são ditos acontecerem em "burst". Os códigos cíclicos corretores desse tipo de erros são chamados CÓDIGOS CÍCLICOS CORRETORES DE ERROS EM "BURST" (ver, por exemplo, Lin, 1970)).

Outras classes importantes de códigos cíclicos corretores de erros aleatórios tais como os códigos de Bose-Chaudhuri-Hocquenghem (BCH) (Lin, 1970) deixaram de ser apresentadas por conveniência, apesar de que alguns dos códigos utilizados no sistema proposto (Capítulo IV) sejam códigos cíclicos BCH.

2.4 - CÓDIGOS CÍCLICOS COM REDUNDÂNCIA VARIÁVEL

Uma capacidade de correção de erros variável pode ser interessante em sistemas de comunicações onde o efeito do ruído varia consideravelmente por períodos relativamente longos ou ainda, em sistemas cuja taxa de informação varia de uma maneira sistemática. Essa variação pode ser obtida pela manipulação conveniente do n e/ou k do código. No primeiro caso, onde o efeito do ruído é variável, mostra-se mais eficiente a variação de n e k simultaneamente. No segundo caso, que é o deste trabalho (Capítulo IV) interessa a variação da capacidade de correção de erros aleatórios conseguida através da variação de k somente, i.é., com o comprimento n do código fixo.

Como foi mostrado na seção 2.3.1, a fatoração de $X^n + 1$ permite a construção de vários códigos cíclicos binários de comprimento fixo n e número de dígitos de informação k variável. Essa variação do número de dígitos de informação e, portanto, do número de dígitos redundantes (de paridade) pode levar a uma substancial diferença na capacidade de correção de erros em um bloco de n dígitos. Fica evidente, uma consequente diferença da eficiência do bloco.

UFCG

CAPÍTULO III

MULTIPLEXAÇÃO DIGITAL

3.1 - INTRODUÇÃO

Num sistema prático de comunicações de dados, o custo das linhas (ou canais) de transmissão é determinante no custo total do sistema. Dessa forma, a minimização do número de linhas (ou canais) nos projetos de sistemas de comunicações de dados assim como, uma melhor utilização das linhas, já existentes, em termos de suas capacidades de transmissão de informação, constituem problemas importantes para os projetistas desses sistemas. Uma maneira prática de se conseguir isso é através da MULTIPLEXAÇÃO ou da CONCENTRAÇÃO que são, basicamente, métodos de se combinarem vários canais de comunicação em um único. A multiplexação, implementada por um dispositivo chamado de MULTIPLEX, se caracteriza pelo fato de

a capacidade de transferência de informação instantânea do canal de saída ser sempre maior ou igual a soma das capacidades instantâneas de cada canal na entrada do multiplex. Por outro lado, quando, potencialmente, a soma das capacidades dos canais na entrada possa exceder a capacidade do canal na saída, tem-se a concentração e, a implementação na prática é feita através de dispositivos chamados CONCENTRADORES. Na concentração é exigido um controle do fluxo de informação de modo que a taxa instantânea de informação na entrada não exceda a da saída. Isso pode ser conseguido através do armazenamento do excesso de informação no concentrador ("store-and-forward")ou pelo controle de atividade dos canais na entrada ("hold-and-forward") (Davies e Barber, 1973).

Uma das utilizações da multiplexação, ou da concentração, pode ser, por exemplo, na transmissão de informação de terminais remotos de baixa velocidade em um canal de alta capacidade, em redes cujos terminais estejam geograficamente concentrados em relação a um processador central (computador, central de comutação, etc...). O uso de um único canal de transmissão nesse tipo de rede pode ser vantajoso em relação ao uso de linhas (ou canais) ponto-a-ponto para cada terminal remoto, pois evita as linhas e modem's individuais. Na Figura 3.1 é ilustrada essa comparação. Uma outra utilização da multiplexação, ou da concentração, pode ser na rede multiponto, onde vários terminais estão dispersos geograficamente ao longo de uma linha comum com o processador central. Essa configuração de rede é ilustrada na Figura 3.2.

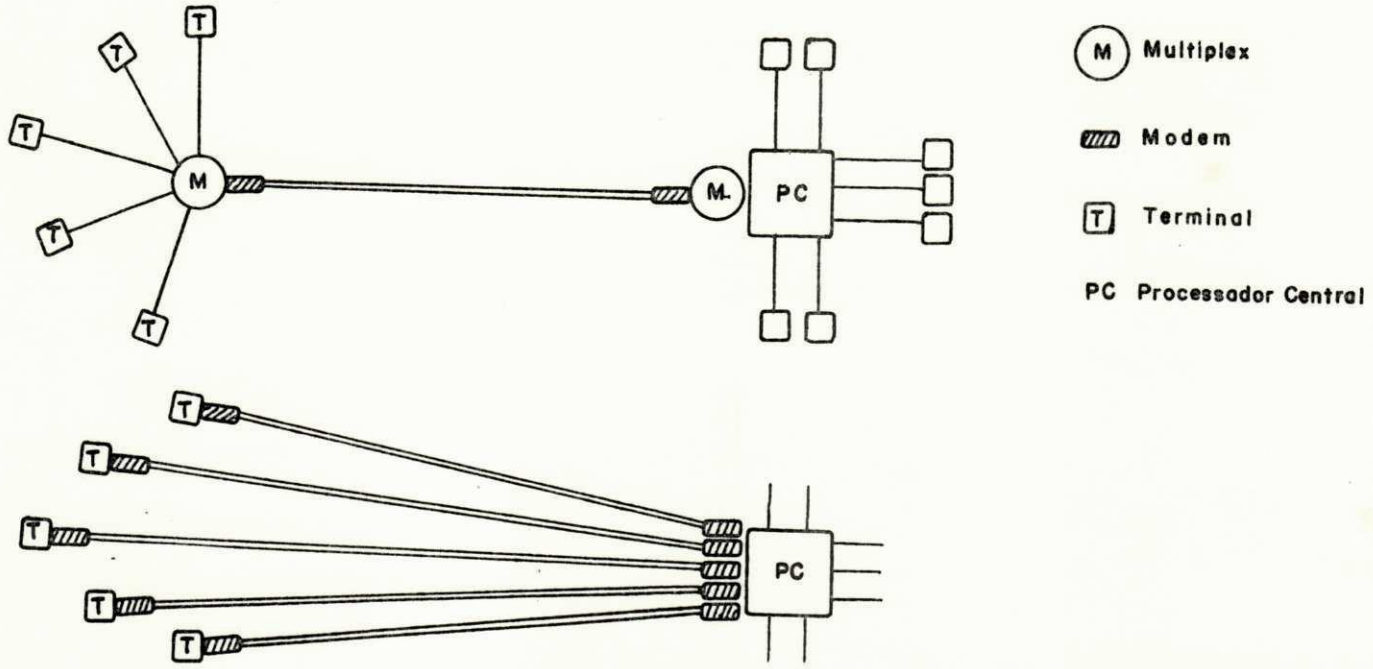


Fig. 3.1 - Configuração de linhas

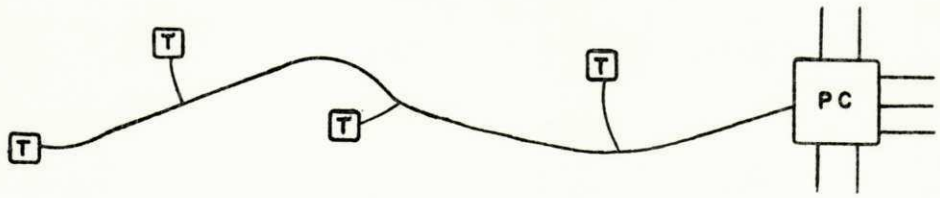


Fig. 3.2 - Configuração Multiponto

A multiplexação pode ser feita de várias maneiras e, a escolha de uma, em particular, é ditada, principalmente, por considerações sobre capacidade de canal, custos dos equipamentos associados ao processo de multiplexação, geografia da rede (configuração das linhas) e flexibilidade de inserção e desvio de canais. Em alguns casos, onde se exige alta confiabilidade na transmissão, se fazem necessárias, considerações sobre capacidade de controle de erros associada a multiplexação. Em princípio, qualquer transformação que permita a combinação dos diversos sinais correspondentes aos canais de comunicação, em um único sinal e, que na recepção,sem ruído, permita a separação desses mesmos sinais, dá origem a um sistema de multiplexação. Historicamente, o primeiro tipo de multiplexação efetivamente usado foi a MULTIPLEXAÇÃO POR DIVISÃO EM FREQUÊNCIA (MDF) onde os diversos sinais são transmitidos simultaneamente mas alocados em faixas diferentes do espectro de frequência. A reversibilidade dessa transformação reside na ortogonalidade do conjunto de funções seno e cosseno. Um outro tipo de multiplexação, bastante usado na prática, é a MULTIPLEXAÇÃO POR DIVISÃO EM TEMPO (MDT) que utiliza-se da ortogonalidade das funções bloco. Nessa transformação, cada um dos sinais ocupa todo espectro de frequências do canal de saída, mas são transmitidos em tempos diferentes. A multiplexação baseada na ortogonalidade de conjuntos de funções é chamada de MULTIPLEXAÇÃO POR DIVISÃO ORTOGONAL (MDO) (Vilar França, 1978). O diagrama em blocos de um sistema MDO é ilustrado na Figura 3.3. Pertencem, ainda, à classe de MDO, a

MULTIPLEXAÇÃO POR QUADRATURA que usa os sinais ortogonais da mesma frequência sen w_c t e cosseno w_c t e a MULTIPLEXAÇÃO POR DIVISÃO EM SEQUÊNCIA que usa a ortogonalidade do conjunto das funções de Walsh (Harmuth e Murty, 1973). Os tipos de multiplexação não pertencentes à classe de MDO são classificados, de uma maneira geral, como MULTIPLEXAÇÃO POR DIVISÃO EM CÓDIGO (MDC) (Vilar França, 1978).

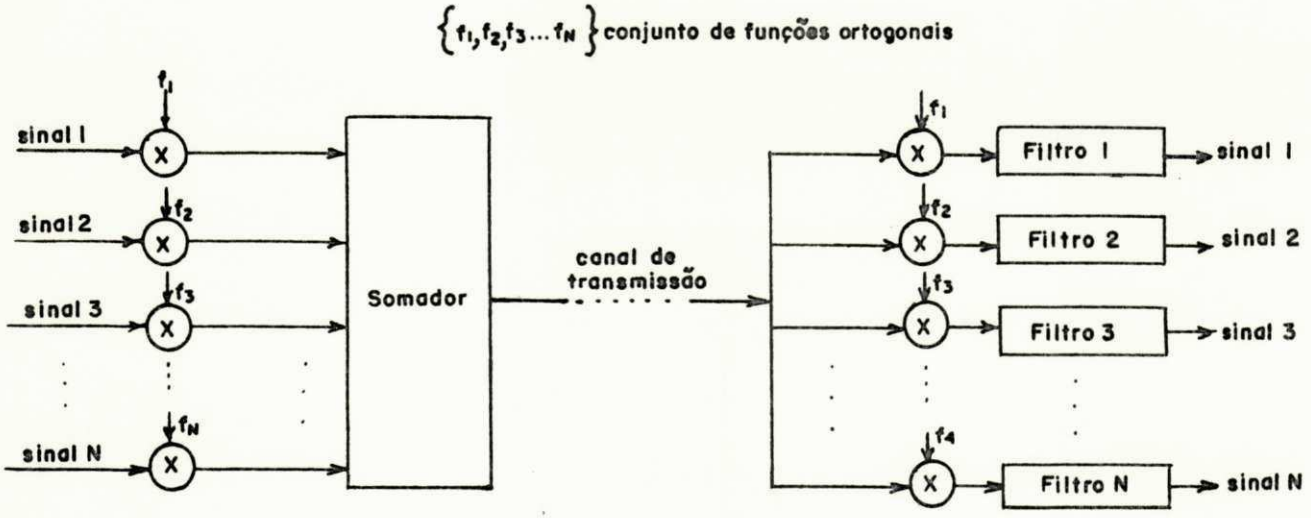


Fig. 3.3 - Sistema MDO

Pertencem a classe dos sistemas não-convencionais de multiplexação MDC, o MULTIPLEX DE WALSH TERNÁRIO, o MULTIPLEX ADAPTATIVO POR FUNÇÃO DE MAIORIA (Rocha Neto, 1975) e o MULTIPLEX

DIGITAL POR DIVISÃO EM CÓDIGOS CÍCLICOS (MDDCC) apresentado no Capítulo IV.

Neste capítulo é feito um estudo comparativo entre os dois sistemas de multiplexação mais utilizados em redes práticas de comunicações de dados, os sistemas MDF e MDT, para a seguir introduzir-se o MDDCC, um sistema de multiplexação não-convencional que incorpora capacidade de controle de erros de uma maneira adaptativa.

3.2 - MULTIPLEXAÇÃO POR DIVISÃO EM FREQUÊNCIAS (MDF)

Em sistemas MDF, um canal de comunicação limitado em frequência é dividido num grupo de canais independentes onde a cada um dos canais corresponde uma porção fixa do espectro de frequências total, comumente, chamado de CANAL DERIVADO ou FAIXA DE DADOS. A Figura 3.4 ilustra a divisão do espectro de frequências num sistema MDF típico. Cada canal derivado, centrado numa frequência específica, tem uma faixa de transmissão num extremo e uma faixa de recepção no outro extremo da faixa de dados. A informação digital transmitida ou recebida pelos canais derivados é representada, de uma maneira geral, pela presença ou ausência de algum tipo de sinal analógico nas faixas correspondentes de transmissão e recepção. O tipo de sinal analógico é caracterizado pelo método de modulação utilizado para adequar o sinal digital, correspondente à informação, ao meio de transmissão. Várias técnicas de modulação para sinais digitais podem ser usados (ver Carlson (1975)).

Uma limitação prática na eficiência de um sistema MDF é a necessidade de faixas de guarda entre os canais derivados para se evitar a interferência entre canais adjacentes. Por exemplo, nos sistemas MDF práticos operando na faixa de voz (3.400 Hz) podemos ter, tipicamente, um máximo de 2.000 bits/seg no sinal multiplexado, resultando numa utilização da ordem de 60% da faixa disponível (Doll, 1972). A

principal vantagem para o usuário dos sistemas MDF é o baixo custo em aplicações onde a capacidade do canal de voz multiplexado não é um fator restritivo (Doll, 1972). Parte da economia provém da eliminação da necessidade de modem's separados para cada terminal remoto pois o multiplex é projetado para, também, efetuar as funções de modulação e demodulação.

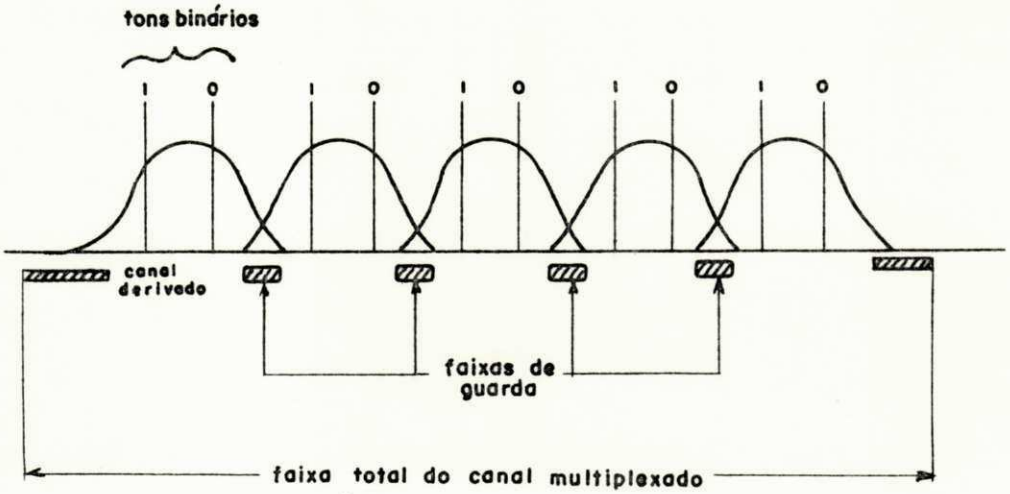


Fig. 3.4 - Divisão do espectro num sistema MDF típico

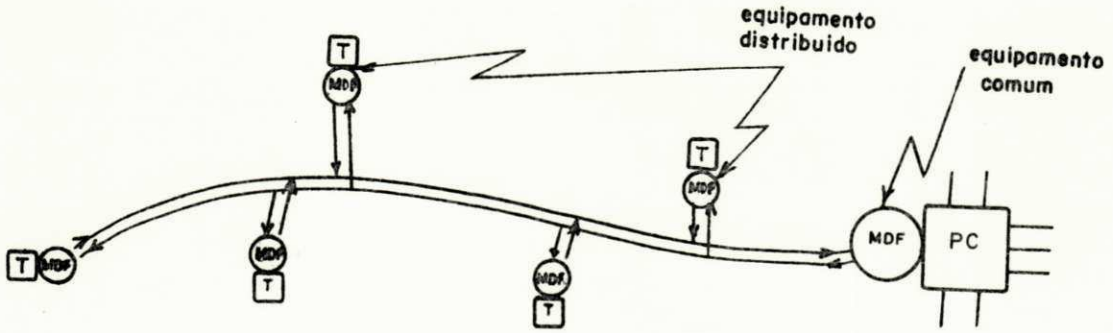


Fig. 3.5 - MDF numa configuração multiponto.

Funcionalmente, o sistema MDF consiste numa seção de equipamento comum e em equipamentos distribuidos pelos terminais conforme ilustrado na Figura 3.5. Do ponto de vista de flexibilidade, o sistema MDF apresenta as facilidades de inserção e desvio de canais em pontos intermediários ao longo do canal multiplexado o que é particularmente atrativo numa configuração multiponto (Smith, 1976). Por outro lado, a descentralização do equipamento num sistema MDF multiponto pode resultar na prática uma desvantagem pela dificuldade de manutenção e a consequente redução da confiabilidade (Davies e Barber, 1973). Outro fator que restringe a confiabilidade dos sistemas MDF é a diversidade de componentes em termos de frequências (filtros, osciladores, etc.).

