UNIVERSIDADE FEDERAL DA PARAÍBA - UFPB

Circuito integrado para transmissão digital de sinal de áudio visando baixos custo e consumo

Antonio Augusto Lisboa de Souza

Dissertação de Mestrado submetida à Coordenação de Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Paraíba – Campus II como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica

Área de concentração: Processamento da Informação

Elmar Uwe Kurt Melcher, Dr.
Raimundo Carlos Silvério Freire, Dr.
Orientadores

Campina Grande. Paraíba. Brasil

S819c Souza, Antonio Augusto Lisboa de
Circuito integrado para transmissao digital de sinal de audio visando baixos custo e consumo / Antonio Augusto Lisboa de Souza. - Campina Grande, 2001.
79 f.

Dissertacao (Mestrado em Engenharia Eletrica) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Microeletronica 2. Circuitos Integrados 3. Transmissao sem Fio 4. Dissertacao - Engenharia Eletrica I. Melcher, Elmar Uwe Kurt II. Freire, Raimundo Carlos Silverio III. Universidade Federal da Paraiba - Campina Grande (PB) IV. Título

CDU 621.3.049.77(043)

CIRCUITO INTEGRADO PARA TRANSMISSÃO DIGITAL DE ÁUDIO VISANDO BAIXOS CUSTO E CONSUMO

ANTONIO AUGUSTO LISBOA DE SOUZA

Dissertação Aprovada em 12.02.2001

PROF. ELMAR UWE KURT MELCHER, Dr., UFPB
Orientador

PROF. RAIMUNDO CARLOS SILVÉRIO FREIRE, Dr., UFPB
Orientador

PROF. FRANCISCO MARCOS DE ASSIS, Dr., UFPB
Componente da Banca

PROF. JOÃO MARQUES DE CARVALHO, Ph.D., UFPB
Componente da Banca

CAMPINA GRANDE - PB
Fevereiro - 2001

Dedicatória

Dedico este trabalho a meus pais, Aluísio e Sara, e à minha esposa, Adriana

Agradecimentos

Gostaria de agradecer a todas as pessoas que ao longo destes anos contribuíram para minha formação profissional e pessoal.

Agradeço também às pessoas que acreditaram neste projeto e que, mesmo na incerteza de resultados, esforçaram-se para que se tornasse realizado. Um pedaço deste esforço está contido aqui neste trabalho.

Resumo

Este trabalho descreve uma implementação de parte de um sistema transmissor/receptor de áudio digital para aplicações que requerem baixos custo e consumo, onde o receptor pode ser controlado via transmissor.

Após análise da quantidade de circuitos de mercado necessária à implementação de tal sistema, decidiu-se por implementar um circuito integrado de aplicação específica.

Vários algoritmos de compressão de áudio e métodos de modulação digital foram estudados. A arquitetura digital proposta foi implementada utilizando-se a linguagem Verilog de descrição de hardware.

Finalmente, resultados de simulações e do mapeamento em FPGA são mostrados.

Abstract

This work describes an implementation of part of a digital audio transmitter and receiver targeted for low power, low cost applications, where the receiver can be controlled via transmitter.

After evaluating the amount of off-the-shelf components needed to implement such a system, it was decided to implement an ASIC (application specific integrated circuit).

Audio compression algorithms along with digital modulation methods were studied. The proposed digital architecture was implemented using Verilog hardware description language.

Finally, results from simulations and FPGA implementation are shown.

Índice

Lista de Tabelas

Lista de Figuras

1. Introdução

A transmissão sem fio de sinais de áudio encontra inúmeras aplicações. Há uma tendência crescente de concepção de produtos eletro-eletrônicos que interagem com o usuário através de sinais de áudio.

Aparelhos de teleconferências e receptores de áudio de aparelhos de televisão são exemplos de produtos já disponíveis no mercado que necessitam da transmissão e recepção de sinais de áudio.

Para estas aplicações, encontram-se no mercado de um lado circuitos integrados de baixo custo e baixa performance, geralmente operando com modulação analógica para transmissão apenas do sinal de áudio [5] e sendo controlados por um circuito digital à parte, e de outro, componentes de alta complexidade, com técnicas digitais de tratamento de sinais em encapsulamentos dispendiosos que se destinam principalmente a sistemas de alta fidelidade, onde o custo final do produto não representa fator determinante[4].

No caso de aparelhos de alto consumo, como os aparelhos de televisão, a pressão sobre o preço final do produto dificulta a utilização de componentes de alto desempenho para funções secundárias. A solução para tal problema seria um componente dedicado, de baixo custo e complexidade exibindo performance satisfatória.

O Ministério das Comunicações estabelece as condições a serem cumpridas para transmissão de sinais. Existe uma faixa de frequências disponível para *Audição em grupo com radiação restrita*. Tal faixa vai de 72.10MHz a 75.9MHz, com canais cuja largura de banda é de 200KHz [3].

Com esta largura de canal é possível transmitir sinal de áudio estéreo de boa qualidade com alta confiabilidade a partir de um par casado transmissor/receptor de baixa complexidade e, portanto baixo custo, que utilize modernas técnicas digitais de tratamento de sinais.

Além disso, pode-se projetar um sistema robusto, onde o receptor seja controlado a partir do transmissor. Para que isso seja alcançado, a utilização de técnicas digitais se torna indispensável.