3.3 - MULTIPLEXAÇÃO POR DIVISÃO EM TEMPO (MDT)

Nos sistemas MDT, em contraste com MDF, utiliza-se todo espectro de frequência da linha dividindo-a, porém, em segmentos de tempo entre os canais. A operação básica de um sistema MDT é ilustrada na Figura 3.4. O sistema varre cada canal numa sequência no tempo, alocando os dígitos (no caso binário, bits) ou caracteres ("bytes") de cada canal na formação de uma sequência contínua no tempo de bits ou bloco de bits ("bytes") que é enviada para a linha através de um modem. Esse ciclo de varredura fixo ou variável se repete continuamente permitindo, então, a transmissão de dados dos diversos canais em um único canal de alta velocidade.

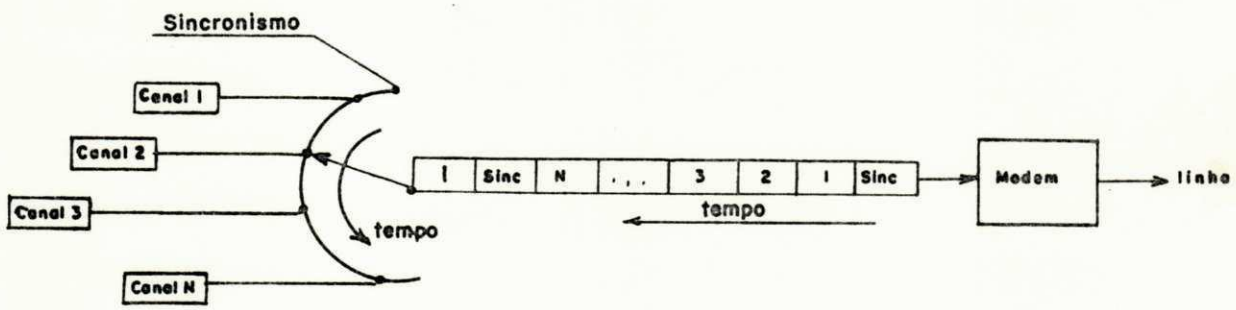


Fig. 3.6 - Sistema MDT

Na recepção o sistema deve efetuar a operação inversa separando os bits ou bytes da longa sequência de

bits e endereçando-os aos destinatários (terminal, armazenador, etc.). Para se assegurar o correto endereçamento dos bits ou bytes na recepção, se faz necessária a identificação dos mesmos quanto ao endereço.

Quando o ciclo de varredura dos canais for fixo, i.é., a cada canal na entrada do sistema se assegura um número fixo de segmentos de tempo durante o ciclo de varredura, o sistema é chamado de MULTIPLEXAÇÃO POR DIVISÃO EM TEMPO SÍNCRONO (MDTS) (Doll, 1972). A identificação dos bits ou bytes no sistema MDTS é feito através do conhecimento prévio da sequência de varredura pelo receptor e, do sincronismo de ciclo com o transmissor. O sincronismo de ciclo pode ser conseguido emitindo-se regularmente, um padrão de dígitos (padrão de sincronismo) conhecido pelo receptor, juntamente com a sequência de dados de comprimento fixo formada durante o ciclo de varredura dos canais. O padrão de sincronismo junto com a sequência de dados formam o que é chamado de QUADRO ("frame") do sinal multiplexado (Davies e Barber, 1973). Por outro lado, quando o ciclo de varredura for variável, i.é., os tempos são alocados dinamicamente, numa base estatística, aos canais efetivamente ativos durante o ciclo tem-se um sistema de MULTIPLEXAÇÃO POR DIVISÃO EM TEMPO ASSÍNCRONO (MDTA) (Doll, 1972). A fim de se identificar cada componente do canal multiplexado, geralmente, adiciona-se um endereço a cada bloco de dígitos (Davies e Barber, 1973). Em alguns casos pode ser conveniente o envio da sequência de varredura em operação junta-

mente com o sincronismo de ciclo. O sistema de multiplexação dinâmica MDTA, ao contrário do MDTS, pode operar com canais de diversas capacidades não sendo, portanto, restrito a certas taxas de informação na entrada. Mais importante ainda, é que a porção de sua capacidade alocada para um canal pode variar; dessa forma, se os canais são usados esporadicamente, a capacidade do canal multiplexado necessita ser suficiente apenas para a demanda de pico real. O sistema MDTA é, na verdade,uma forma híbrida de multiplexação e concentração, frequentemente, denominado de MULTIPLEXAÇÃO ESTATÍSTICA (Doll, 1972).

O sistema MDTS é, geralmente, mais eficiente que o MDF na utilização de um dado canal de comunicações , pois utiliza todo espectro de frequências do canal. Por exemplo, o MDTS pode operar num canal de voz, em algunas circunstâncias, a velocidade de até 9600 bits/segundo, enquanto que o MDF tem uma limitação prática na faixa dos 2.000 bits/seg no mesmo canal (Smith, 1976). O sistema MDTS pode ser usado para terminais assíncronos ("start-stop"), síncronos ou combinação destes. No caso de aplicações, exclusivamente com terminais assíncronos, a formação do quadro multiplexado através da interpolação de "bytes", ao invés da interpolação de bits, é mais conveniente por causa da compressão de banda que proporciona (Doll, 1972). Por outro lado, em aplicações onde os dados na entrada são sequências síncronas, geralmente, o MDTS forma o quadro multiplexado por interpolação de bits ignorando o formato das sequências de dados na entrada. Essa transparência ao formato de dados pode ser um requisito muito importan-

te para a incorporação de sistemas de multiplex em grandes redes de dados síncronas (Doll, 1972).

O MDTS com interpolação de bytes é menos sensível ao ruído do que com interpolação de bits porém, a ressincronização (reestabelecimento do início do quadro) é mais demorada em relação ao MDTS com interpolação de bits. Os sistemas MDTS, na prática, tem sido projetados com a filosofia de que erros aleatórios ou em "burst" causam erros nos dados propriamente ditos mas, virtualmente, nunca causam erros nos sinais de controle entre os terminais e entre os próprios MDTS's. Esse objetivo pode ser alcançado usando-se grandes redundâncias na codificação de todos sinais de controle vitais. A sincronização do quadro multiplexado, por exemplo, é crítica no desempenho do sistema e, uma maneira típica de se aumentar a confiabilidade na recepção do sincronismo de quadro é repetindo várias vezes o padrão de sincronismo. Dessa forma, a sincronização é assumida pelo receptor quando o padrão de sincronismo for detetado um determinado número de vezes. Por outro lado, a perda de sincronismo é assumida quando essa mesma condição não for detetada. Outras estatégias de proteção do sincronismo existem e, a utilização de uma em particular vai depender, principalmente, das características dos canais envolvidos no processo de multiplexação.

Do ponto de vista da flexibilidade em configurações de linha multiponto, o sistema MDTS, em comparação com MDF, se apresenta em desvantagem. A inserção de um canal, por exemplo, em qualquer ponto exige, praticamente, um siste-

ma completo de MDTS e um par de modem's para o novo canal. Também, a complexidade na coordenação da operação das múltiplas sequências síncronas de dados em um único sinal torna-o menos flexível em relação ao MDF. Por outro lado, em redes cujos terminais estejam relativamente concentrados torna-se vantajosa sua utilização sob o ponto de vista da manutenção e confiabilidade.

3.4 - MULTIPLEXAÇÃO POR DIVISÃO EM CÓDIGOS (MDC)

A busca de uma sempre melhor utilização dos canais de comunicação em termos de eficiência e confiabilidade, influenciou determinantemente a evolução dos sistemas de multiplexação digital. Por exemplo, a natureza intermitente das comunicações geradas pelos terminais em redes de comunicações de dados deu margem ao desenvolvimento do sistema MDTA a partir dos princípios do MDT. Por outro lado, a evolução tecnológica da eletrônica digital permitiu a competição de novas técnicas de multiplexação não-convencionais (por exemplo, o MDS de Harmuth, (1973)). Estima-se que na próxima década (Karp, 1976), com a proliferação de novas redes de comunicações digitais aliada a utilização dos cada vez mais competitivos micro processadores, grande parte dos sistemas MDT sejam "inteligentes", i.é., contenham capacidade de controle de erros automático e de alocação dinâmica dos canais.

A inatividade dos canais na entrada de um multiplex pode ser aproveitada, basicamente, de duas maneiras. Uma seria a maior utilização do canal de saída pelos canais efetivamente ativos que é o caso do MDTA e dos concentradores em geral. Outra maneira seria a utilização da capacidade do canal não aproveitada, para se aumentar a confiabilidade na transmissão da informação. Nesse último sentido, Gordon e Barrett (1971) desenvolveram um sistema de multiplex não-convencional pertencente à classe dos sistemas MDC, que troca ca

pacidade de canal por capacidade de controle de erros de uma maneira adaptativa. Uma limitação prática do sistema de Gordon e Barrett é o número de canais permitidos pela reversibilidade da transformação usada no processo de multiplexação (Rocha Neto, 1975). O Multiplex Digital por Divisão em Códigos Cíclicos (MDDCC) proposto no Capítulo IV foi desenvolvido com a filosofia de incorporação de capacidade de controle de erros automático ao sistema de multiplexação e, também, de aproveitamento da inatividade dos canais para se aumentar essa capacidade. O MDDCC, basicamente, multiplexa os canais síncronos no tempo por interpolação de bits e, a cada bloco de bits, assim formado, adiciona ciclicamente dígitos redundantes, formando uma palavra-código a ser transmitida. O uso de redundância variável permite o aproveitamento da inatividade de um certo número de canais para um substancial aumento da capacidade de correção de erros aleatórios. O número de canais multiplexados por esse sistema é definido pelo comprimento dos códigos cíclicos com redundância variável usados no processo e, em princípio, esse número não é limitado.

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 Tel (083) 321-7222-R 355
58.100 - Campina Grande - Paraíba

CAPÍTULO IV

MULTIPLEX DIGITAL POR DIVISÃO EM CÓDIGOS CÍCLICOS (MDDCC)

4.1 - INTRODUÇÃO

O Multiplex Digital por Divisão em Códigos Cíclicos (MDDCC) proposto nesta tese é apresentado neste Capítulo. O MDDCC, um sistema de multiplexação pertencente à classe dos sistemas não-convencionais MDC (ver Capítulo III), pode ser visto como um sistema MDTS por interpolação de bits onde a cada bloco de bits, assim formado, adiciona-se um processamento adaptativo ao número de canais ativos que inclue um reposicionamento dos bits no bloco e a inserção de dígitos redundantes, i.é., a codificação dos bits de informação correspondente aos canais ativos na entrada do sistema. O número de canais possíveis de serem multiplexados com esse sistema de

pende exclusivamente dos códigos cíclicos binários, de comprimento fixo e redundância variável, utilizados, i.é., do número de dígitos de informação k do código cíclico (n,k) usado para codificar os bits de informação quando todos canais estiverem ativos. O MDDCC foi implementado na prática para 11 canais. Essa escolha levou em conta, além da simplicidade de implementação, o fato de que os sistemas MDT da próxima década tendem a ser menores (que os atuais) com um número de canais da ordem de 10 (Karp, 1976). Na implementação do MDDCC usou-se os códigos cíclicos binários de comprimento 15 com redundância variável, apresentados no Capítulo II, para se codificar, de uma maneira adaptativa, a informação correspondente aos canais ativos. A correspondência entre a atividade dos canais na entrada do MDDCC e o código cíclico usado na codificação da informação correspondente aos bits dos canais multiplexados é mostrada na Tabela 1. Observa-se nessa tabela que a correspondência entre o número de canais ativos e a capacida-

Nº DE CANAIS ATIVOS	(n,k) $G_i(X)$ i=1,2,3,4,5	t
8,9,10 ou 11	(15,11)	1
6 ou 7	(15,7)	2
3, 4 ou 5	(15,5)	3
2	(15,2)	4
1	(15,1)	7

Tabela 1

de de correção de erros aleatórios t associada ao código cíclico não é linear. A escolha dos códigos cíclicos de comprimento 15 foi feita de modo que a redundância adicionada (devido a inatividade dos canais) contribuisse para um aumento efetivo na capacidade de correção de erros aleatórios do sistema. Dessa forma, códigos cíclicos binários de comprimento 15, tais como, (15,10), (15,9) e (15,8) (Peterson, 1972), por exemplo, não foram utilizados pois possuem capacidade de correção de erros aleatórios idêntica ao código cíclico (15,11). Baseado em argumentação semelhante, também, não foram usados os códigos cíclicos (15,6), (15,4) e (15,3) (Peterson, 1972).

O MDDCC foi implementado com circuitos digitais TTL ("transistor-transistor logic") de pequena e média integração disponíveis durante o desenvolvimento deste trabalho de tese. Dessa forma, a disponibilidade de componentes, em geral, influenciou, significativamente, no desenvolvimento do sistema (ver seção 4.4). O sistema é composto basicamente por dois grandes blocos: o transmissor e o receptor do MDDCC (Figs. 4.1 e 4.2, respectivamente). O transmissor (Conversor Paralelo-Série) multiplexa no tempo por interpolação de bits os canais na entrada do MDDCC formando um bloco de 11 bits. Esse bloco de 11 bits é, então, processado pelo transmissor (Compressor de Dados), com ajuda da informação sobre a atividade dos canais, para formar a palavra-código de comprimento 15 (Codificador Cíclico Adaptativo) a ser enviada. É função, também, do transmissor o envio da informação de atividade a cada quadro multiplexado, para um correto endereçamento dos bits na