As técnicas digitais de processamento de sinais oferecem várias vantagens em relação às técnicas analógicas: a tolerância de componentes não afeta a performance do sistema, facilitando a produção em massa, sendo ainda imune às condições externas, como exemplo a temperatura; baixos consumo e tensão, requeridos nas aplicações móveis; qualidade

superior, sendo a única restrição de qualidade a quantidade de bits dos conversores analógico-digitais utilizados; flexibilidade, principalmente no caso dos filtros digitais, que podem ser re-configurados facilmente através de valores armazenados em memória[10].

Tal fato tem impulsionado a migração dos sistemas analógicos para a tecnologia digital, como aconteceu com a mídia em vinil (analógica). O baixo consumo dos circuitos digitais teve um impacto acentuado no mercado de celulares, haja vista a competição dos fabricantes em oferecer um aparelho de peso reduzido por utilizar uma bateria mais leve.

O problema da complexidade de tratamento de sinais foi resolvido com a densidade alcançada pelos circuitos VLSI. Desta forma, torna-se possível não apenas transmitir o sinal desejado, mas integrar em uma mesma pastilha de silício um circuito complexo de tratamento de sinal junto com o circuito de transmissão, de forma que hoje, é possível haver mais recursos computacionais num simples telefone celular para transmissão de voz humana que nos primeiros computadores pessoais[11]

Por esta razão, estamos propondo um projeto tendo como objetivo a transmissão de áudio digital via ondas eletromagnéticas. Através da transmissão digital, o consumidor tem acesso não apenas a um sinal de áudio de qualidade comparável ao CD, como também a uma infinidade de serviços que outrora não eram concebíveis com as tecnologias analógicas (AM/FM) utilizadas até então para a transmissão de áudio[1].

Esta dissertação está dividida em 10 capítulos. No capítulo 2 , intitulado *Sinal Digital de Áudio*, é feita uma análise sumária do teorema da amostragem e da representação de amostras através do método PCM linear, onde comentam-se detalhes de implementação, enfatizando vantagens e desvantagens.

Nos capítulos 3 e 4 , mostra-se o estudo feito sobre algoritmos de compressão de áudio e técnicas de modulação digital, respectivamente. Este estudo, abrangendo análise de eficiência e complexidade de implementação, culminou na escolha do algoritmo e técnica a serem implementados.

No capítulo 5 são descritas as características de desempenho que o circuito final deve apresentar, ou seja, descreve os pontos de aferição de satisfação do sistema proposto .

O capítulo 6 trata da implementação do algoritmo de compressão de áudio escolhido, o ADPCM, ou Adaptive Differential Pulse Code Modulation.

Inicialmente é feita uma descrição do algoritmo em etapas de funcionamento, tendo em vista a implementação em hardware. Em seguida, usa-se a linguagem Verilog de descrição de hardware para a síntese de diferentes circuitos que executam a mesma função, para que se compare os recursos computacionais requeridos por cada um deles. Finalmente, o hardware a ser utilizado no circuito final é escolhido e os resultados de simulações são analisados.

O capítulo 7 descreve a implementação do modulador digital usando basicamente a mesma estruturação do capítulo 6.

No capítulo 8 tem-se o "*big picture*", ou seja, a visão completa do circuito final. O circuito digital final é descrito e implementado a partir dos blocos implementados nos capítulos 6 e 7. Alguns detalhes de concepção são inseridos e são feitas simulações com arquivos binários servindo como fonte de amostras de áudio. Estas simulações levam em conta a inserção de ruído Gaussiano pelo canal de comunicação.

Finalmente, no capítulo 9 sugerem-se trabalhos futuros a partir dos resultados obtidos neste trabalho e no capítulo 10 consta a literatura estudada. Além desta dissertação, foi criada uma página HTML de interface amigável com o usuário, que contém todos os arquivos de descrição de hardware desenvolvidos ao longo deste trabalho, assim como as rotinas de simulação em linguagem C, MATLAB e Verilog HDL.

2. Sinal Digital de Áudio

O sinal de áudio disponível na natureza é contínuo no tempo e em amplitude. A fim de tornar possível o processamento digital do sinal de áudio, este será discretizado tanto no tempo quanto em amplitude. O processo de discretização no tempo recebe o nome de amostragem, enquanto a discretização em amplitude recebe o nome de quantização.

2.1 Amostragem

A conversão do sinal analógico para o domínio digital começa pela amostragem, processo em que o sinal é medido a intervalos igualmente espaçados que definem a taxa de amostragem (frequência de amostragem).

A figura 2.1 mostra o processo de amostragem. Este processo consiste na multiplicação de um trem de pulsos com amplitude constante e frequência f_S, onde f_S é a frequência de amostragem, pelo sinal analógico a ser convertido. O resultado desta multiplicação é um trem de pulsos com amplitude variável. O espectro do sinal amostrado é o resultado da convolução em frequência dos espectros originais (sinal e trem de pulsos) :

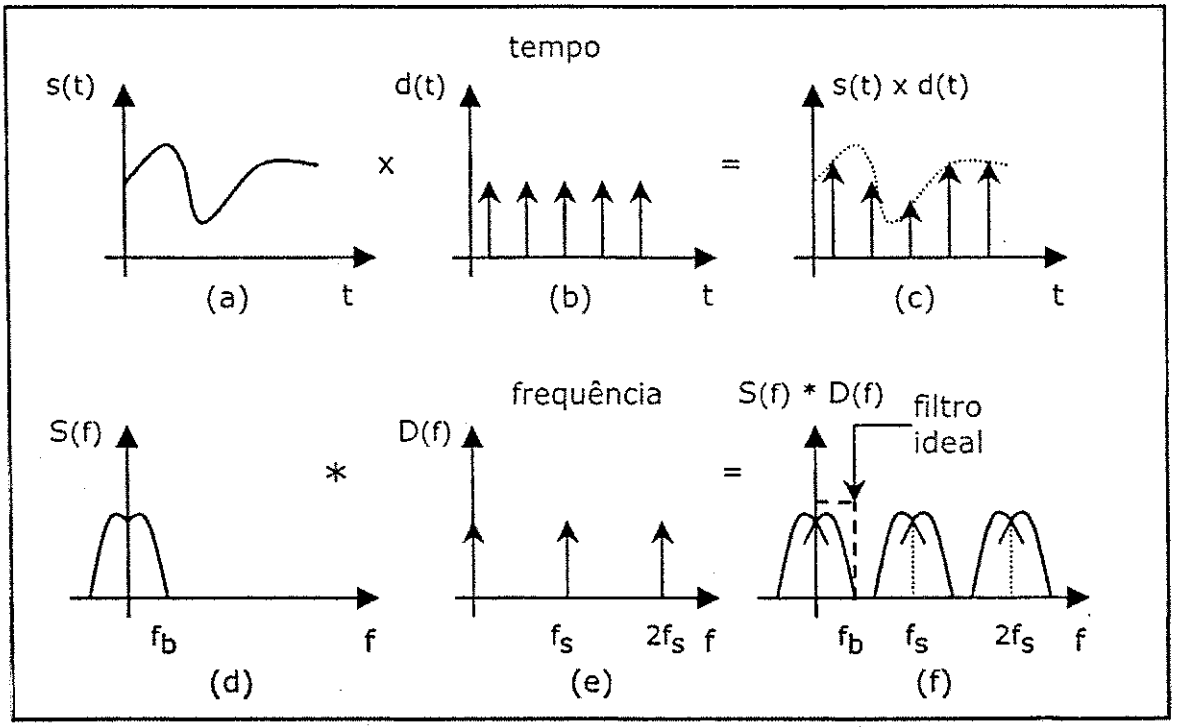


Figura 2.1 Amostragem. Multiplicação de (b) um trem de pulsos com amplitude constante por (a) um sinal qualquer limitado em frequência f_b. O resultado desta multiplicação é (c) um trem de pulsos com amplitude variável e frequência f_S. Este processo corresponde à convolução do (e) espectro do trem de pulsos com (d) o espectro do sinal sendo amostrado. A análise do (f) espectro resultante mostra que o sinal original pode ser reconstruído a partir de um filtro passa baixa ideal com frequência de corte f_b.

Se então a frequência de amostragem for maior ou igual ao dobro da frequência limite do sinal, na figura 2.1 tendo o valor f_b, não haverá o efeito de superposição de componentes espectrais(efeito aliasing), e de posse de um filtro passa-baixas ideal com frequência de corte f_b pode-se assim recuperar o sinal contínuo no tempo a partir de suas amostras. O limite teórico da frequência de amostragem $2f_b$ como o menor valor de frequência de amostragem que torna possível a recuperação do sinal amostrado, chamado frequência de Nyquist, pressupõe a existência de filtros ideais, motivo pelo qual é processo usual a amostragem a uma frequência pouco acima da frequência de Nyquist [2], processo chamado de sobre-amostragem. Além disso, para assegurar o limite em frequência do sinal, um filtro passa-baixas anti-aliasing geralmente precede à amostragem.

2.2 Modulação por Codificação de Pulso (PCM)

Um dos métodos mais utilizados na representação de amostras A_n de um sinal A qualquer, é a modulação linear por codificação de pulso (Linear PCM, Pulse Code Modulation), motivo pelo qual conversores analógico-digitais lineares, dispositivos eletrônicos que executam tal função, são facilmente encontrados no mercado. Um conjunto de R bits é atribuído a cada amostra, podendo esta assumir, portanto, um entre 2^R valores distintos.

Desta forma, tem-se que:

$$A_n = Aq_n + eq_n \qquad (2.1)$$

onde:

A_n é o valor da amplitude da amostra n do sinal de áudio;

Aq_n é o valor atribuído à amostra;

eq_n é o erro introduzido no processo de quantização, ou seja, a diferença entre o valor real e o valor atribuído à amostra.

O *erro de quantização* pode ser definido em termos do número de bits de um conversor A/D: definindo o limiar de escolha do conversor como sendo o valor médio entre dois valores adjacentes, percebe-se que o maior desvio possível do valor quantizado em relação ao valor real equivale à metade da diferença entre os dois valores adjacentes (diferença do bit menos significativo na palavra quantizada). Assim, se o bit menos significativo representa diferenças de 10 mV, o maior erro de quantizacão possível será de 5 mV.

Pode-se mostrar que a relação sinal-ruído, SNR, de um quantizador PCM linear de R bits é dado pela equação 2.2 [6]:

$$SNR = 6R - 7.2 \ dB \qquad (2.2)$$

O acréscimo de 1 bit na *palavra de quantização* implica na duplicação da precisão de representação da amostra (R+1 bits $=>2^{R+1}=2^{R+1}x2$ possíveis valores), de forma que o erro máximo de quantização será reduzido à metade. Este acréscimo representa um ganho de 6 dB na relação sinal-ruído do sinal quantizado.