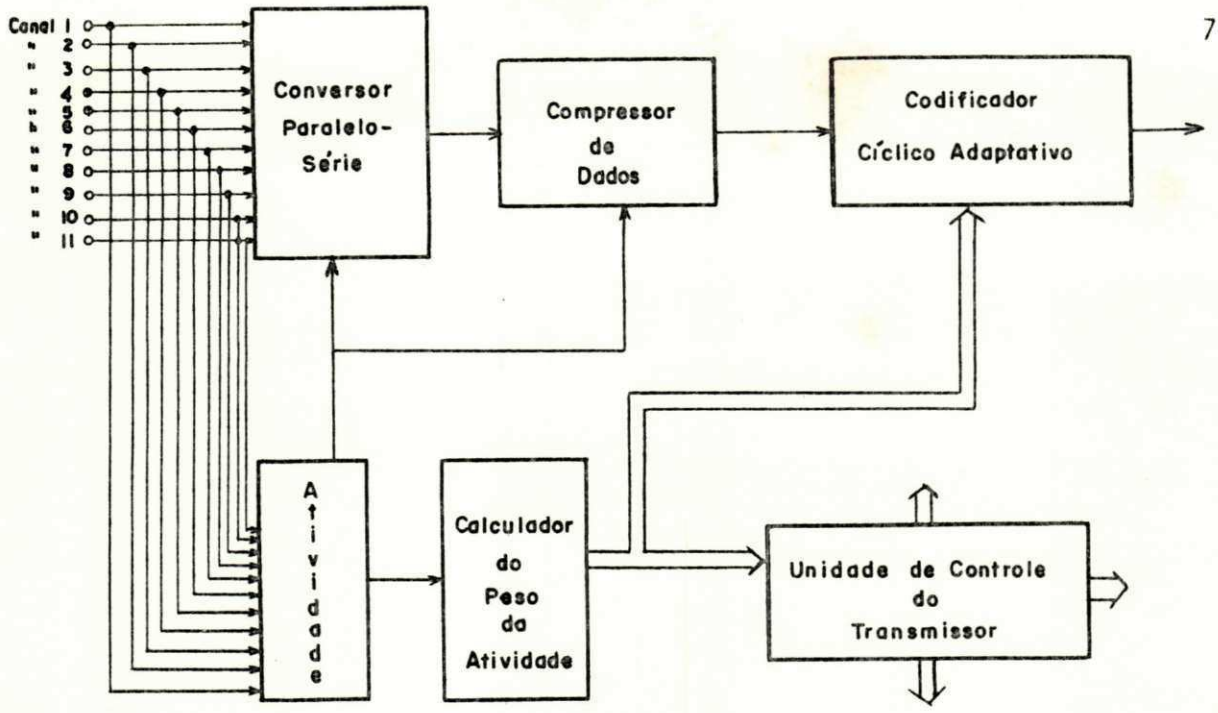


Figura 4.1 Transmissor do MDDCC

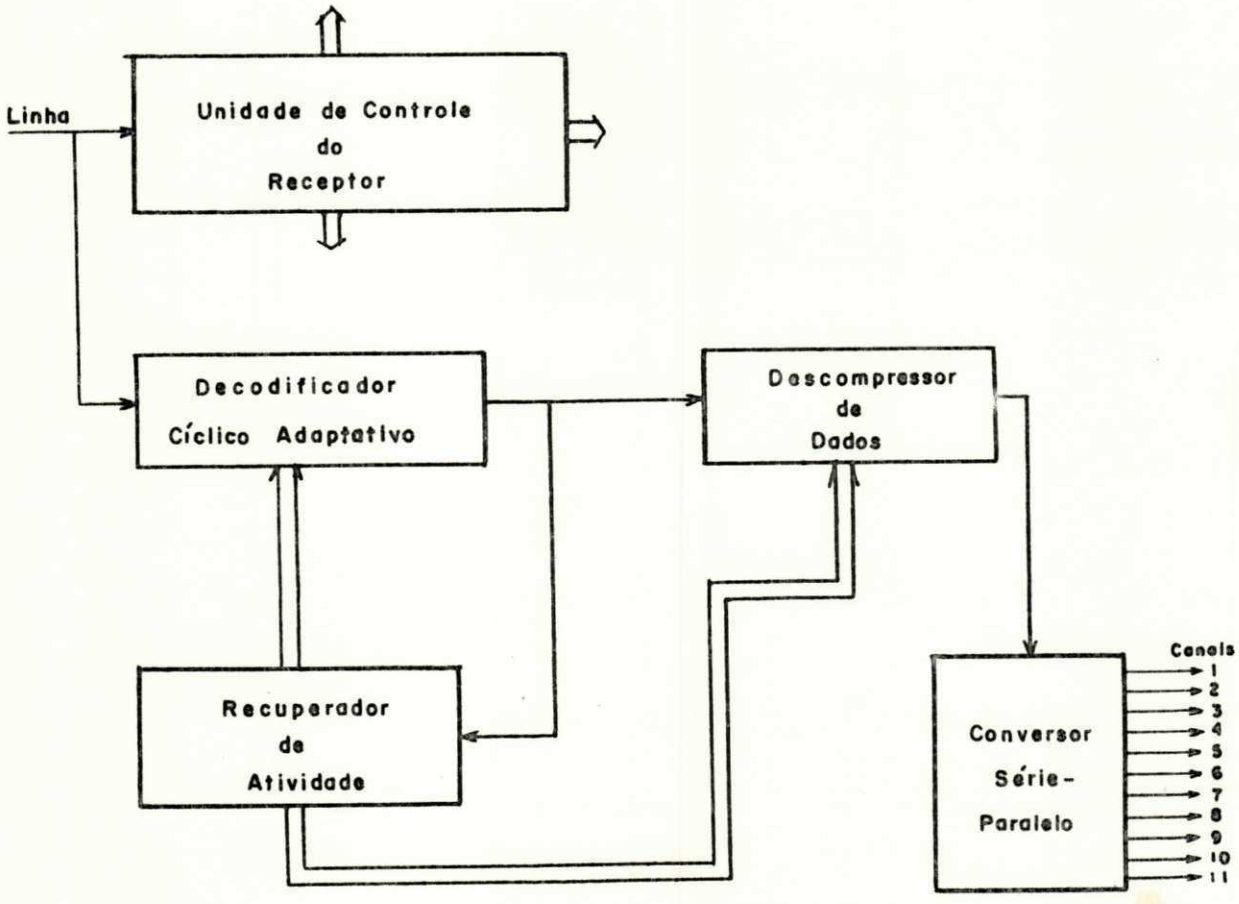


Figura 4.2 Receptor do MDDCC

recepção. O sincronismo de quadro, assim como, uma maior proteção da informação de atividade contra o ruído, são feitos através da repetição da informação de atividade codificada. A atividade dos canais foi simulada através de 11 chaves do tipo "on-off" onde um "1" lógico (+Vcc) conectado a chave diz ao sistema que o canal correspondente aquela chave, em particular, está ativo. Por outro lado, um "0" lógico (terra) diz que o canal está inativo. Na Figura 4.3 é ilustrada a simulação do padrão de atividade. Observe que o padrão de atividade é essencial na formação dos k bits de informação de cada palavra-código, i.é., na compressão dos bits, correspondentes aos canais efetivamente ativos, que compõem o bloco de bits multiplexados no tempo pelo Conversor Paralelo-Série. Isso significa dizer que o MDDCC precisa saber quais canais na sua entrada estão ativos. Por outro lado, observe que a escolha do código cíclico, portanto, do codificador cíclico utilizado, depende, exclusivamente, do número de canais ativos, não importando quais sejam os canais efetivamente ativos. Dessa forma, o cálculo do peso do padrão de atividade (Calculador de Peso da Atividade), i.é., do número de 1's no padrão de atividade,fornece ao MDDCC a informação suficiente para a escolha do código cíclico a ser utilizado em cada quadro multiplexado. Os dados correspondentes aos canais na entrada do MDDCC foram simulados através de um gerador de sequência binária pseudo-aleatória (Golomb,1969) de comprimento 2043 (Figura 4.4). A Unidade de Controle do Transmissor tem a função de gerar a base de tempo ("clock") do sistema assim como os sinais de controle

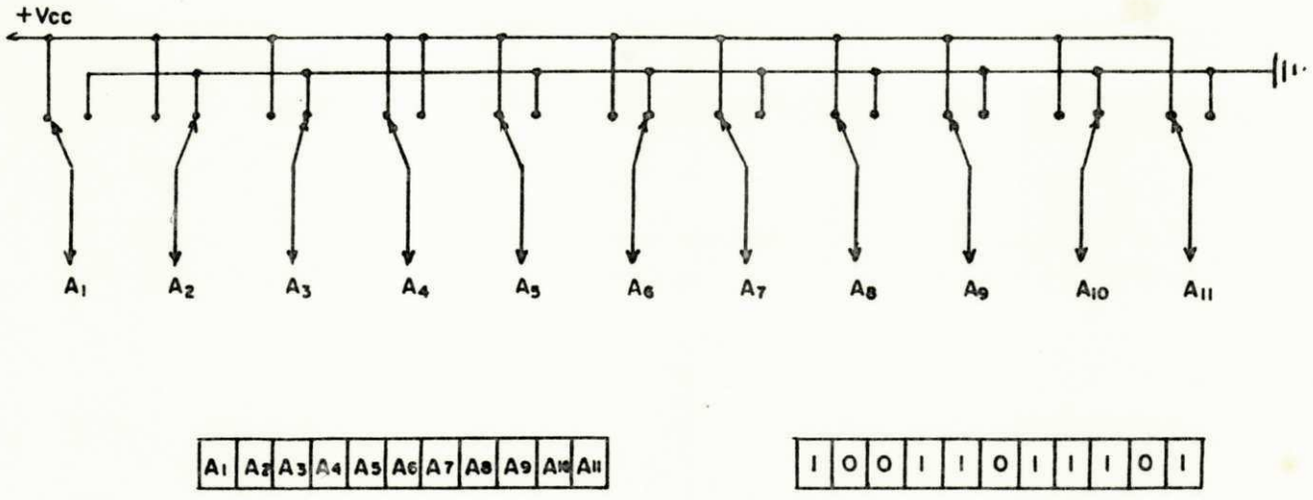


Figura 4.3 Padrão de Atividade

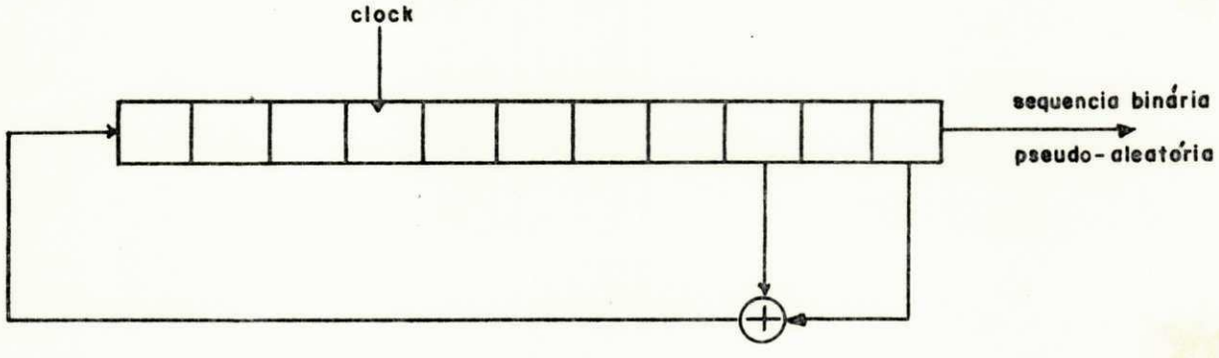


Figura 4.4 Simulação dos dados com sequencia binária pseudo-aleatória de comprimento 2043 (registrador de deslocamento com 11 estágios)

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 - Tel (083) 321-7222-R 353
58.100 - Campina Grande - Paraíba

necessários ao processamento da informação. O receptor do MDDCC, por sua vez, com reconhecimento do "clock" e do sincronismo de quadro (Unidade de Controle do Receptor), tem a função de recuperar a informação de atividade (Recuperador de Atividade) e, de posse dessa informação, proceder a decodificação das palavras-código que compõem o quadro multiplexado (Decodificador Cíclico Adaptativo), alocando os bits de informação, correspondentes aos canais multiplexados, para os respectivos destinatários (Decompressor de Dados e Conversor Série-Paralelo).

4.2 - TRANSMISSOR DO MDDCC

4.2.1 - FORMAÇÃO DO QUADRO MULTIPLEXADO

O quadro multiplexado do MDDCC é formado por 67 palavras-código binárias de comprimento 15 perfazendo um total de 1005 bits onde as tres primeiras correspondem a informação de atividade (11 bits) codificada pelo código cíclico (15,11) e as 64 palavras-código restantes, correspondem à informação dos canais ativos codificada pelo código cíclico associado a atividade em questão. A formação do quadro multiplexado é ilustrada na Figura 4.5. A repetição da atividade

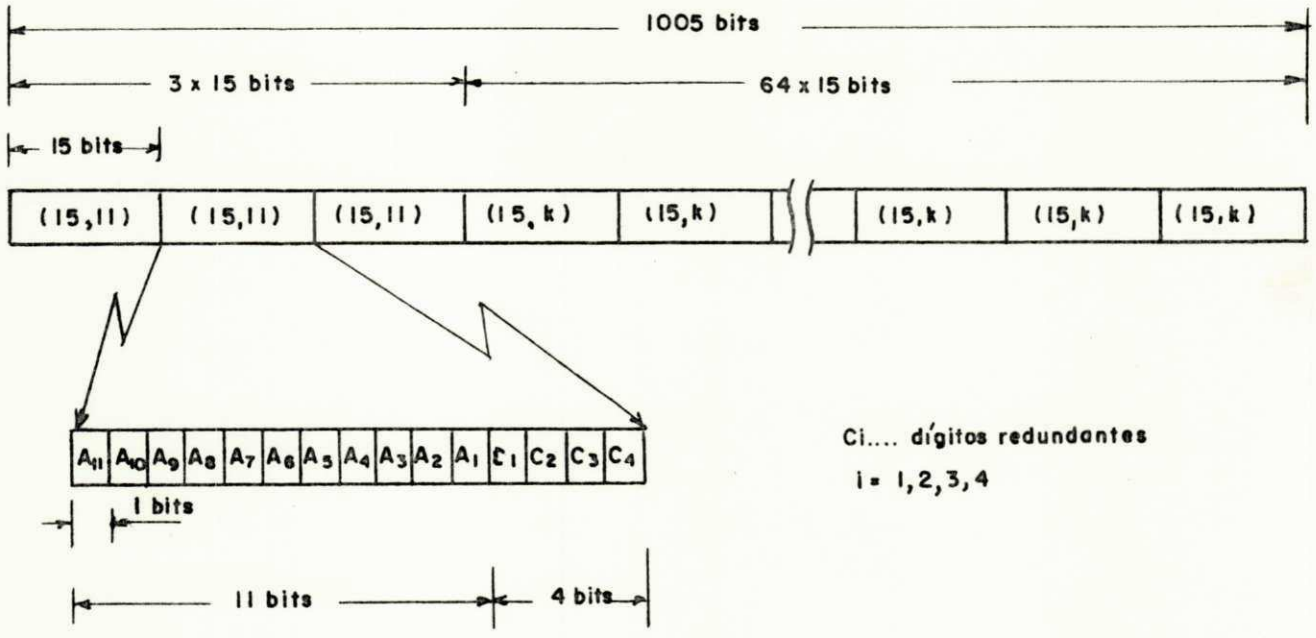


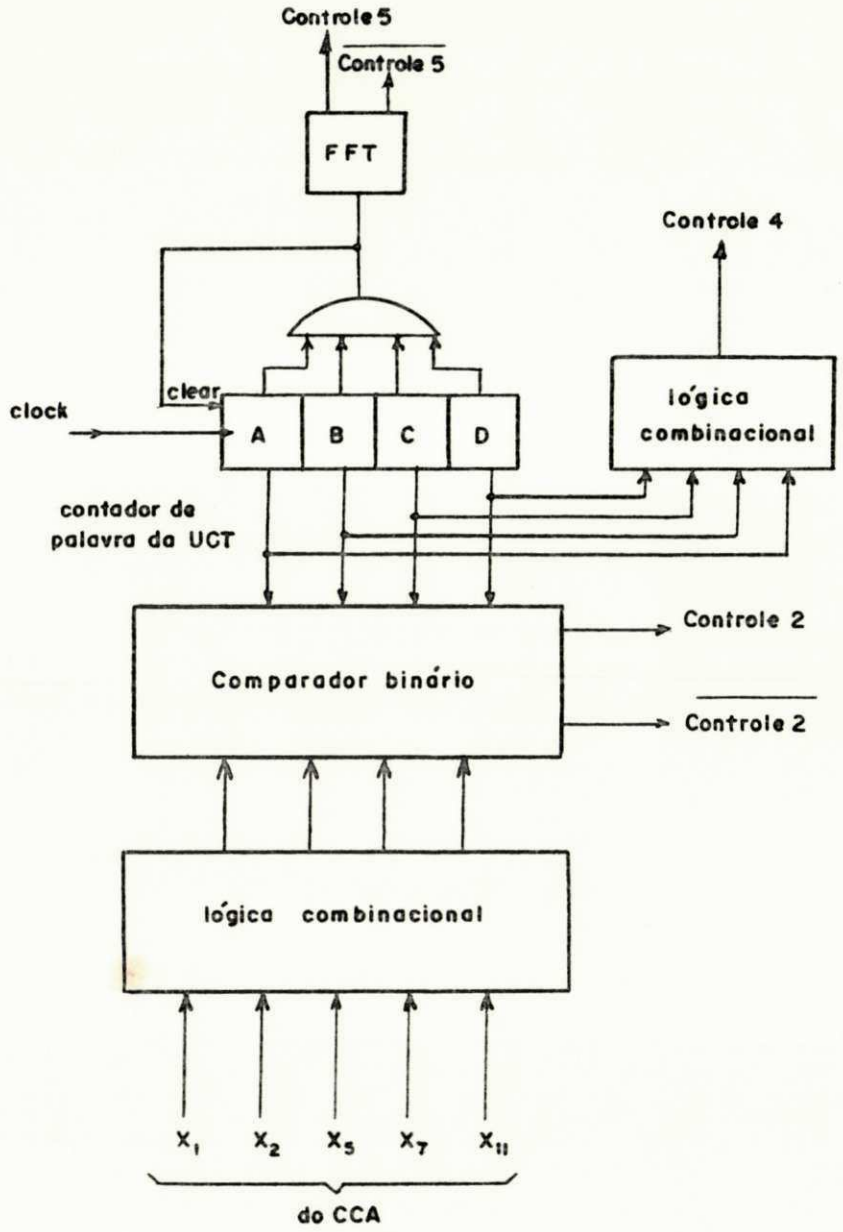
Fig. 4.5 - Quadro do MDDCC

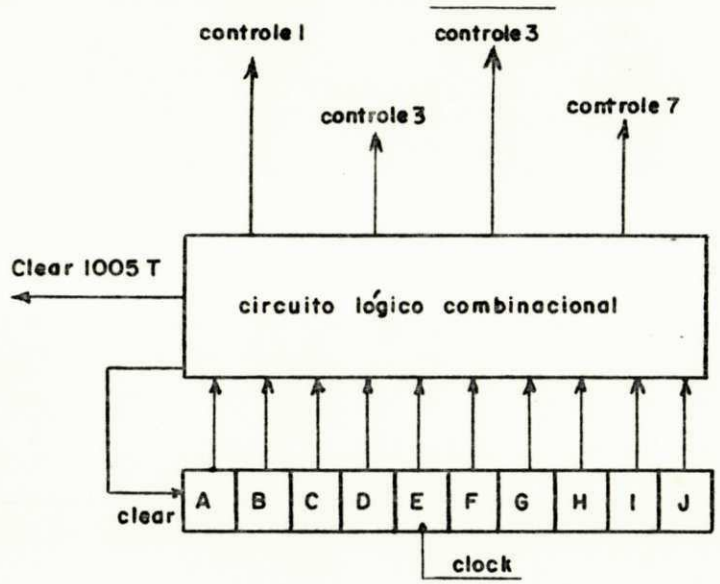
codificada é utilizada para se aumentar a confiabilidade de sua recepção, assim como, para a recuperação do sincronismo de quadro pelo receptor (ver seção 4.3.1). O comprimento do quadro foi escolhido de modo que a taxa líquida de informação transmitida ("net data throughput"), i.é., a eficiência de utilização do canal de saída do MDDCC, fosse determinada basicamente pela eficiência do código cíclico usado para codificar a informação multiplexada e, que os canais na entrada do MDDCC tivessem suas atividades reconhecidas pelo MDDCC com um atraso máximo de 64 bits. Por exemplo,no caso de todos canais ativos, a eficiência de utilização do canal de transmissão é aproximadamente igual a 70% ($k/n = 11/15 \cong 0,73$). Uma interface dos canais com o transmissor do MDDCC do tipo "hold-and-forward" (Davies e Barber, 1973) exigiria por exemplo, armazenadores de no máximo 64 bits para cada fonte de informação.

4.2.2 - UNIDADE DE CONTROLE DO TRANSMISSOR (UCT)

A Unidade de Controle do Transmissor (UCT) do MDDCC, mostrado na Figura 4.6, tem a função de gerar a base de tempo ("clock") do sistema que define a velocidade em bits/segundo do sinal multiplexado e,os sinais de controle necessários as funções do transmissor (conversão paralelo-série dos dados, leitura e inserção da atividade no quadro multiplexado, compressão dos dados no tempo e codificação) na formação do quadro multiplexado. O diagrama no tempo dos diversos sinais derivados da UCT é apresentado na Figura 4.7. As funções dos sinais de controle são descritos a seguir.