Obter-se-á ao final da quantização um fluxo de f_S*R bits/s, onde f_S é a frequência de amostragem e R é o número de bits atribuído a cada amostra. A escolha do tamanho da palavra de quantização (número de bits representando cada amostra) leva em consideração a fidelidade a ser atingida.

O sistema do atual compact disc (CD), referência em qualidade de som, utiliza conversores de 16 bits, obtendo uma relação sinal-ruído de aproximadamente 90 dB. Utiliza frequência de amostragem de 44.1KHz (44.100 amostras por segundo) para garantir a recuperação de sinais de até 20KHz. Portanto, a taxa de bits necessária para representação do sinal de áudio estéreo é de 16X2(direito e esquerdo)X44.100=1.411.200 bits por segundo[2].

Uma das principais vantagens dos sistemas PCM é a flexibilidade: a partir das amostras do sinal no tempo quantizadas utilizando PCM, pode-se proceder diretamente ao processamento digital, aritmética, análise espectral, ou filtragem[7]. Como se trata de uma técnica bastante utilizada, o projetista de sistemas discreto encontra com facilidade dispositivos para tal conversão. Além disso, para os projetistas de circuitos integrados, algumas empresas disponibilizam células destes conversores em seus design kits.

A desvantagem do sistema para o caso específico de tratamento de sinais de áudio é a alta taxa requerida para transmissão, como visto acima.

3. Compressão de áudio

O fluxo de bits associado ao sinal de áudio sem compressão (p.ex. PCM) contém redundâncias. A compressão digital de áudio se propõe, através de algoritmos específicos, a extrair das amostras de áudio apenas a informação julgada pertinente. Tais algoritmos baseiam-se na peculiaridade dos sinais de áudio comumente transmitidos, p.ex. voz, ou nas características do sistema auditivo humano, ou em ambos. Desta forma, busca-se conseguir uma menor taxa de bits para representação do sinal de áudio no tempo, o que se reflete numa melhor armazenagem e transmissão.

Existem duas classes básicas de algoritmos de compressão de áudio: os que manipulam as amostras exclusivamente no domínio do tempo e os que fazem uso de ferramentas de análise espectral. Os diversos algoritmos que se enquadram nestas classes diferem nos compromissos entre complexidade de síntese, qualidade do sinal de áudio comprimido e capacidade de compressão[2]. A seguir serão apresentados alguns dos algoritmos mais utilizados.

3.1 Domínio do tempo

Os compressores que trabalham exclusivamente no domínio do tempo são geralmente de menor complexidade em relação aos compressores que necessitam de análise espectral do sinal.

3.1.1 μ-LAW

O ouvido humano apresenta uma característica logarítmica de sensibilidade. Isto significa que sinais de menor amplitude requerem uma maior precisão de quantização em relação aos sinais de maior amplitude. O sistema PCM utiliza um quantizador uniforme para toda a faixa dinâmica do sinal de entrada, ou seja, a diferença entre valores adjacentes é a mesma para toda a escala. Uma melhor alternativa para quantização do sinal de áudio consiste na utilização de um quantizador não-uniforme [8]. A transformação μ-law é uma técnica de compressão de áudio especificada pela Norma G.711 do Comiteé Consultatif Internationale de Telégraphic et Téléphonic (CCITT) que utiliza quantizador logarítmico[2]. O quantizador baseia-se na seguinte relação logarítmica entre entrada e saída:

$$|y| = \frac{\log(1+\mu \times |x|)}{\log(1+\mu)} \qquad (3.1)$$

onde:
|y| é a amplitude normalizada do sinal de saída (magnitude)
|x| é a amplitude normalizada do sinal de entrada (magnitude)
μ é um parâmetro de compressão.

A figura 3.1 mostra a relação entre *y* e *x*, de acordo com a equação 3.1, para diversos valores de μ :

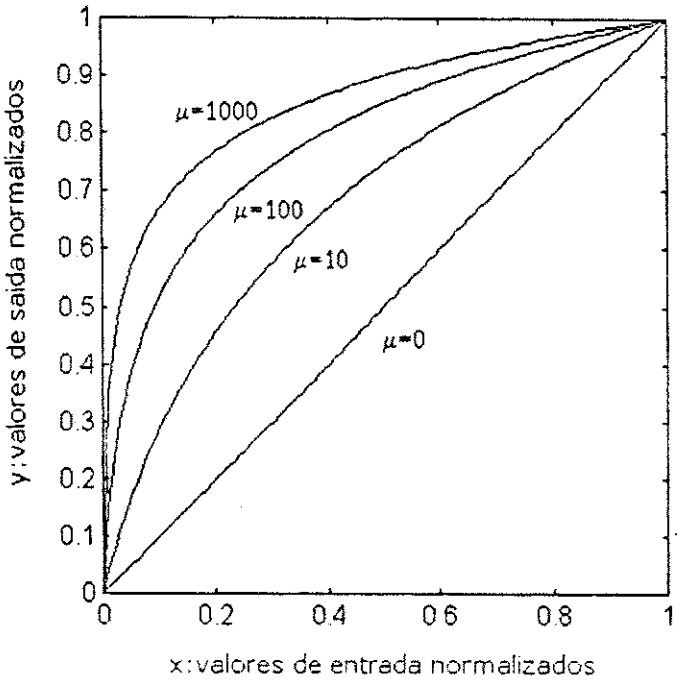


Figura 3.1 Relação entrada-saída função logarítmica utilizada na compressão μ-law.