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 Tel (083) 321 7222-R 355
58.100 - Campina Grande - Paraíba





Contador de quadro da UCT

FIGURA 4.6 Unidade de Controle do Transmissor

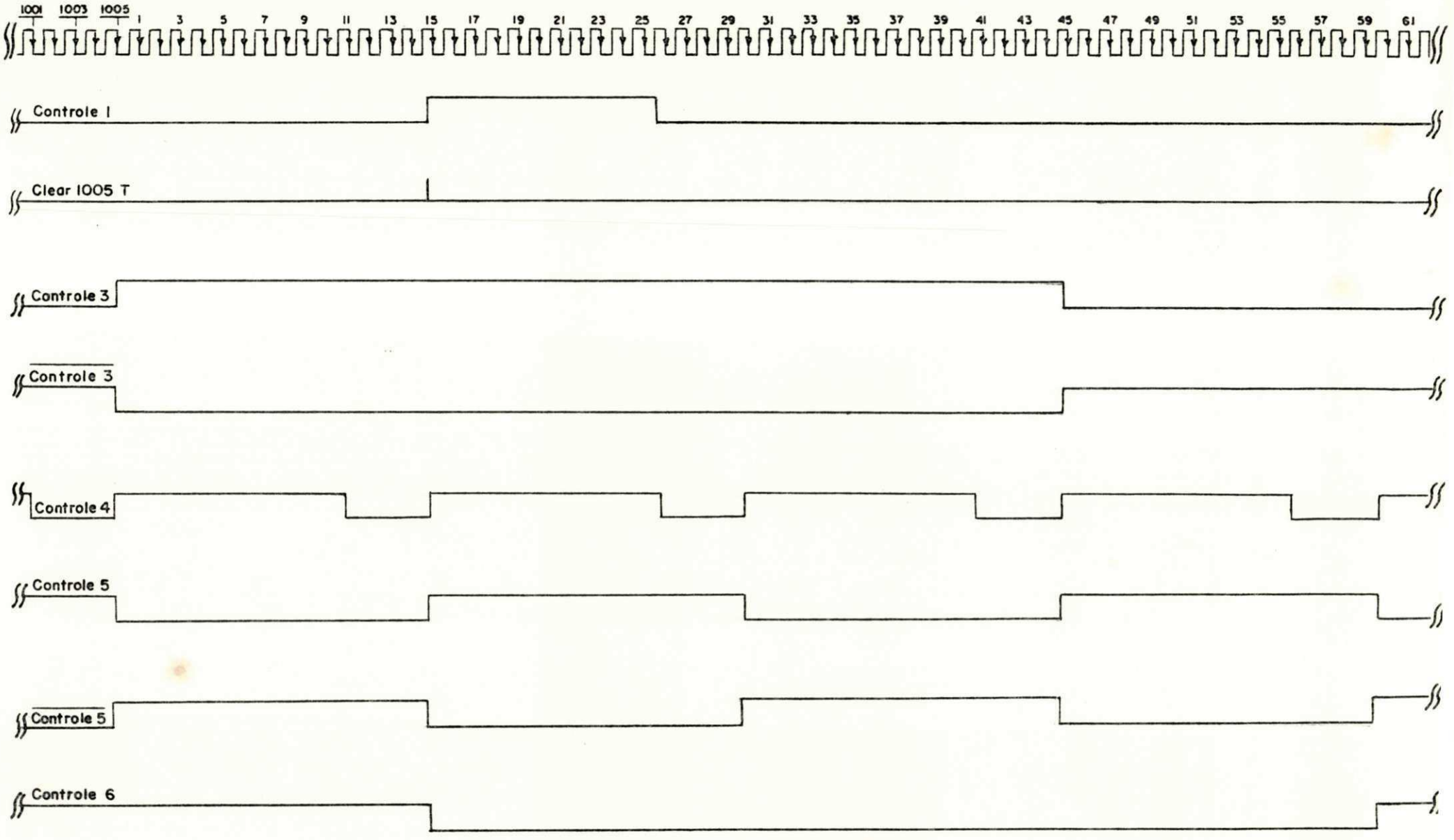


Figura 4.7 DIAGRAMA NO TEMPO DOS SINAIS DA UCT

Controle 1 - fornece a cada ciclo de operação (quadro) do MDDCC o tempo necessário ao cálculo sequencial do peso da atividade (ver seção 4.2.3).

Controle 2 - é o sinal, adaptativo ao peso da atividade, que controla o tempo de recepção e envio dos k bits de informação pelo Codificador Cíclico Adaptativo (seção 4.2.6).

$\overline{\text{Controle 2}}$ - é o sinal, complementar do sinal Controle 2, que controla o tempo de inserção e envio dos bits de paridade pelo Codificador Cíclico Adaptativo. Na Figura 4.8 é ilustrado o sinal Controle 2 e $\overline{\text{Controle 2}}$ (acima e abaixo, respectivamente) para os cinco códigos utilizados pelo MDDCC.

Controle 3 - fornece o tempo necessário (a cada quadro) para a inserção do padrão de atividade, repetido por tres vezes, no quadro multiplexado (ver seção 4.2.4).

$\overline{\text{Controle 3}}$ - é o sinal, complementar ao Controle 3, que fornece o tempo necessário, a cada ciclo de operação do MDDCC, para o processamento da informação multiplexada dos canais.

Controle 4 - é o sinal que temporiza os blocos de 11 bits, sejam os de informação multiplexada dos canais ou os correspondentes ao padrão de atividade, em blocos de 15 posições no tempo necessários as pa

lavras-códigos. Corresponde, também, ao sinal Controle 2 quando o código cíclico utilizado é o (15,11). (Ver Seção 4.2.4 e 4.2.5).

Controle 5 e $\overline{\text{Controle 5}}$ - controlam a alternância dos armazenadores de duplo deslocamento no Compressor de Dados (ver seção 4.2.5).

Controle 6 - estabelece o tempo de codificação da atividade repetida por tres vezes. A defasagem em relação ao sinal $\overline{\text{Controle 3}}$ é devido ao atraso de 15 bits no processo de compressão de dados. (ver seção 4.2.6).

Clear 1005 T - inicializa a cada quadro o contador de peso da atividade. (seção 4.2.3).

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 Tel (083) 321 7222-R 355
58.100 - Campina Grande - Paraíba

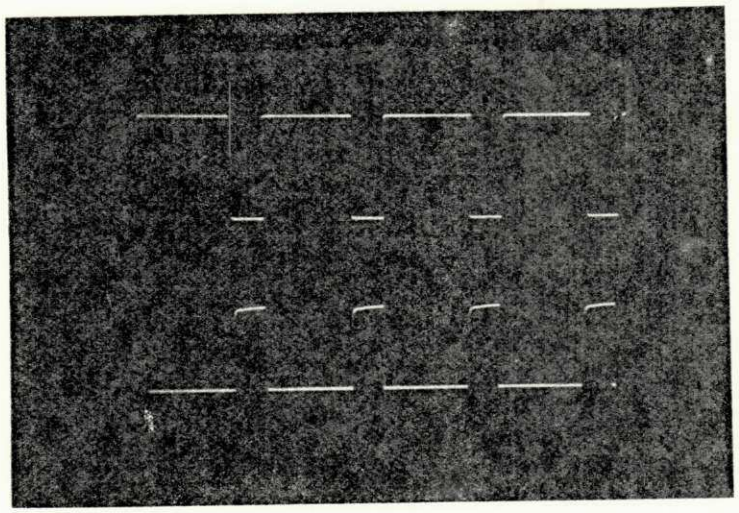


Fig. 4.8 (a) - Controle 2 e $\overline{\text{Controle 2}}$ para o código (15,11)

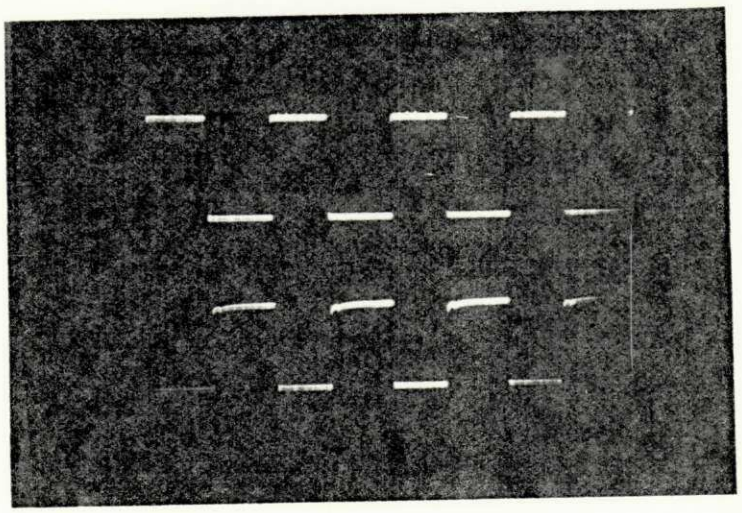


Fig. 4.8 (b) - Controle 2 e $\overline{\text{Controle 2}}$ para o código (15,7)

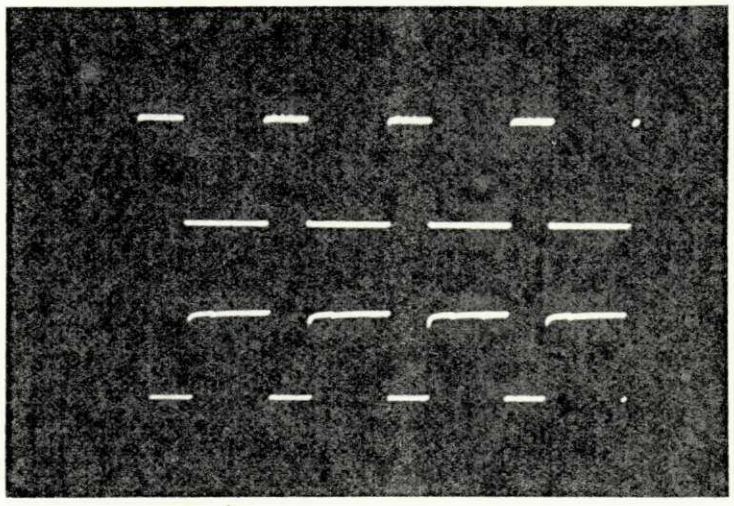


Fig. 4.8 (c) - Controle 2 e $\overline{\text{Controle 2}}$ para o Código (15,5)

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprigio Veloso, 882 Tel (083) 321 7222-R 355
58 100 - Campina Grande - Paraíba

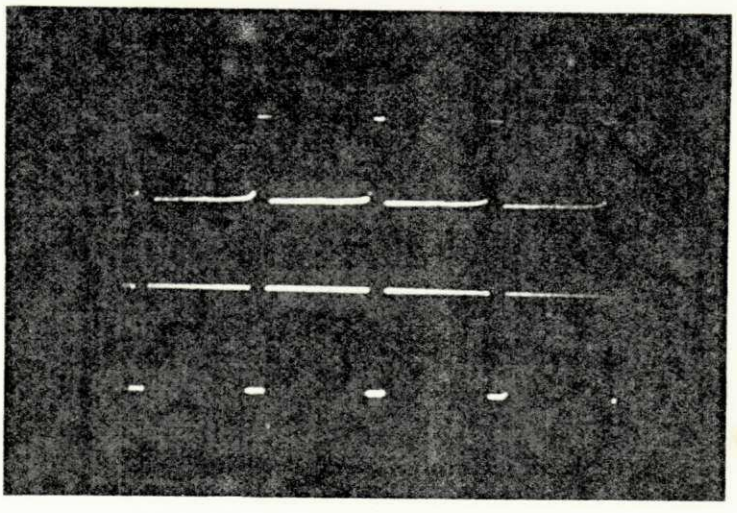


Fig. 4.8 (d) - Controle 2 e $\overline{\text{Controle 2}}$ para o Código (15,2)

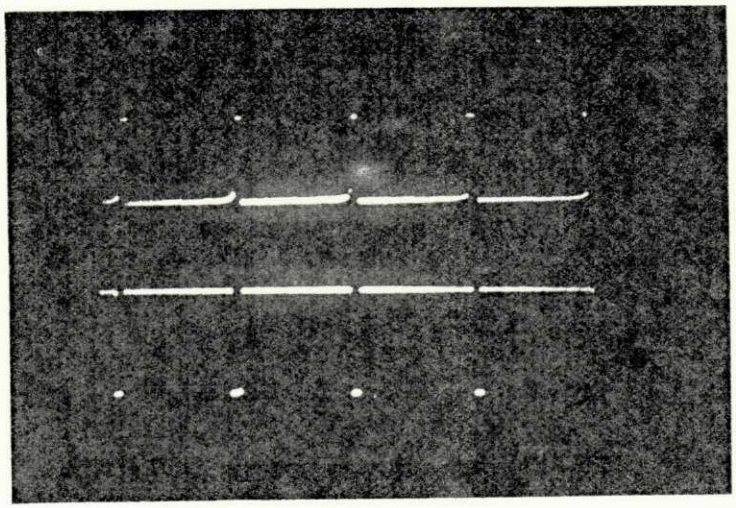


Fig. 4.8 (e) - Controle 2 e $\overline{\text{Controle 2}}$ para o código (15.1).

4.2.3 - INFORMAÇÃO DE ATIVIDADE E CALCULADOR DE PESO DA ATIVIDADE

A informação de atividade, como já foi dito anteriormente, é simulada em paralelo por 11 chaves do tipo "on-off". A conversão paralelo-série do padrão de atividade, necessária para o processo de compressão de dados no CD e para o cálculo do peso da atividade, é efetuada por um circuito multiplexador, sob o comando do "clock" do sistema e do sinal Controle 4, conforme mostrado na Figura 4.9. O cálculo do peso da atividade, i.é., do número de canais ativos, é feito, sequencialmente, pelo contador de 4 bits, sob o comando do sinal Controle 1. A informação do peso da atividade é usada em paralelo na escolha das conexões pelo CCA e na derivação do sinal Controle 2 pela UCT. Ao final de cada quadro multiplexado o contador de 4 bits é zerado ("clear") para nova contagem da atividade.

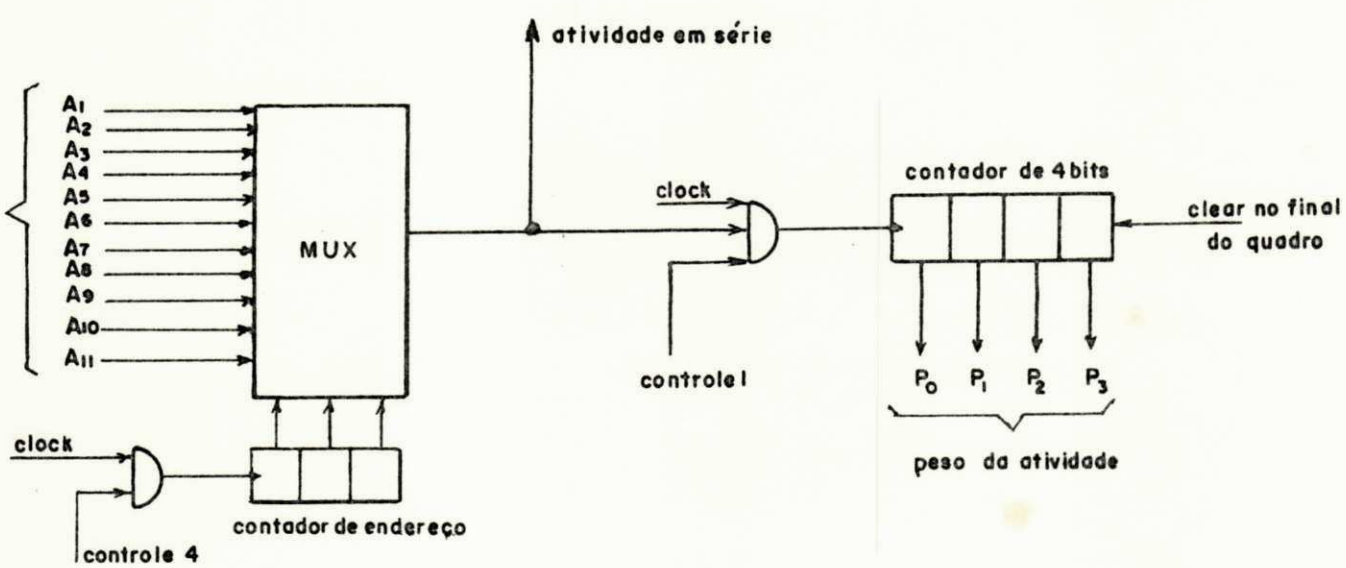


Fig. 4.9 - Conversão paralelo-série do padrão de atividade e Calculador de Peso da Atividade

Na Figura 4.10 é ilustrada a saída do circuito multiplexador para a atividade 1 0 0 1 1 0 0 1 1 0 1 , enquanto que na Figura 4.11 é apresentada a entrada do contador de 4 bits para essa atividade.

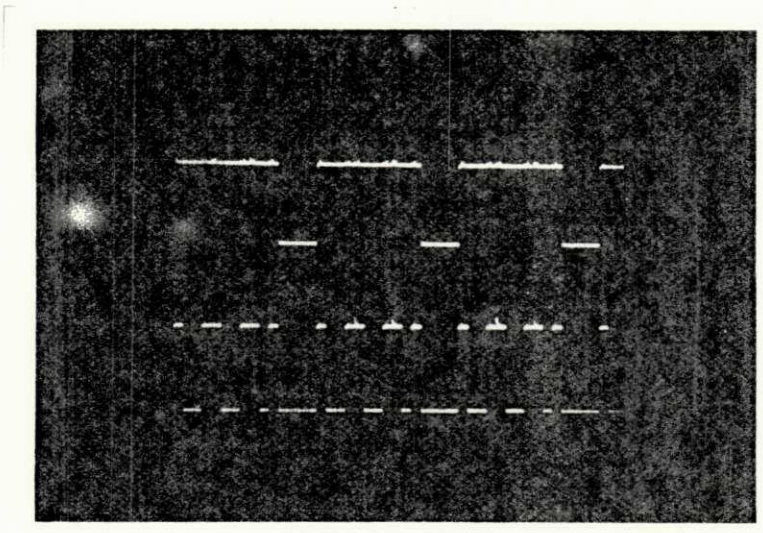


Fig. 4.10 - Atividade 1 0 0 1 1 0 0 1 1 0 1 em série

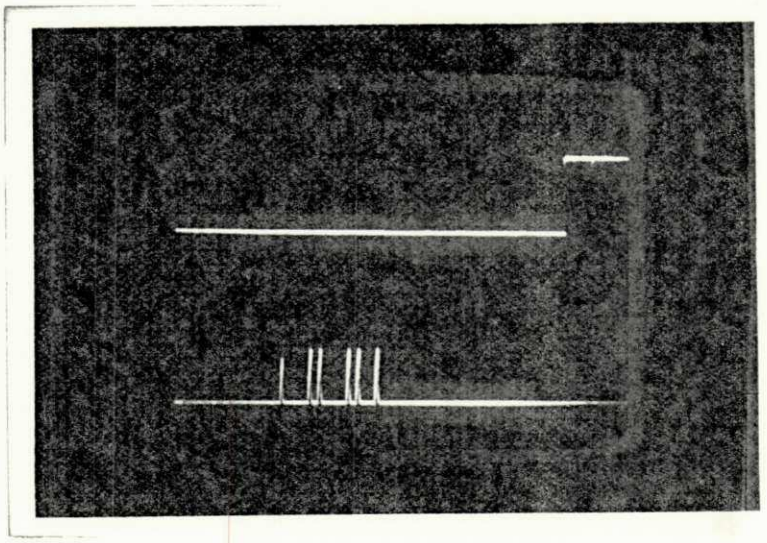


Fig. 4.11 - Cálculo do peso da atividade
1 0 0 1 1 0 0 1 1 0 1

4.2.4 - CONVERSOR PARALELO-SÉRIE

A função do Conversor Paralelo-Série (CPS) do MDDCC é, basicamente, a de multiplexar, por interpolação de bits, os canais na entrada do MDDCC. O bloco de 11 bits multiplexado pelo conversor é enviado em série para o Compressor de Dados (CD) a fim de que seja efetuada a compressão dos bits significativos no bloco de acordo com a atividade dos canais na entrada. A implementação prática do CPS, foi bastante simplificada, assumindo-se os dados em série simulados pelo gerador de sequência binária pseudo-aleatória (Figura 4.4). Dessa forma, a função do CPS se constituiu em enviar para o CD ora blocos de 11 bits temporizados pelo sinal Controle 4, ora o padrão de atividade em série também temporizado pelo sinal Controle 4. O envio da atividade é comandado pelo sinal Controle 3 enquanto que os dados são comandados pelo seu complementar $\overline{\text{Controle 3}}$. Na Figura 4.12 é ilustrada a simulação do CPS utilizada na prática.

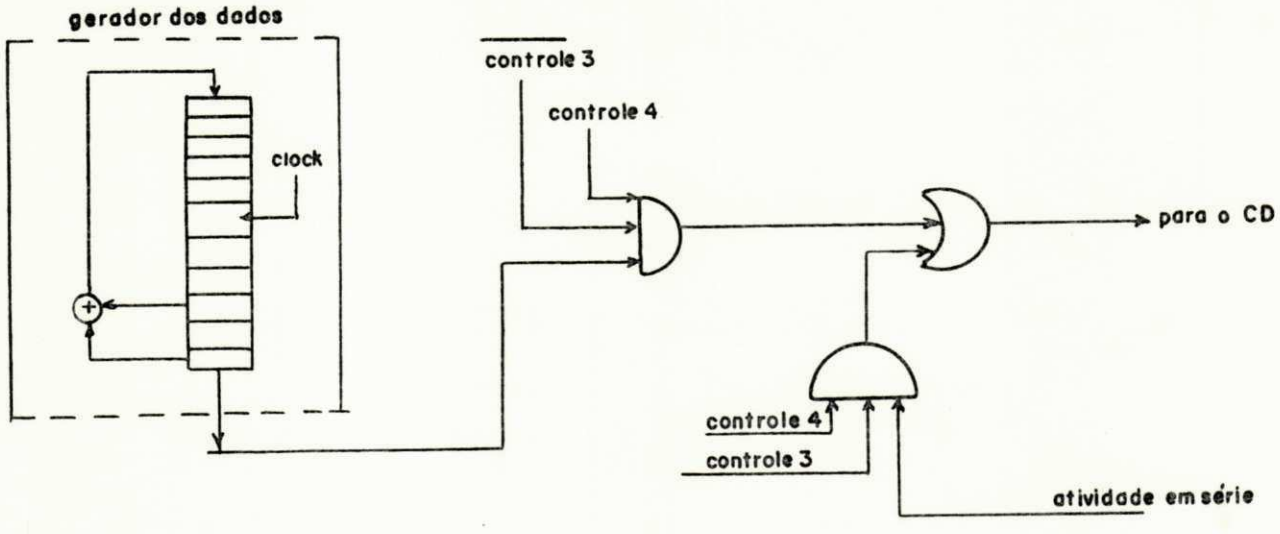


Fig. 4.12 - Simulação do CPS e dos dados

4.2.5 - COMPRESSOR DE DADOS (CD)

O Compressor de Dados (CD) do transmissor do MDDCC forma, juntamente com o Codificador Cíclico Adaptativo (CCA) (seção 4.2.6), o que poderia ser chamado de "parte inteligente" do MDDCC. O CD, apresentado na Figura 4.13, é constituido, basicamente, por dois armazenadores de duplo deslocamento (à direita e à esquerda) de 11 bits. Seu funcionamento pode ser descrito como se segue. O bloco de bits de comprimento 11 (temporizado pelo sinal Controle 4) resultado da "conversão paralelo-série" dos canais na entrada ($\overline{\text{Controle 3}}$) é deslocado à direita nos armazenadores de acordo com a atividade dos canais, i.é., com pulsos de "clock" somente nas posições (no tempo) dos canais ativos. Assim, os bits correspondentes aos canais ativos são comprimidos na porção mais à esquerda dos armazenadores. Após o deslocamento completo (15 pulsos ou não-pulsos de "clock") do bloco à direita nos armazenadores, procede-se o deslocamento à esquerda (agora, de acordo com o "clock" do sistema) dos bits comprimidos e das posições "vazias" (zeros) do bloco. O CCA sob comando do sinal Controle 2 e $\overline{\text{Controle 2}}$ aproveita as posições "esvaziadas" no processo de compressão para inserir mais dígitos redundantes além dos quatro usuais do código (15,11). No caso dos blocos de 11 bits correspondentes ao padrão de atividade repetido (Controle 3) não se faz necessária a compressão e, o deslocamento inicial à direita é feito com base no "clock" do transmissor. O uso de dois armazenadores em paralelo se faz

necessário por causa do processo sequencial de compressão. Dessa forma, enquanto os dados estão sendo comprimidos num armazenador, o outro está enviando para o CCA os dados já comprimidos. Os sinais Controle 5 e $\overline{\text{Controle 5}}$ controlam essa alternância dos armazenadores. Observe que o processo de compressão em um dos armazenadores pode ser visto como um empilhamento, de bits significativos, do tipo LIFO ("last in-first out) (Davies e Barber (1979)). Assim os últimos bits a serem deslocados (à direita) para o armazenador serão os primeiros a serem recebidos (deslocamento à esquerda) pelo CCA. Isso justifica a ordem do padrão de atividade codificado ilustrado na formação do quadro multiplexado da Figura 4.5. No caso em que o número de bits comprimidos (nº de canais ativos) for menor que o número de bits de informação do código k o CCA recebe posições "vazias" (zeros) em que não aloca redundâncias e que na recepção devem ser desprezados.

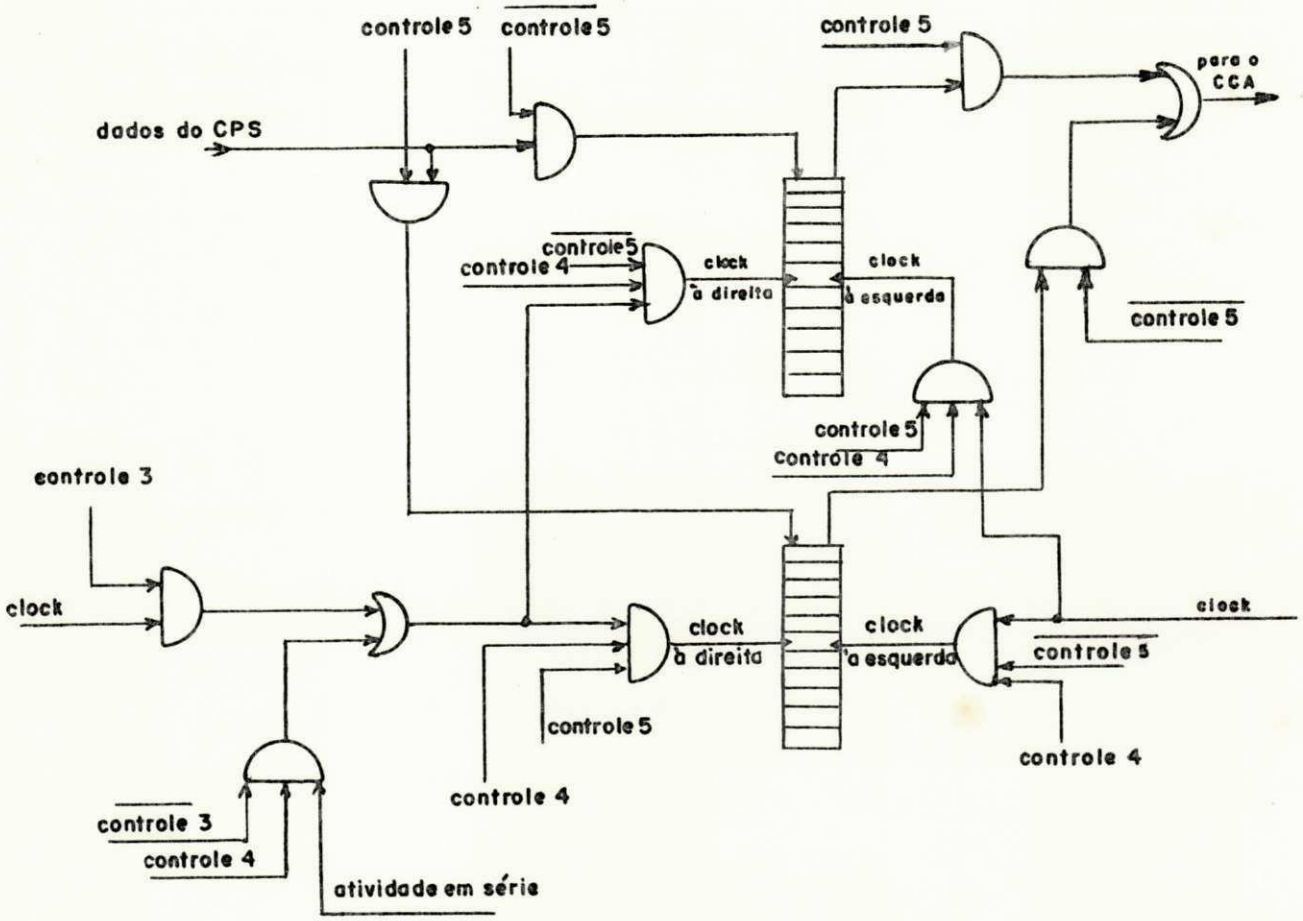


Fig. 4.13 - Compressor de Dados

4.2.6 - CODIFICADOR CÍCLICO ADAPTATIVO

O Codificador Cíclico Adaptativo (CCA) tem a função de formar as palavras-códigos que compõem o quadro

multiplexado do MDDCC. O diagrama geral do CCA é mostrado na Figura 4.14. O tipo de codificador cíclico utilizado é o de k estágios (seção 2.3.4) onde os bits de informação são deslocados, simultaneamente, para o canal de transmissão e para o armazenador de deslocamento com realimentação formar os bits de paridade. Dessa forma, após o envio para o canal dos k bits de informação o sinal Controle 2, adaptativo ao número de canais ativos, comanda o envio dos (n-k) bits de paridade para o canal formando a palavra-código a ser enviada. As conexões de realimentação nesse tipo de codificador correspondem aos coeficientes dos polinômios de paridade H(X). Na tabela 2 abaixo são listados os polinômios de paridade correspondentes aos códigos cíclicos binários utilizados pelo MDDCC. O circuito lógico combinacional (CLC), cuja tabela de verdade é apresenta-

(n,k)	H(X)
(15,11)	$H_5(X) = 1 + X + X^2 + X^3 + X^5 + X^7 + X^8 + X^{11}$
(15,7)	$H_4(X) = 1 + X^4 + X^6 + X^7$
(15,5)	$H_3(X) = 1 + X + X^3 + X^5$
(15,2)	$H_2(X) = 1 + X + X^2$
(15,1)	$H_1(X) = 1 + X$

Tabela 2

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 Tel (083) 321 7222-R 355
58 100 - Campina Grande - Paraíba

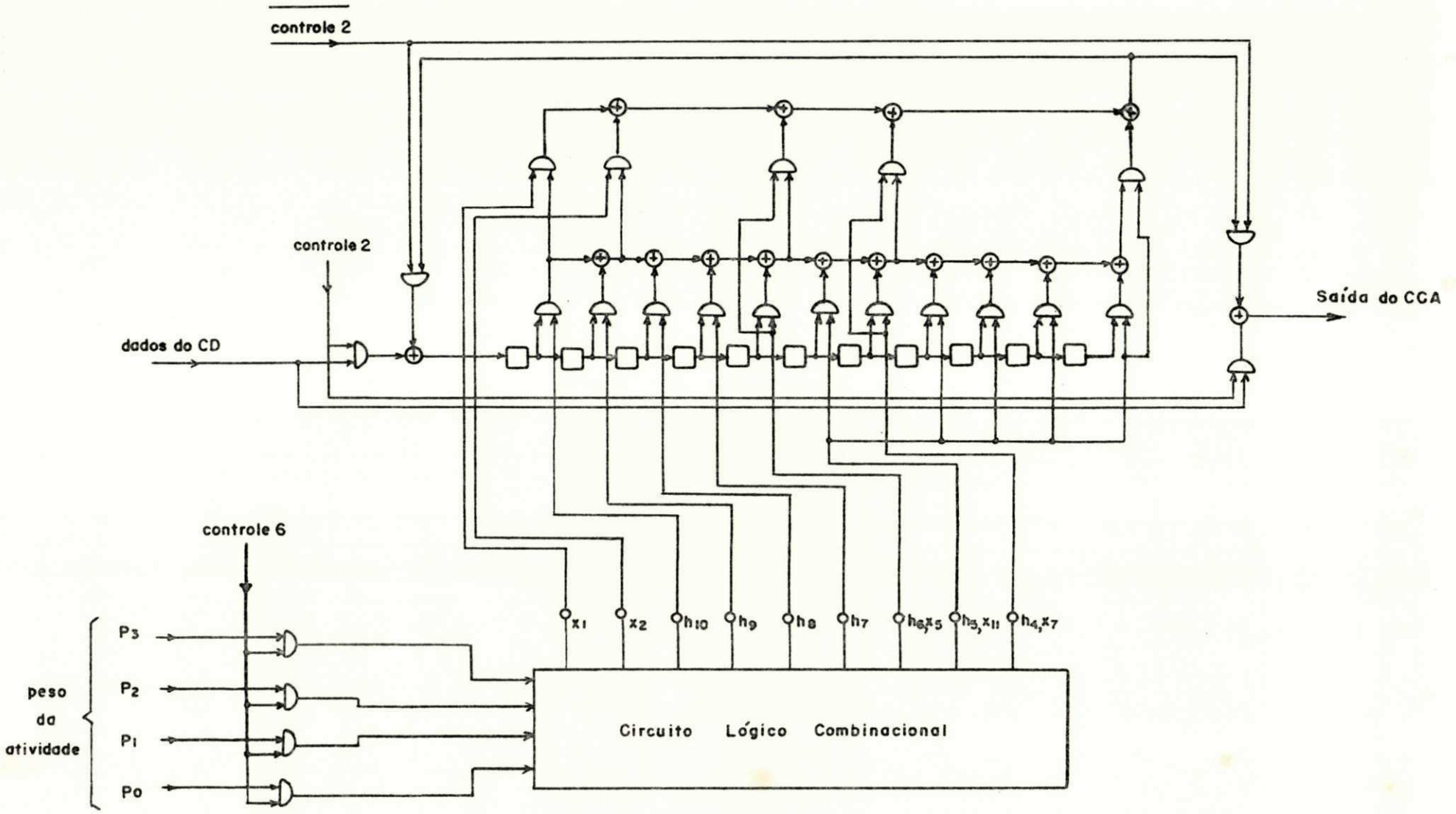


Figura 4.14 Codificador Cíclico Adaptativo (CCA)

da na tabela 3, estabelece as conexões de acordo com o peso da atividade. No caso particular da codificação da própria atividade o CLC, comandado pelo sinal Controle 6, estabelece as conexões para o código (15,11). O sinal Controle 6 corresponde ao sinal $\overline{\text{Controle 3}}$ atrasado de 15 clocks . Essa defasagem se deve ao fato de que o processo de compressão de dados atrasa em 15 bits a entrada da informação de atividade no codificador.

PESO DA ATIVIDADE				$X_k \leftrightarrow (n,k)$					C O N E X Õ E S										
P_3	P_2	P_1	P_0	X_1	X_2	X_5	X_7	X_{11}	h_{10}	h_9	h_8	h_7	h_6	h_5	h_4	h_3	h_2	h_1	h_0
0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0
0	1	0	1	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	0
0	1	1	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
1	0	0	1	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
1	0	1	0	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
1	0	1	1	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1

Tabela 3

4.3 - RECEPTOR DO MDDCC

4.3.1 - UNIDADE DE CONTROLE DO RECEPTOR (UCR)

A Unidade de Controle do Receptor (UCR) do MDDCC tem a função de recuperar, do sinal multiplexado recebido, o sincronismo de bit ("clock") e de quadro e, a partir disso, gerar os sinais de controle que, juntamente com a informação de atividade recuperada, permitem a decodificação e o correto endereçamento dos bits de informação aos destinatários . Por questões de simplicidade, usou-se um canal independente para envio do "clock", ao invés da implementação prática da recuperação de "clock". No entanto, na Figura 4.15 é sugerido em diagrama de blocos, um sistema com PLL ("phase-lock-loop") que permite a recuperação do "clock" a partir da periodicidade do quadro multiplexado de 1005 bits (Vilar França, 1978) . O diagrama da UCR é mostrado na Figura 4.16.