Os sinais de amplitude reduzida serão assim quantizados com maior precisão em relação aos sinais de grande amplitude. O valor μ=255 tem sido adotado como norma para sinais de áudio nos Estados Unidos e Canadá. No caso específico de sinais de voz, a utilização do valor supracitado resulta no aumento de aproximadamente 24 dB na relação sinal-ruído do sinal quantizado, em relação à quantização uniforme PCM com mesmo número de bits[9].

Isto significa que um sistema uniforme PCM necessita de 12 bits/amostra para conseguir equivalente fidelidade a um sistema μ-law com 8 bits/amostra.

A implementação de um quantizador não linear (p.ex: logarítmico) pode ser concebida através da passagem prévia do sinal por um dispositivo não-linear, seguido de uma quantização uniforme. A característica entrada-saída do dispositivo não-linear deve estar de acordo com a curva de quantização que se pretende atingir.

A norma G.711 do CCITT também especifica outro método de quantização logarítmica similar ao método μ-law, chamado de A-law. Este método é adotado pelo sistema europeu de telefonia[2].

3.1.2 DPCM

Amostras adjacentes de áudio são correlacionadas. No caso de sinais de voz, essa condição é válida em intervalos contendo diversas amostras[2, 7]. Esta correlação fornece subsídios para a predição da amostra a ser quantizada, e a quantização da diferença entre o valor real e o predito, ao invés da quantização da amostra, como no caso do sistema PCM. Assim, tem-se o sistema descrito na figura 3.2, conhecido como Differential Pulse Code Modulation (DPCM, ou modulação diferencial por código de pulso):

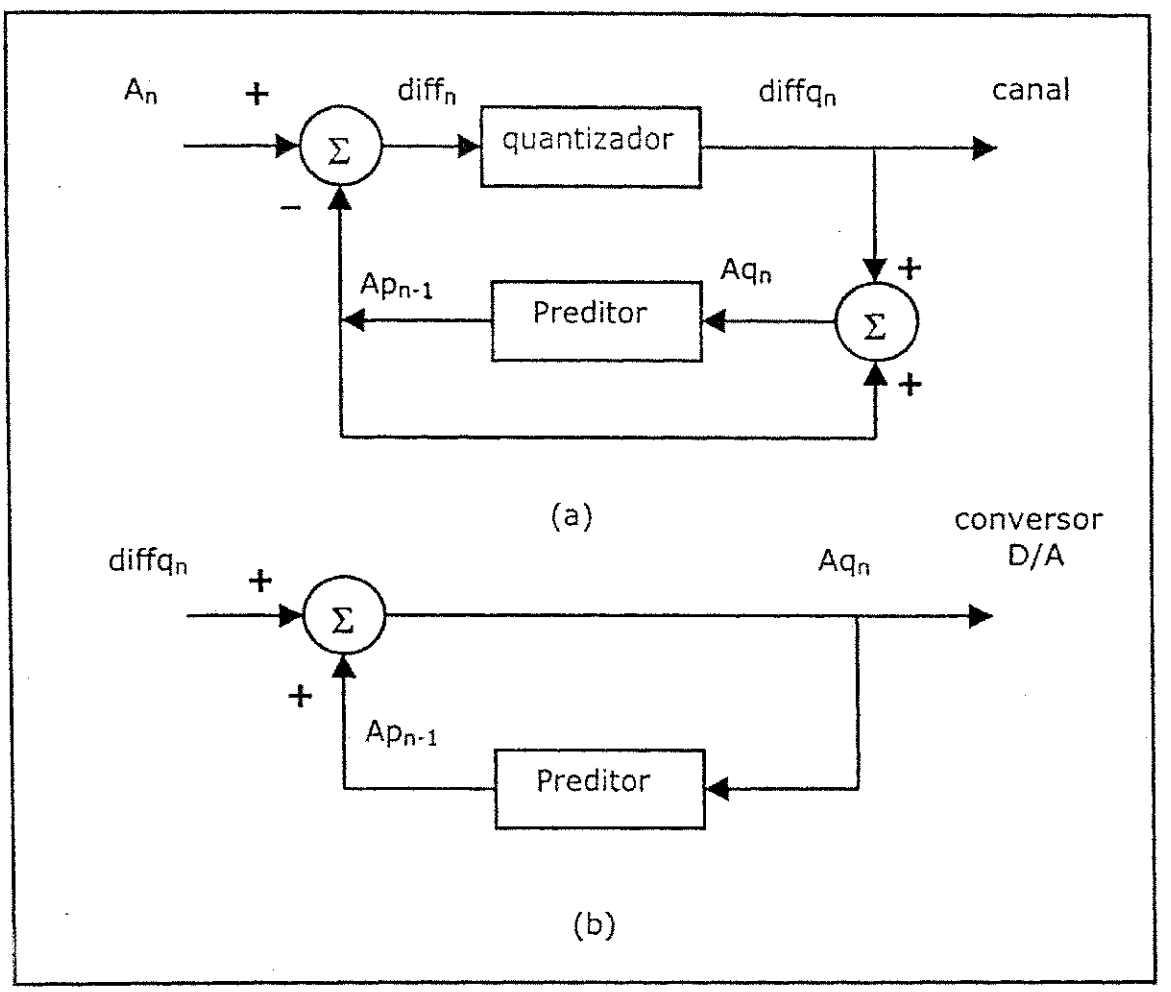


Figura 3.2 Método DPCM: (a) codificador e (b) decodificador. A_n é a amostra atual Ap_{n-1} é o valor predito para a amostra atual, calculado ao longo da quantização da amostra anterior, sendo $diff_n$ o valor da diferença entre A_n e Ap_{n-1}; $diffq_n$ é o valor atribuído pelo quantizador à diferença $diff_n$, ou seja, $diffq_n = diff_n + eq_n$, onde eq_n é o erro de quantização da diferença $diff_n$, de forma que $Aq_n = A_n + eq_n$.