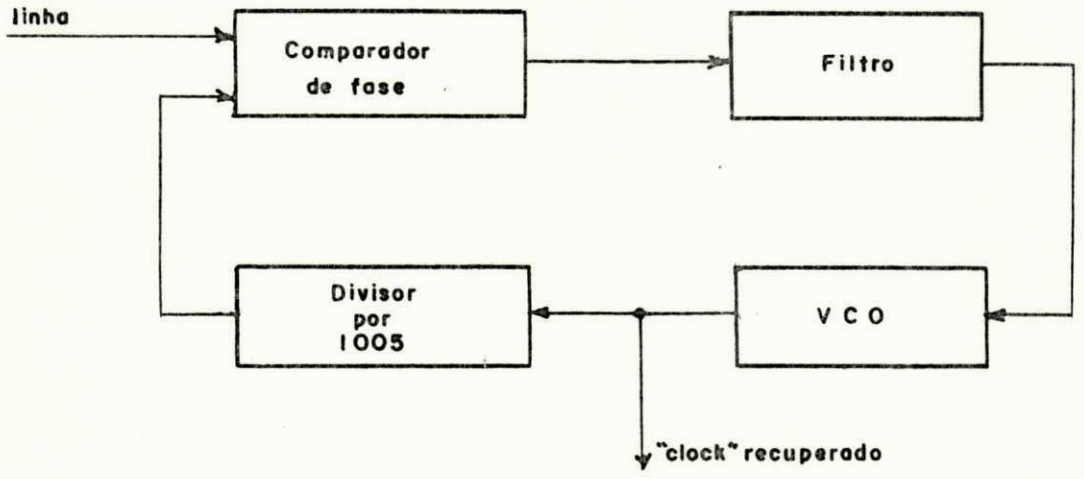


Fig. 4.15 - Recuperação do "clock" usando PLL

A recuperação do sincronismo de quadro é feita pela procura, a cada quadro multiplexado, do padrão de atividade codificado repetido por tres vezes. Essa procura é efetuada verificando-se a identidade bit a bit de tres quaisquer palavras-código consecutivas recebidas. É garantida, a menos de erros introduzidos pelo ruído, a existência dessa situação em cada quadro multiplexado através da repetição por tres vezes do padrão de atividade codificado e, por outro lado, assegura-se uma probabilidade baixa de simulação da situação de tres palavras-código consecutivas iguais por parte das palavras-código correspondentes à informação dos canais multiplexados. Quando a identidade bit a bit de tres palavras-código consecutivas for alcançada o contador de sincronismo de quadro CSQ (Figura 4.16) terá contado 15 pulsos de "clock" consecutivos então, assume-se o sincronismo e é emitido o sinal SINC 1005. O sinal SINC 1005 controla a inicialização do contador de controle de quadro (CCQ) que a cada ciclo conta de 1 até 1005 emitindo o sinal CONT 1005. A perda de sincronismo é detetada pela defasagem desses dois sinais. A estratégia de recuperação de sincronismo utilizada no MDDCC foi a de considerar o sincronismo de quadro perdido quando por tres vezes consecutivas os sinais SINC 1005 e CONT 1005 não aparecem em fase. Essa estratégia, diminui a probabilidade de que se considere uma perda de sincronismo quando ocorrer erros no padrão de atividade codificado ou quando houver simulação, por parte das palavras-código de informação propriamente dita, da situação de tres palavras-código iguais consecutivas. Por ou-

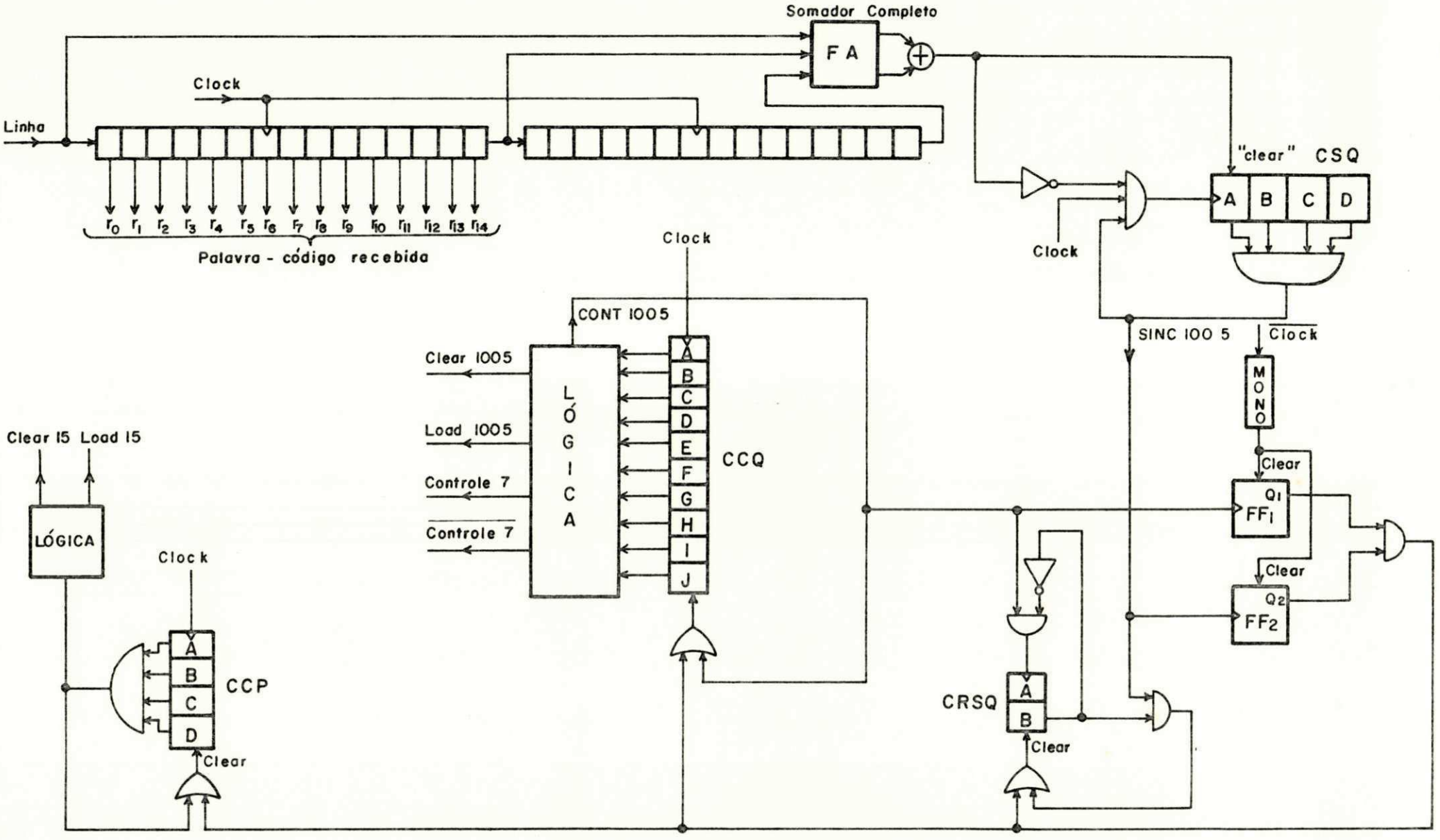


Figura 4.16 Unidade de controle do receptor

UFPb/BIBLIOTECA/PRAI

tro lado, essa estratégia ocasiona um atraso (3 quadros) na recuperação do sincronismo quando houver uma perda real de sincronismo. Essa perda de sincronismo por tres vezes consecutivas é detetada pelo conjunto de "Flip-Flops" (FF1 e FF2) e do contador de recuperação de sincronismo (CRSQ) de dois bits. Uma vez que o CRSQ tenha identificado por tres vezes consecutivas a defasagem dos sinais SINC 1005 e CONT 1005, é aberto, imediatamente após, o circuito onde o sinal SINC 1005 procede a inicialização do contador de controle de quadro (CCQ) pondo em fase os dois sinais. Os outros sinais de controle da UCR, necessários no processo de decodificação (Decodificador Cíclico Adaptativo) e alocação dos bits aos destinatários correspondentes (Recuperador de Atividade, Decompressor de Dados e Conversor Série-Paralelo), são derivados do CCQ e do contador de controle de palavra-código (CCP) através de circuitos lógicos. Os sinais de controle, ilustrados na Figura 4.17, são descritos a seguir.

Controle 7 - fornece o tempo de decodificação e de recuperação da atividade.

$\overline{\text{Controle 7}}$ - fornece o tempo para a decodificação e processamento da informação correspondente aos canais multiplexados no transmissor.

Clear 15 - fornece os pulsos de apagamento dos registradores de deslocamento para uma posterior leitura em paralelo das palavras-código (ou blocos de 15 bits).

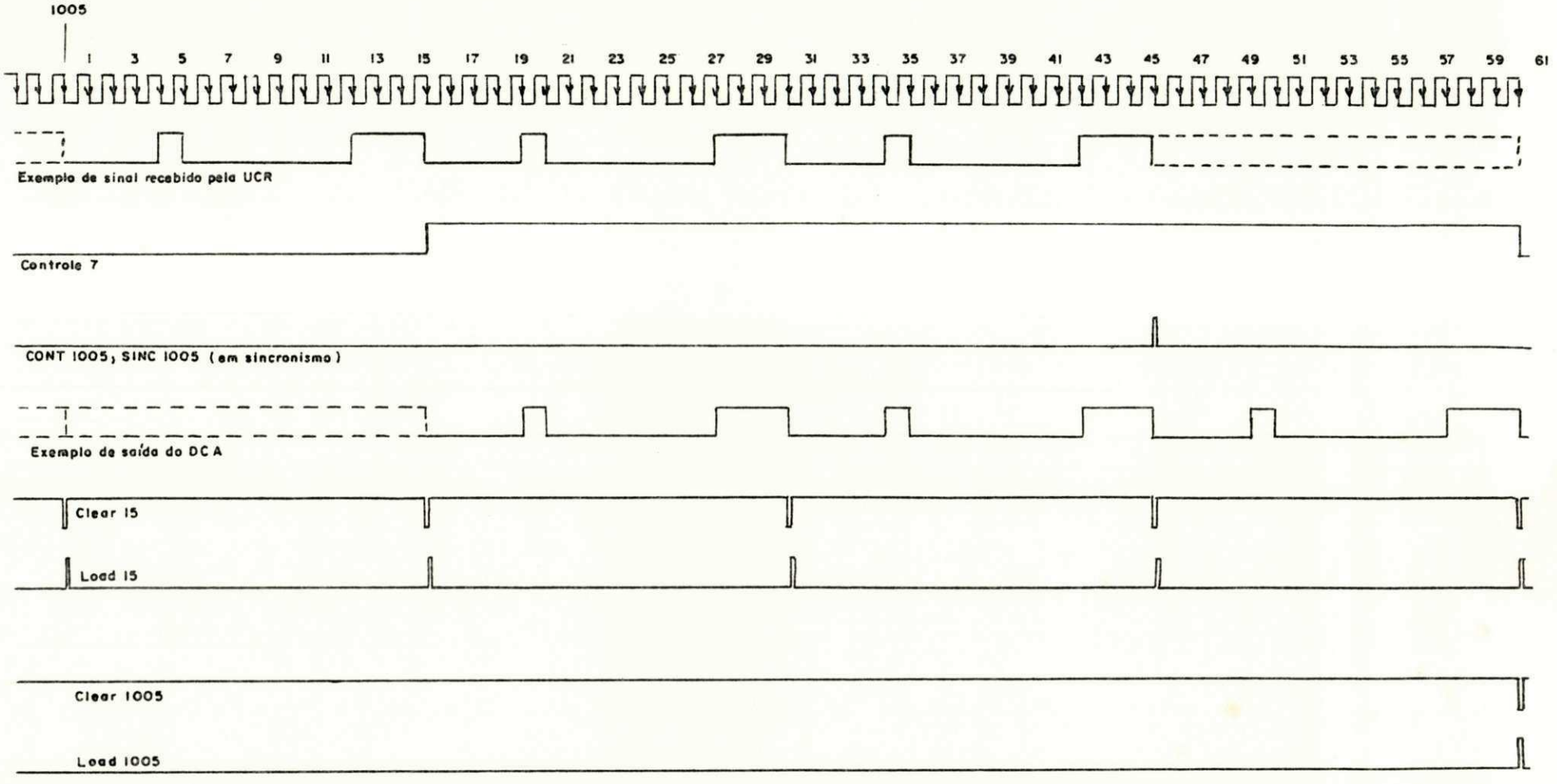


Figura 4.17 Diagrama no tempo dos sinais derivados da UCR

Load 15 - fornece os pulsos de leitura (carregamento) em paralelo de palavras-código (ou blocos de 15 bits) aos registradores de deslocamento; é sempre precedido pelo sinal Clear 15.

Clear 1005 - fornece o pulso de apagamento do registrador de armazenamento da atividade no Descompressor de Dados (seção 4.3.4).

Load 1005 - fornece o pulso de leitura em paralelo da nova atividade, a cada quadro multiplexado, pelo registrador de armazenamento da atividade no Descompressor de Dados (seção 4.3.4).

4.3.2 - RECUPERADOR DE ATIVIDADE

O Recuperador de Atividade, ilustrado na figura 4.18, tem a função de recuperar a atividade dos canais multiplexados a cada quadro recebido. Como a informação de atividade é codificada no transmissor para controle de erros, procede-se primeiro a decodificação no Decodificador Cíclico Adaptativo (DCA) (seção 4.3.3) das três palavras-código correspondentes à atividade para depois determinar-se a atividade, do quadro em questão, por maioria bit a bit dos tres padrões de atividade decodificados. Esse procedimento dá uma maior confiabilidade na recepção da informação de atividade. A decodificação das tres palavras-código correspondentes a informação de atividade é efetuada pelo DCA sob o comando do si

nal $\overline{\text{Controle 7}}$ que, de maneira semelhante ao sinal $\overline{\text{Controle 3}}$ na codificação da atividade (seção 4.2.6), estabelece o decodificador cíclico para o código (15,11). Uma vez determinada a atividade, esta tem seu peso calculado combinacionalmente pelo DCA (seção 4.3.3). O DCA de posse do peso da atividade estabelece então, as conexões necessárias para a decodificação das 64 palavras-código restantes. A atividade recuperada é essencial, também, para o processo de descompressão de dados efetuado pelo Descompressor de Dados (DD). Assim, uma vez determinada a atividade do quadro em questão, ela é transferida em paralelo para o registrador de atividade do DD, sob o comando dos sinais Clear 1005 e Load 1005. Na figura 4.19 é ilustrado o sinal que entra no recuperador de atividade após a decodificação no DCA.

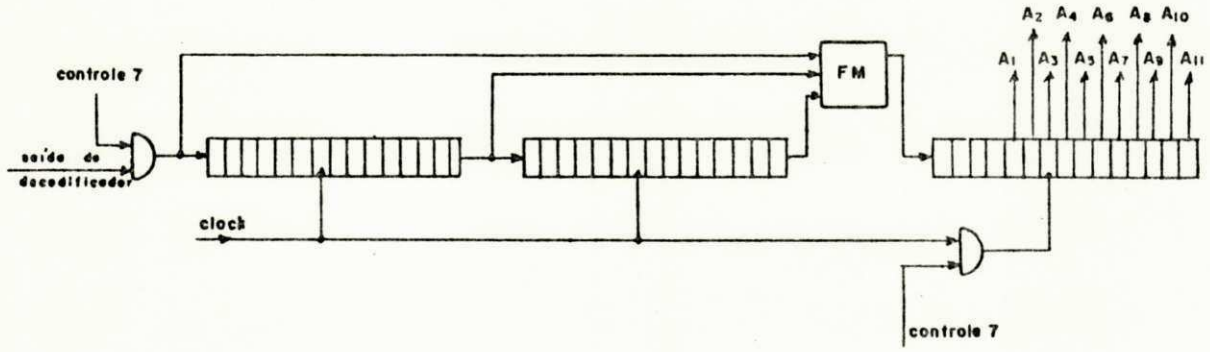


Fig. 4.18 - Recuperador de Atividade

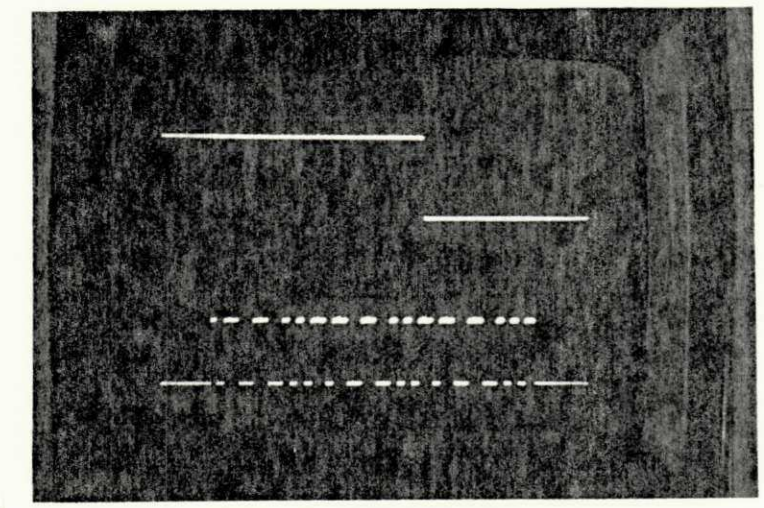


Fig. 4.19 - Sinal na entrada do primeiro registrador do Recuperador de Atividade para padrão de atividade 1 0 0 1 1 0 0 1 1 0 1

4.3.3 - DECODIFICADOR CÍCLICO ADAPTATIVO (DCA)

O Decodificador Cíclico Adaptativo (DCA) é o bloco do receptor do MDDCC que efetua a decodificação das palavras-código recebidas de acordo com a informação de atividade recuperada. Como já foi dito anteriormente (seção 4.3.2), no caso da decodificação das palavras-código que contêm a informação de atividade, o Controle 7 (ver Figura 4.20) estabelece o decodificador para o código cíclico (15,11). A decodifi

cação utilizada pelo DCA é a Decodificação por Função de Maioria Tipo II (seção 2.3.8). A escolha desse tipo de decodificação em particular, levou em conta, além da minimização do número de componentes, o fato de que a Decodificação por Função de Maioria, fornece, em alguns casos, capacidade extra de correção de erros aleatórios, aos códigos cíclicos (seção 2.3.8). Os códigos cíclicos binários (15,1), (15,2) e (15,7) são ortogonalizáveis em um passo enquanto que os códigos (15,11) e (15,5) são ortogonalizáveis em dois passos. O conjunto de equações de paridade referentes a cada código são apresentados a seguir.