O preditor pode utilizar predição linear, calculando o valor predito a partir da combinação linear dos valores quantizados anteriormente. A ordem do preditor é o número de coeficientes, ou seja, o número de valores anteriores armazenados para o cálculo do valor predito atual.

Espera-se, a partir da predição, que a faixa dinâmica da *diferença* entre o sinal real e o sinal predito seja menor que a faixa dinâmica do sinal, portanto requerendo menos bits para sua representação. Outra vantagem em relação ao sistema PCM é a menor degradação do sinal recuperado, quando da presença de bits errôneos, já que apenas a diferença entre amostras é transmitida, ao invés das amostras propriamente ditas.

A complexidade da síntese do algoritmo DPCM aumenta com a ordem do preditor. Além disso, os coeficientes do preditor podem variar com o tempo, para que a predição se adapte dinamicamente às características do sinal de áudio. No caso dos coeficientes variáveis, o sistema é dito DPCM com predição adaptativa, enquanto no caso de coeficientes fixos, o sistema é dito DPCM com predição fixa.

Um estudo quantitativo foi feito por NOLL [9] avaliando a melhoria de desempenho do sistema DPCM com predição fixa em relação ao sistema PCM, para sinais de voz humana. Os resultados encontram-se na figura 3.3:

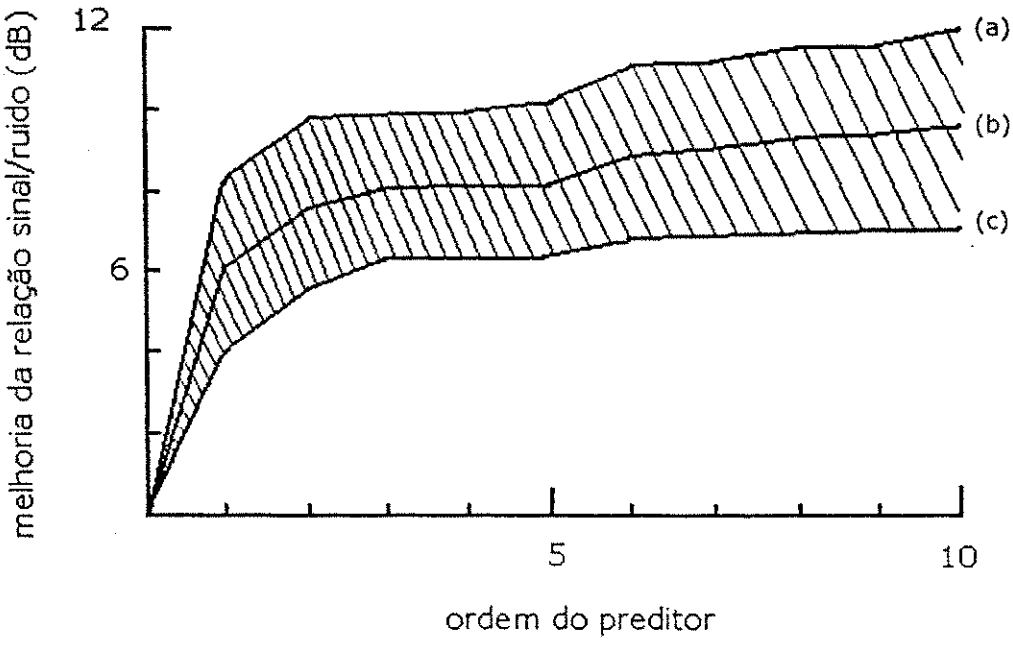


Figura 3.3 Gráfico da melhoria da relação sinal-ruído em função da ordem do preditor.

A área sombreada, no gráfico da figura 3.3, é limitada superiormente pela curva (a) e inferiormente pela curva (c), como resultado da quantização de diferentes sinais, como vozes de homens, crianças, mulheres, etc. A curva (b) a média aritmética entre os dois extremos. O gráfico ainda mostra que a maior diferença de desempenho acontece quando se parte de um sistema sem predição(PCM) para o sistema DPCM com preditor de ordem 1. Além disso, a melhoria a partir da ordem de predição 2 é bastante suave, apesar do acréscimo de complexidade do algoritmo.

3.1.3 PCM adaptativo

Os sistemas PCM e DPCM podem ainda exibir melhor desempenho se forem implementados a partir de quantizadores adaptativos. Desta forma, o passo de quantização, ou seja, o espaço entre duas palavras adjacentes, varia dinamicamente na tentativa de se adequar às características do sinal de áudio.