Código Cíclico (15,11) ($H_5(X) = 1 + X + X^2 + X^3 + X^5 + X^7 + X^8 + X^{11}$)

1º Passo:

$$E_1^{(1)} = \{r_7, r_{12}, r_{13}, r_{14}\}$$

$$A_1^{(1)} = r_0+r_4+r_7+r_8+r_{10}+r_{12}+r_{13}+r_{14}$$

$$A_2^{(1)} = r_3+r_6+r_7+r_9+r_{11}+r_{12}+r_{13}+r_{14}$$

$$E_2^{(1)} = \{r_5, r_8, r_9, r_{14}\}$$

$$A_1^{(2)} = r_0+r_1+r_5+r_8+r_9+r_{11}+r_{13}+r_{14}$$

$$A_2^{(2)} = r_2+r_3+r_5+r_8+r_9+r_{10}+r_{14}$$

2º Passo:

$$E_1^{(2)} = \{r_{14}\}$$

$$A_1^{(1)} = r_7+r_{12}+r_{13}+r_{14}$$

$$A_2^{(1)} = r_2+r_8+r_9+r_{14}$$

Código Cíclico (15,5) ($H_3(X) = 1 + X + X^3 + X^5$)

1º Passo:

$E_1^{(1)} = \{r_{12}, r_4\}$

$$A_1^{(1)} = r_0 + r_{10} + r_{12} + r_{14}$$
$$A_2^{(1)} = r_3 + r_{11} + r_{12} + r_{14}$$
$$A_3^{(1)} = r_7 + r_{13} + r_{12} + r_{14}$$
$$A_4^{(1)} = r_1 + r_2 + r_{12} + r_{14}$$
$$A_5^{(1)} = r_4 + r_8 + r_{12} + r_{14}$$
$$A_6^{(1)} = r_6 + r_9 + r_{12} + r_{14}$$

$E_2^{(1)} = \{r_{13}, r_{14}\}$

$$A_1^{(2)} = r_4 + 4_{10} + r_{13} + r_{14}$$
$$A_2^{(2)} = r_9 + r_{11} + r_{13} + r_{14}$$
$$A_3^{(2)} = r_7 + r_{12} + r_{13} + r_{14}$$
$$A_4^{(2)} = r_0 + r_8 + r_{13} + r_{14}$$
$$A_5^{(2)} = r_1 + r_5 + r_{13} + r_{14}$$
$$A_6^{(2)} = r_3 + r_6 + r_{13} + r_{14}$$

2º Passo:

$E_1^{(2)} = \{r_{14}\}$

$$A_1^{(1)} = r_{12} + r_{14}$$
$$A_2^{(1)} = r_{13} + r_{14}$$

Código Cíclico (15,7) ($H_4(X) = 1 + X^4 + X^6 + X^7$)

$E_1 = \{ r_{14} \}$

$A_1 = r_3 + r_{11} + r_{12} + r_{14}$

$A_2 = r_7 + r_8 + r_{10} + r_{14}$

$A_3 = r_1 + r_5 + r_{13} + r_{14}$

$A_4 = r_0 + r_2 + r_6 + r_{14}$

Código Cíclico (15,2) ($H_2(X) = 1 + X + X^2$)

$E_1 = \{ r_{14} \}$

$A_1 = r_2 + r_{14}$

$A_2 = r_5 + r_{14}$

$A_3 = r_8 + r_{14}$

$A_4 = r_{11} + r_{14}$

$A_5 = r_0 + r_1 + r_{14}$

$A_6 = r_3 + r_4 + r_{14}$

$A_7 = r_6 + r_7 + r_{14}$

$A_8 = r_9 + r_{10} + r_{14}$

Código Cíclico (15,1) ($H_1(X) = 1 + X$)

$E_1 = \{ r_{14} \}$

$A_1 = r_0 + r_{14}$

$A_2 = r_1 + r_{14}$

$A_3 = r_2 + r_{14}$

$A_4 = r_3 + r_{14}$

$A_5 = r_4 + r_{14}$

$A_6 = r_5 + r_{14}$

$A_7 = r_6 + r_{14}$

$A_8 = r_7 + r_{14}$

$A_9 = r_8 + r_{14}$

$A_{10} = r_9 + r_{14}$

$A_{11} = r_{10} + r_{14}$

$A_{12} = r_{11} + r_{14}$

$A_{13} = r_{12} + r_{14}$

$A_{14} = r_{13} + r_{14}$

O diagrama do DCA é apresentado na Figura 4.20. O funcionamento do DCA pode ser descrito como se segue. Cada palavra-código recebida ($\lfloor R \rfloor = \lfloor r_0, r_1, r_2, r_3, r_4, r_5 \cdot r_6 \cdot r_7, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{13}, r_{14} \rfloor$) pelo registrador da UCR (ver Figura 4.16, seção 4.3.1) que recebe o sinal multiplexado da linha, é transferida em paralelo (Clear 15 e Load 15) para o registrador do DCA. O circuito de lógica linear no GF(2) (ou-exclusivo's) forma então as equações de paridade com os dígitos da palavra recebida e, alimenta as entradas dos blocos de função de maioria (FM na Figura 4.20). A saída de um bloco função de maioria é um "1" lógico quando uma clara maioria das somas de paridade na sua entrada forem também "1". Caso contrário, i.é., no caso de que a maioria das somas de paridade na entrada derem um resultado igual a "0" ou, quando ocorrer um empate, o bloco função de maioria fornece em sua saída um "0" lógico. A escolha da saída do bloco de FM (ou do conjunto de blocos FM no caso de decodificação em dois passos), i.é. , a escolha do decodificador próprio a um dos códigos cíclicos utilizados pelo MDDCC, é feita através do peso da atividade recuperada que é calculado em paralelo por um contador combinacional que conta o número de 1's do padrão de atividade recuperado. Um circuito lógico combinacional, semelhante ao utilizado no CCA (seção 4.2.6), deriva as saídas X_1, X_2, X_5, X_7 e X_{11} a partir do peso da atividade. Esses sinais derivados selecionam o bloco de FM (ou blocos de FM) referentes a atividade, portanto, ao código cíclico em questão. Na tabela 4 é apresentada a tabela verdade do circuito lógico combinacional que deriva os

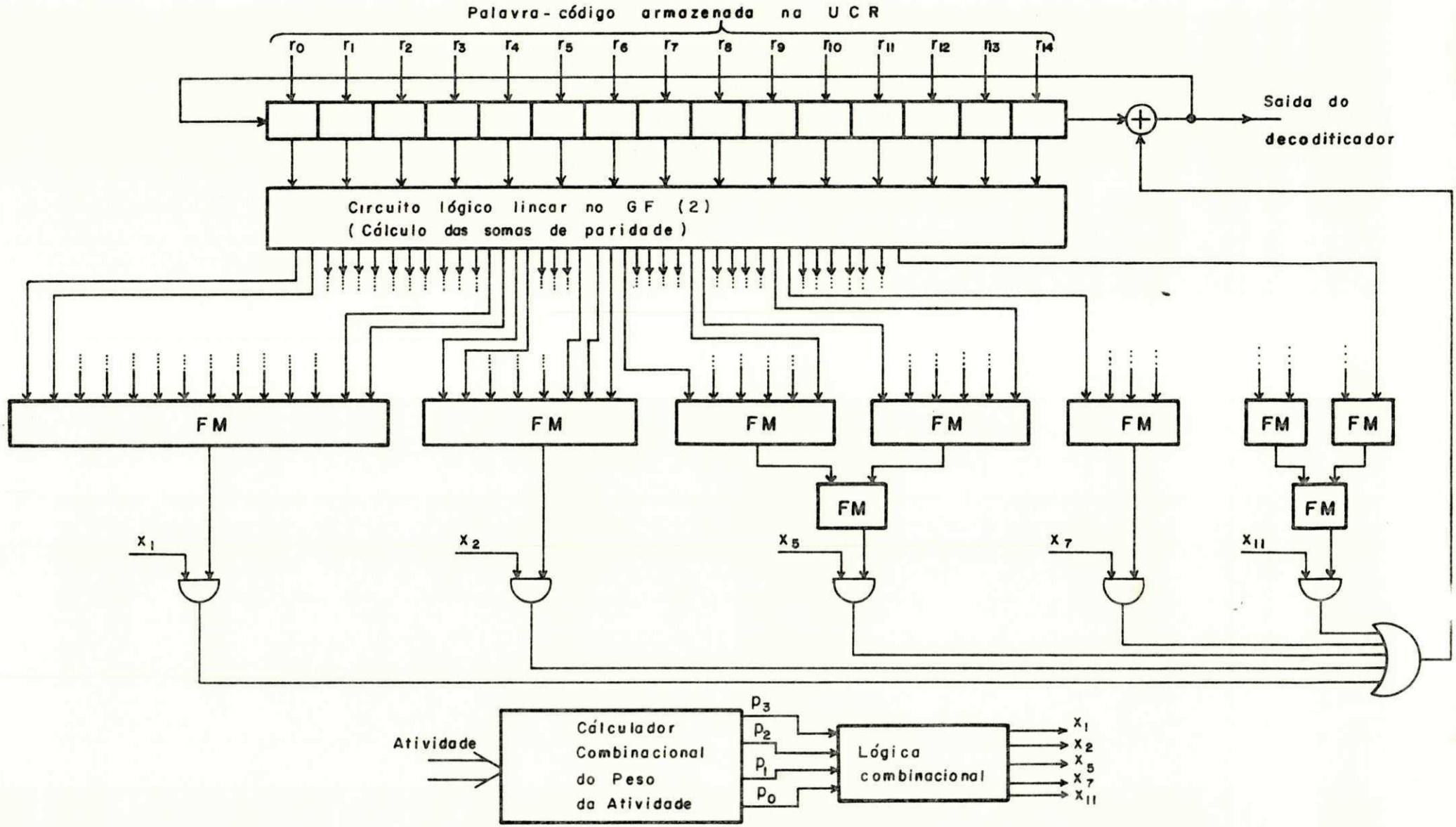


Figura 4. 20 Decodificador cíclico adaptativo

sinais X_1, X_2, X_5, X_7 e X_{11}. No caso da decodificação da palavra -código contendo a informação de atividade, o sinal $\overline{\text{Controle}}$ $\overline{7}$ atua e estabelece através do circuito lógico combinacional (observe na tabela 4 as saídas para o peso 0000) a saída do conjunto de blocos FM referentes ao código cíclico (15,11). A saída do bloco FM (ou blocos FM) escolhida é somada módulo - 2 com a saída do registrador do DCA para que a cada "clock", durante 15 "clock's", proceda a correção (dentro da capacidade t do código) dos erros que, por ventura, tenham ocorrido no dígito mais à direita no registrador.

ENTRADAS				SAÍDAS				
P_1	P_2	P_1	P_0	X_1	X_2	X_5	X_7	X_{11}
0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	0	0
0	0	1	0	0	1	0	0	0
0	0	1	1	0	0	1	0	0
0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0	0
0	1	1	0	0	0	0	1	0
0	1	1	1	0	0	0	1	0
1	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	1
1	0	1	0	0	0	0	0	1
1	0	1	1	0	0	0	0	1

Tabela 4

4.3.4 - DECOMPRESSOR DE DADOS (DD)

O Descompressor de Dados (DD) do receptor do MDDCC, mostrado na Figura 4.21, tem uma função exatamente inversa aquela do CD (seção 4.2.5) no transmissor. O DD reposiciona os bits de informação das 64 palavras-código, correspondentes aos canais multiplexados e decodificados pelo CCA , de acordo com a atividade recuperada, e os envia ao Conversor Série-Paralelo (CSP) (seção 4.3.5) para que se efetue a alocação dos mesmos aos destinatários correspondentes. O sinal $\overline{\text{Controle 7}}$ controla a entrada no DD das palavras-código decodificadas enquanto que os sinais Clear 1005 e Load 1005 e os sinais Clear 15 e Load 15 controlam o carregamento (leitura) em paralelo dos registradores de armazenamento da atividade e dos bits comprimidos, respectivamente.

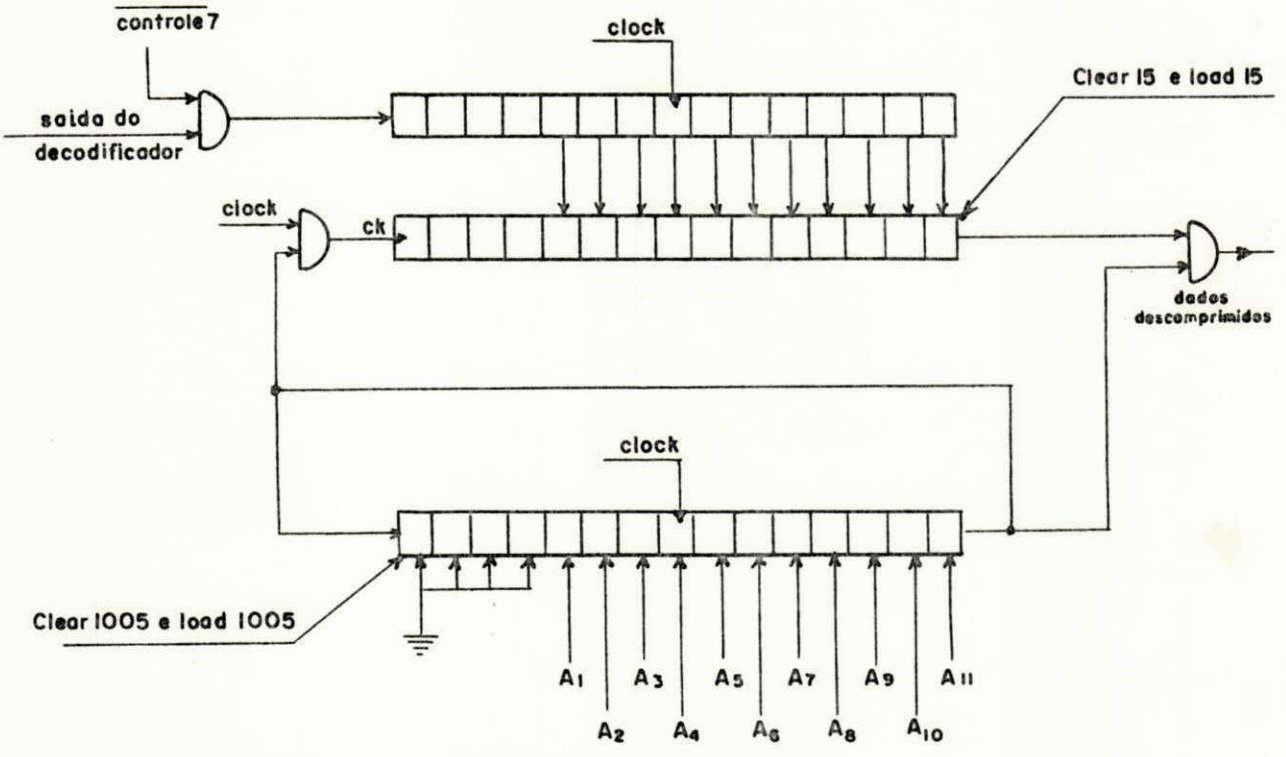
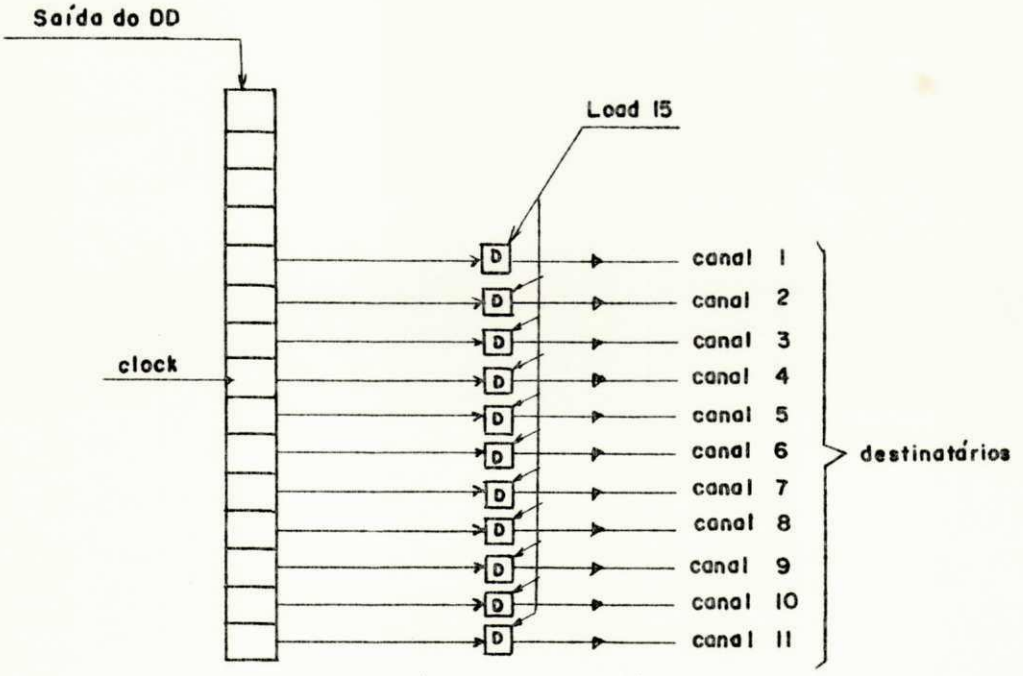


Fig. 4.21 - Descompressor de Dados

4.3.5 - CONVERSOR SÉRIE-PARALELO (CSP)

O Conversor Série-Paralelo mostrado na Figura 4.22 tem a função de receber em série do DD, os bits de informação dos canais multiplexados e alocá-los em paralelo aos destinatários. O sinal Load 15 controla a leitura em paralelo dos onze "Flip-Flops" tipo D que derivam os bits correspondentes aos canais multiplexados pelo MDDCC.



UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprigio Veloso, 882 - Tel (083) 321 7222-R 355
58 100 - Campina Grande - Paraíba

Figura 4.22 - Conversor Série-Paralelo

4.4 - COMENTÁRIOS

A implementação prática do MDDCC (ver Figuras 4.23 e 4.24) exigiu cerca de 150 circuitos integrados de pequena e média integração com um consumo médio de potencia da ordem de 20 watts (5 volts). Grande parte desses circuitos foram usados na implementação dos circuitos lógicos combinacionais que compõem o MDDCC. Uma minimização no número de componentes pode ser conseguida, usando-se memórias (PROM,EPROM, etc.) na implementação dos circuitos lógicos combinacionais . A escolha do tipo de decodificador cíclico utilizado, levou em conta, além das características vantajosas na correção de erros aleatórios da decodificação por função de maioria (seção 2.3.8), a economia do número de circuitos integrados (dentro dos disponíveis) na implementação de um tipo de decodificação comum a todos os códigos cíclicos usados pelo MDDCC. No receptor do MDDCC utilizou-se, em alguns casos, registradores de armazenamento de 15 estágios quando eram necessários somente 11 estágios. Isso se deve ao fato de que os registradores de leitura em paralelo utilizados possuiam 5 estágios e, também, o fato de que o uso de registradores de 11 estágios (de fato, 12 ou 16 estágios) exigiria uma temporização, semelhante ao Controle 4 no transmissor (seção 4.2.2), portanto, um aumento na complexidade da UCR. O tipo de implementação, sequencial com "clock" único, usado, levou em conta a velocidade dos circuitos TTL e a idéia de maximizar a velocidade do canal multiplexado permitida.

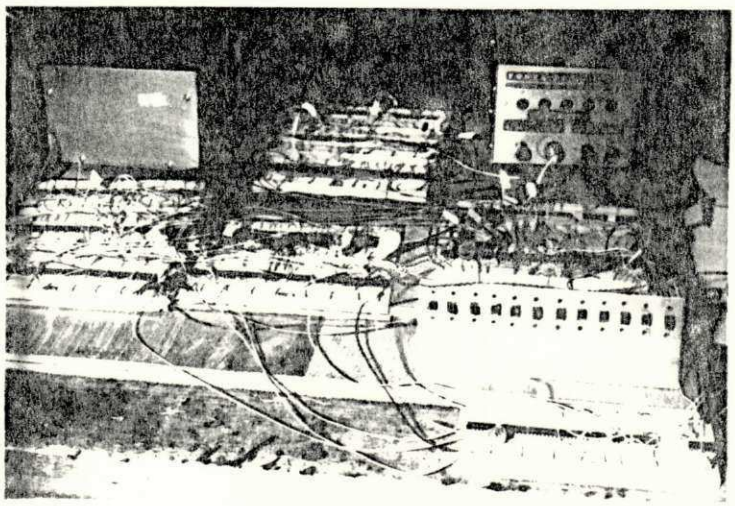


Fig. 4.23 - Montagem do transmissor do MDDCC

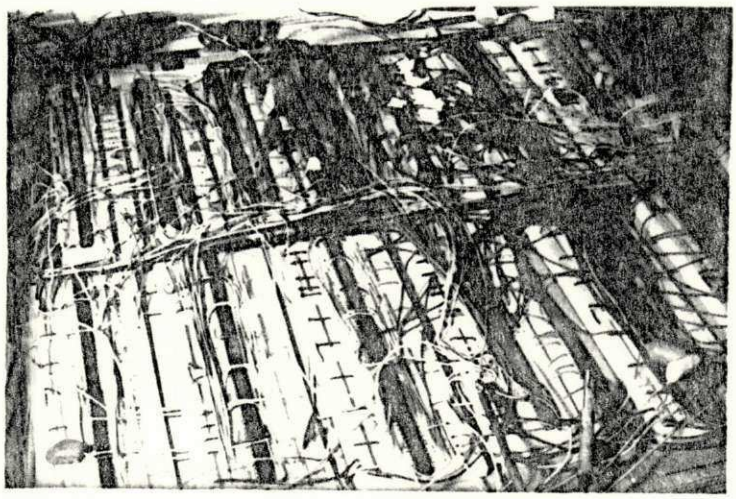


Fig. 4.24 - Montagem do receptor do MDDCC

O sistema de sincronismo de quadro usado mostrou-se eficiente, mesmo quando a probabilidade de simulação de tres palavras-código iguais consecutivas era máxima, i.é., quando apenas um canal estava ativo. Nesse caso, a probabilidade de simulação de tres palavras-código iguais consecutivas em um quadro corresponde a probabilidade de ocorrência no canal ativo de tres 0's ou tres 1's consecutivos.

A capacidade de correção de erros aleatórios dos códigos cíclicos utilizados foi verificada através da simulação de erros nos diversos bits das palavras-código. Observou-se, também, a capacidade extra de correção de erros aleatórios com alguns padrões de erro particulares. Esse fato sugere um estudo posterior quanto as prioridades dadas aos canais na entrada do MDDCC no que diz respeito à proteção contra o ruído.

Supondo-se um canal de transmissão CSB (seção 2.2.6) com probabilidade de erro igual a $p << 1$, tem-se que a probabilidade de erro na decodificação de uma palavra-código de 15 bits (ou um símbolo do canal extensão de ordem 15 do CSB (Abramson, 1963)) por função de maioria é limitada por

$$P_e \leq C_{15}^{t+1} \; p^{t+1} \; (1-P)^{15-(t+1)} + C_{15}^{t+2} \cdot p^{t+2} \; (1-P)^{15-(t+2)} + \ldots + C_{15}^{15} \; p^{15}$$

$$\leq C_{15}^{t+1} \cdot p^{(t+1)} \; (1-P)^{15-(t+1)}$$

$$\leq C_{15}^{t+1} \cdot p^{t+1}$$

onde $p << 1$, C_n^i é o número de combinações de i elementos tomados de um conjunto de n elementos e t é a capacidade de correção do código que forma a palavra-código. (Lucky e outros , 1968). Dessa forma, a probabilidade de erros aleatórios para os canais multiplexados pode ser estimada da seguinte maneira.

Código (15,11)	$P_e \cong 105\ p^2$	para $p = 10^{-3}$	$P_e \cong 10^{-4}$
Código (15,7)	$P_e \cong 405\ p^3$	para $p = 10^{-3}$	$P_e \cong 4 \times 10^{-7}$
Código (15,5)	$P_e \cong 1365 p^4$	para $p = 10^{-3}$	$P_e \cong 1{,}3 \times 10^{-9}$
Código (15,2)	$P_e \cong 3003 p^5$	para $p = 10^{-3}$	$P_e \cong 3 \times 10^{-12}$
Código (15,1)	$P_e \cong 6435 p^8$	para $p = 10^{-3}$	$P_e \cong 6 \times 10^{-21}$

A probabilidade de erro na recuperação da atividade devido a erros aleatórios ocorridos na transmissão do quadro multiplexado pode ser obtida estimando-se a probabilidade de ocorrência de mais de um erro aleatório por palavra, em pelo menos duas das palavras-código do quadro multiplexado que contém a informação de atividade. Essa estimativa pode ser dada por

P erro na recuperação da atividade devido a erros aleatórios no canal de transmissão

$$\cong \frac{1}{C_{67}^3} \left\{ C_3^2 \cdot \left[C_{15}^2 \cdot p^2 (1-p)^{13} \right]^2 \cdot \left[1 - C_{15}^2 \cdot p^2 (1-p)^{13} \right] \right\}$$

$$\cong \frac{1}{5 \times 10^4} \left\{ 3 \cdot \left[105\ p^2 \right]^2 \right\}$$

$$\cong 3/5\ p^4$$

para $p = 10^{-3}$ $\quad\quad \cong 6 \times 10^{-13}$

Daí tem-se que a recuperação da atividade está, em termos de erros aleatórios no canal de transmissão, com uma proteção melhor que a informação dos canais multiplexados no caso de apenas dois canais ativos. Por outro lado, a repetição por tres vezes da atividade codificada, dá ao sistema de recuperação de atividade uma capacidade de correção de erros em "burst" (seção 2.3.9) superior a de todos códigos cíclicos usados na decodificação da informação dos canais multiplexados. Por exemplo, o sistema de recuperação de atividade utilizado, corrige todos "burst" de erros de comprimento 16 sobre a informação de atividade repetida e os "burst" de comprimento 45 com densidade menor ou igual a 4/45 (Lin, 1970).

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 Tel. (083) 321-7222-R 355
58.100 - Campina Grande - Paraíba

CAPÍTULO V

C O N C L U S Õ E S

Este trabalho segue uma das linhas de pesquisa de interesse da equipe de comunicações digitais do CCT-UFPb que é a de desenvolvimento de sistemas de multiplexação digital "inteligentes".

O sistema MDDCC foi implementado para 11 canais síncronos por conveniência (seção 4.1), podendo ser ampliado para um número maior de canais sem maiores complexidades na concepção do sistema. Dessa forma o sistema MDDCC não apresenta, em princípio, limitações quanto ao número de canais possíveis de serem multiplexados. A flexibilidade dos padrões de atividade na troca de capacidade de canal por capacidade de correção de erros aleatórios (seção 4.1) é limitada pelos códigos cíclicos binários utilizados, porém, não exige complexidade adicional no processo de codificação (seção 4.2.4). Um outro trabalho de tese, ora em desenvolvimento na equipe de comunicações digitais, estuda os sistemas

de multiplexação por divisão em códigos no sentido de chegar a um sistema universal com códigos simples de redundância variável que incorpore as vantagens dos já existentes (por exemplo, o MDDCC) evitando as limitações quanto ao número de canais possíveis de serem multiplexados e quanto a flexibilidade de troca de capacidade de canal por capacidade de controle de erros.

O MDDCC foi implementado na prática com circuitos integrados de pequena e média integração (seção 4.4) e apresentou um desempenho satisfatório. A complexidade envolvida com os circuitos integrados de pequena e média integração sugere a aplicação de microprocessadores ao sistema quando a velocidade do canal multiplexado não for um fator restritivo. Nesse sentido, foi começado um trabalho de iniciação científica com o "software" do transmissor do MDDCC. A flexibilidade dos microprocessadores sugere, ainda, a incorporação de alocação dinâmica de canais ao sistema MDDCC em conjunto com a capacidade de controle de erros adaptativa.

O novo método de sincronização desenvolvido para o MDDCC mostrou-se experimentalmente, satisfatório. Sugere-se o estudo posterior da recuperação do "clock" aproveitando-se a periodicidade das palavras-código.

A decodificação cíclica por função de maioria sugere (seção 4.4) um estudo de prioridades dos canais na entrada do MDDCC quanto a confiabilidade.

O trabalho carece de aprofundamentos nos

aspectos teóricos envolvidos no desenvolvimento do sistema já que o objetivo concentrou-se no desenvolvimento e implementação prática do sistema MDDCC particular apresentado.

APÊNDICE I

ALGEBRA MODERNA E ESPAÇOS VETORIAIS

As definições apresentadas neste Apêndice podem ser encontradas, por exemplo, em Peterson (1972), Lin (1970), Berlekamp(1968) e Rocha (1976).

GRUPO : Um conjunto de elementos para os quais é definida uma operação (+) onde os seguintes axiomas são válidos. Se a, b e c são elementos do conjunto então:

(1) Fechamento: a + b (lê-se a operado com b) também está no conjunto.

(2) Lei Associativa: (a + b) + c = a + (b + c)

(3) Existe um elemento identidade único. No caso de operação (+) chamada de adição o elemento identidade é 0 e 0 + a = a.

(4) Cada elemento do grupo tem um elemento inverso único. No caso da operação denominada adição o elemento inverso de a, por exemplo, é denotado por -a onde a + (-a) = 0.

SUB-GRUPO : Um sub-conjunto de elementos de um grupo que satisfaz todas as propriedades de um grupo e, portanto, forma ele próprio um grupo em relação a operação definida (+) pelo grupo.

GRUPO ABELIANO: Um grupo que satisfaz a lei comutativa a + b = b + a.

ANEL : Um conjunto de elementos que forma um grupo Abeliano sob a operação denominada adição (+) e satisfaz os seguintes axiomas com respeito a uma outra operação denominada multiplicação (.):

(1) Fechamento: a.b está também no conjunto.

(2) Lei Associativa: a.(b.c) = (a.b).c

(3) Lei Distributiva: a.(b + c) = a.b + a.c e (b + c).a = b.a + c.a

Um anel é comutativo se a.b = b.a

CAMPO : Um conjunto de elementos que formam um anel e os elementos diferentes de zero formam um grupo Abeliano sob a operação (.).

CAMPOS DE GALOIS: GF(q) - Pode ser mostrado que para $q = p^i$, onde p é um número primo e i um inteiro qualquer, existe um campo contendo um número finito q de elementos. Esse campo é chamado de Campo de Galois de q elementos. Esses q elementos serão os inteiros de 0 a q-1 se e só se q for primo. GF(2), por exemplo, é o Campo de Galois com os elementos binários {0,1}.

ANEL POLINOMIAL: O conjunto de todos polinômios em X com coeficientes do GF(q), forma um anel comutativo que é chamado anel de polinômios sobre o GF(q).

IDEAIS : O conjunto de todos polinômios, com coeficientes no GF(q), de forma g(x) . p(x), onde p(x) é um polinômio qualquer, é chamado de Ideal gerado por g(x).

CLASSES RESÍDUOS: Um ideal pode ser visto como um sub-conjunto de um anel polinomial. Os elementos desse anel, que não estão no ideal, formam o que é chamado Classes Resíduo. Os elementos de cada classe resíduo são caracterizados pelo fato de que a subtração entre qualquer par de seus membros dá um elemento no ideal.

ANEL CLASSE RESÍDUO: O conjunto de classes resíduos, defini -

dos acima, forma um anel chamado de anel classe resíduo com respeito a um ideal.

ESPAÇOS VETORIAIS: A sequência $[V] = [v_1, v_2, v_3, \ldots v_n]$, onde os componentes são elementos do GF(2), i.é., 0 ou 1, é chamada uma n-upla sobre o GF(2). Existem 2^n n-uplas diferentes, para um determinado n, devido a natureza binária dos componentes v_i. A adição de duas n-uplas $[U]$ e $[V]$ é definida como se segue:

$$[U] = [u_1, u_2, u_3, \ldots, u_n]$$

$$[V] = [v_1, v_2, v_3, \ldots, v_n]$$

$$[U] + [V] = [(u_1+v_1), (u_2v_2), (u_3+v_3), \ldots, (u_n+v_n)]$$

onde $u_i + v_i$ representa a soma módulo-2 de u_i e v_i

A multiplicação escalar de uma n-upla binária por um elemento do GF(2) é definida assim

$$a.[v_1, v_2, \ldots, v_n] = [av_1, av_2, \ldots, av_n]$$

o produto interno de duas n-uplas $[U]$ e $[V]$ é definida assim

$$[U] . [V] = u_1v_1 + u_2v_2 + \ldots + u_nv_n$$

onde a adição e multiplicação são módulo-2. Um espaço vetorial V_n sobre o GF(2) é definido como o conjunto de todas n-uplas binárias possíveis. Um sub-espaço S_n de V_n é definido como um sub-conjunto de V_n que contém a n-upla com todos componentes zeros e a adição de quaisquer duas n-uplas em S_n sempre resulta numa terceira n-upla pertencente a S_n. Dadas i n-uplas $[V_1]$, $[V_2]$, ..., $[V_i]$, uma combinação linear delas é definida como

$$[U] = c_i[V_1] + c_2[V_2] + \ldots + c_i[V_i]$$

onde os coeficientes c_i são elementos do GF(2). Se existe c_i's onde pelo menos um deles é diferente de zero, tal que

$$c_i[V_1] + c_2[V_2] + \ldots + c_i[V_i] = 0$$

então o conjunto $\{[V_1], [V_2], \ldots, [V_i]\}$ é dito ser linearmente dependente. Se a condição acima só é satisfeita para a condição $c_1 = c_2 = \ldots = c_i = 0$, então o conjunto $[V_1]$, $[V_2]$, ..., $[V_i]$ é dito ser linearmente independente. Em qualquer espaço vetorial ou sub-espaço vetorial existe pelo menos um conjunto de vetores (n-uplas) linearmente independentes que, através de combinações lineares, geram todos os outros vetores no espaço ou sub-espaço. Esse conjunto é chamado de base do espaço vetorial ou sub-espaço. A dimensão de um espaço vetorial é o número de vetores em sua base.

B I B L I O G R A F I A

01. ABRAMSON, N. - Information Theory and Coding, McGraw-Hill, New York, 1963.

02. BERLEKAMP, E. R. - Algebraic Coding Theory, McGraw-Hill , New York, 1968.

03. CARLSON, B. - Communication Systems: An Introduction to Signals and Noise in Electrical Communication, McGraw-Hill, New York, 1968.

04. DAVIES, D. & BARBER, D. - Communication Networks for Computers, Wiley, 1973.

05. DOLL, D.R. - Multiplexing and Concentration, Proc. IEEE, vol. 60, pp. 1313-1321, November, 1972.

06. DOLL, D.R. - Basics of Network Design, Basics of Data Communications, Electronic Book Series, McGraw-Hill , New York, 1976.

07. ELIAS, P. - Error-Free Coding, IRE Transactions, IT-4, p. 29, September, 1954.

08. GOLOMB, S. - Digital Communications with Space Application, Prentice-Hall, Englewood Cliffs, 1969.

09. GORDON, J. & BARRETT, R. - Digital Majority Logic Multiplex using Walsh Functions, Proceedings of the Symposium on Application of Walsh Functions, pp. 171-176 , Washington, 1971.

10. GORDON, J. & BARRETT, R. - Correlation Recovered Adaptive Majority Multiplexing, Proc. IEEE, vol. 118, nº 3/4, March/April, pp. 417-422, 1971.

11. HARMUTH, H.F. & MURTY, S. - Sequency Multiplexing of Digital Signals,Proceedings of the Symposium on Application of the Walsh Functions, Washington, 1973.

12. KARP, H. R. & LAPIDUS, G. - What the Future Has in Store for Data Communications, Basics of Data Communication, Electronic Book Series, McGraw-Hill, New York, 1976.

13. LIN, S. - An Introduction to Error-Correcting Codes,Prentice-Hall, New Jersey, 1970.

14. LUCKY, R.W; SALZ, J. & WELDON Jr., E.J. - Principles of Data Communication, McGraw-Hill, New York, 1968.

15. MASSEY, J.L. - Threshold Decoding, the MIT Press, Cambridge, Massachusetts, 1963.

16. MULLER, D.E. - Applications of Boolean Algebra to Switching Circuit Design and to Error Detection, IRE Trans.,EC-3, p.6, September, 1954.

17. PETERSON, W.W. & WELDON Jr., E.J. - Error-Correcting Codes, the MIT PRESS, Cambridge, Massachusetts, 1972.

18. ROCHA Jr., V.C. - Versatile Error-Control Coding Systems, Ph.D Thesis, University of Kent at Canterbury, May, 1976.

19. ROCHA NETO, I. - Adaptive Majority Multiplexing Techniques Ph.D Thesis, University of Kent at Canterbury, 1975.

20. RUDOLPH, L.D. - A Class of Majority Logic Decodable Codes IEEE Trans. on Information Theory, vol. II - 13, p.305, April, 1967.

21. RUDOLPH, L.D. & HARTMANN, R.P. - Decoding by Sequential Code Reduction, IEEE Trans. on Information theory,vol. II - 19, nº 4, pp. 549-555, July, 1973.

22. SHANNON, C.E - A Mathematical Theory of Communication , Bell Systems Technical Journal, 27, 1948.

23. SMITH, R.S. - Multiplexing Cuts Cost of Communications Lines, Basics of Data Communication, Electronic Book Series, McGraw-Hill, 1976.

24. TOWNSEND, R.L & WELDON Jr., E.J. - Self-Orthogonal Quasi - Cyclic Codes, IEEE Trans. on Information Theory, vol. II-13, 1967.

25. VILAR FRANÇA, R.M. - Multiplex Adaptativo por Função de Maioria e Decisão Suave, Tese de Mestrado, Universidade Federal da Paraíba, Setembro, 1978.