Existem dois métodos de adaptação: o feed-forward e o feedback. A figura 3.4 mostra o método PCM adaptativo feed-forward:

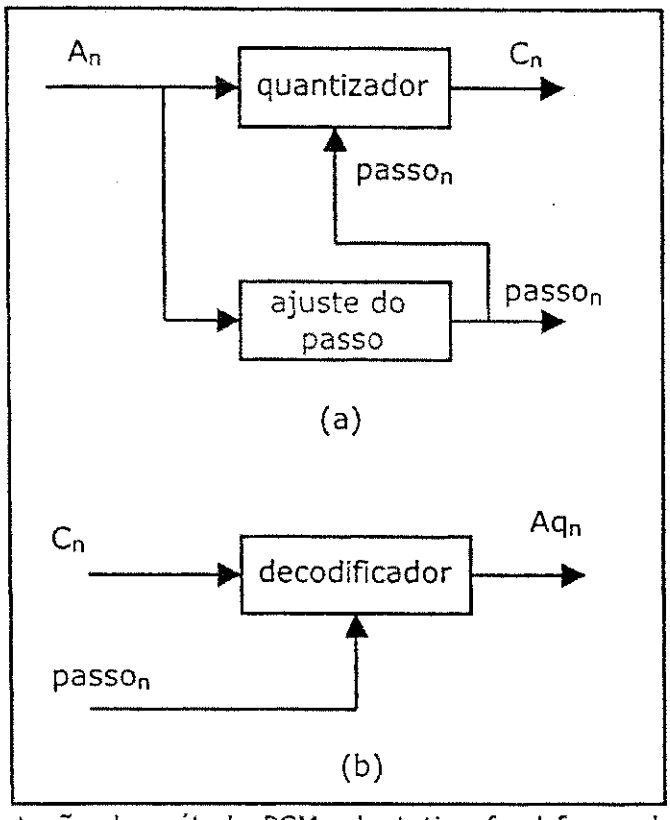


Figura 3.4 Implementação do método PCM adaptativo feed-forward. (a)codificador e (b) decodificador. Neste método, o valor do passo de quantização, $passo_n$, é ajustado de acordo com o valor da amostra, devendo este ser transmitido em conjunto com a palavra código resultante da quantização, C_n . No decodificador, a partir do código C_n e do valor do passo, $passo_n$, recupera-se Aq_n , sendo $Aq_n = A_n + eq_n$, onde eq_n é o erro introduzido na quantização.

Neste método, o tamanho do passo de quantização varia de acordo com o nível do sinal de entrada. Tanto a palavra de quantização quanto o valor do passo devem ser transmitidos ao receptor (decodificador). Se houver erro na palavra recebida ou no valor do passo recebido, a amostra recuperada no receptor não será igual à transmitida. O efeito de tais erros varia de acordo com o tipo de implementação do método[7].

A figura 3.5 mostra o método PCM adaptativo feed-back:

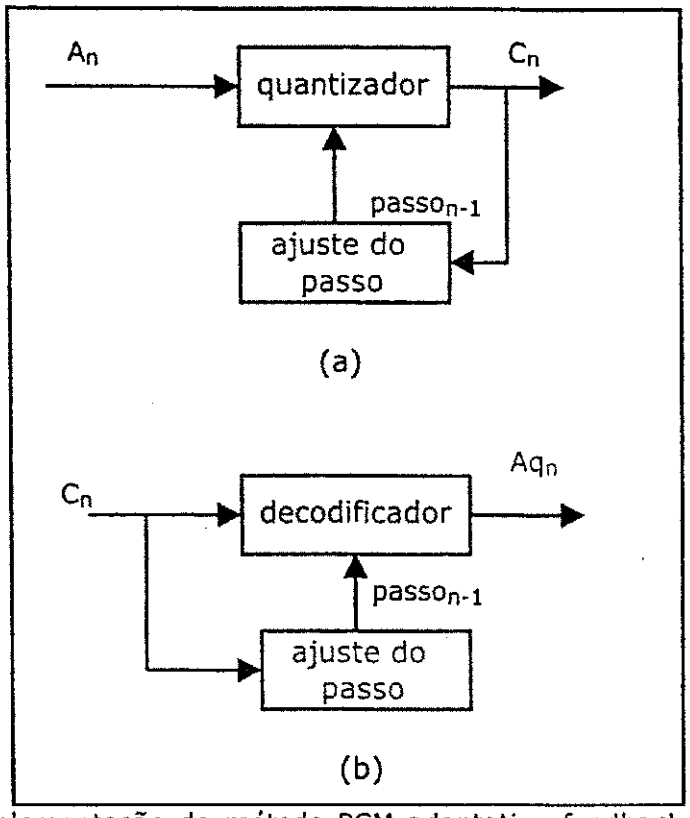


Figura 3.5 Implementação do método PCM adaptativo feedback. (a)codificador e (b) decodificador. O valor do passo de quantização, $passo_n$, para a amostra atual, A_n ,é ajustado de acordo com o resultado da quantização, C_n, da amostra anterior, de forma que não é necessária a transmissão do valor do passo, já que este é decorrente das palavras quantizadas C_n recebidas pelo decodificador. $A_n = Aq_n + eq_n$, onde eq_n é o erro de quantização.

Neste método, o valor do passo de quantização é ajustado de acordo com a palavra na saída do quantizador. Desta forma, torna-se desnecessária a transmissão do valor do passo, já que este é decorrente da palavra recebida no receptor. A vantagem deste esquema é a redução da taxa de bits, já que se economiza a transmissão do valor do passo. Em contrapartida, um erro na palavra recebida resulta não somente na recuperação errônea da amostra, mas também no desvio do valor do passo[7].

Existem diversas formas de implementação do mecanismo de adaptação do passo de quantização. O método proposto por Jayant[12] obedece à seguinte equação :

$$passo_n = P \times passo_{n-1} \qquad (3.2)$$

onde:

$passo_n$ é o valor do passo a ser utilizado na amostra atual

$passo_{n-1}$ é o valor do passo utilizado na última amostra

P é uma constante de multiplicação que depende da magnitude da última palavra quantizada.

Jayant [12] buscou encontrar um conjunto de constantes de multiplicação que minimizasse o valor médio quadrático do erro de quantização. Encontrou resultados teóricos para sinais Gaussianos, e obteve resultados para sinais de voz através de métodos comparativos. Os resultados são descritos no gráfico da figura 3.6 :

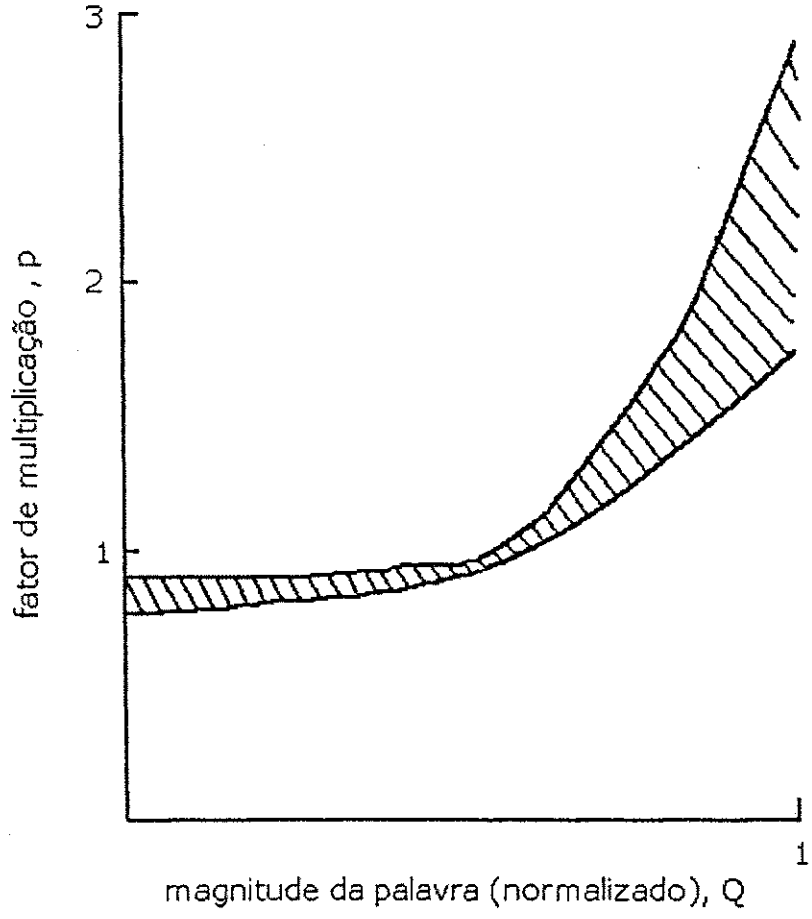


Figura 3.6 Fatores de multiplicação do passo de quantização em função da palavra quantizada. Percebe-se que a região de aumento do passo (P>1) deve ser bem mais abrupta que a região de diminuição do passo(P<1). A área sombreada indica que foram encontrados valores diferentes de P, de acordo com o sinal analisado (voz de homens, mulheres,etc.)

A tabela 01 mostra os valores da constante de multiplicação para quantizadores com 2,3,4 e 5 bits :

bits do quantizador	Fatores Multiplicativos
2	0.6 , 2.2
3	0.85 , 1 , 1 , 1.5
4	0.8 , 0.8 , 0.8 , 0.8 1.2 , 1.6 , 2.0 , 2.4
5	0.85 ; 0.85 , 0.85 , 0.85 0.85 ; 0.85 , 0.85 , 0.85 1.2 , 1.4 , 1.6 , 1.8 2.0 , 2.2 , 2.4 , 2.6

Tabela 01. Fatores multiplicativos do passo de quantização para quantizadores PCM adaptativos de 2, 3, 4 e 5 bits[12].

Em um outro estudo com utilização de quantizadores adaptativos de 3 bits, Noll[24] encontrou similar melhoria da relação sinal-ruído utilizando as constantes {0.8, 0.8, 1.3, 1.9}, em contraste com as sugeridas na tabela acima {0.85, 1.0, 1.0, 1.5}. Isto leva a crer que os valores específicos dos multiplicadores não são críticos[7].

3.1.4 DPCM adaptativo (ADPCM)

Utilizando quantização adaptativa (descrita acima para o método PCM adaptativo) em quantizadores diferenciais (DPCM), chega-se à técnica ADPCM.

A associação IMA[18] propôs um algoritmo de baixa complexidade para quantização DPCM adaptativa. O valor predito para a amostra atual é o valor da última quantização, de forma que o preditor mostrado na figura 3.2, para o caso do método DPCM, é apenas uma célula de atraso (registrador).

A figura 3.7 mostra o diagrama em blocos do compressor de áudio ADPCM proposto pela associação IMA [18]:

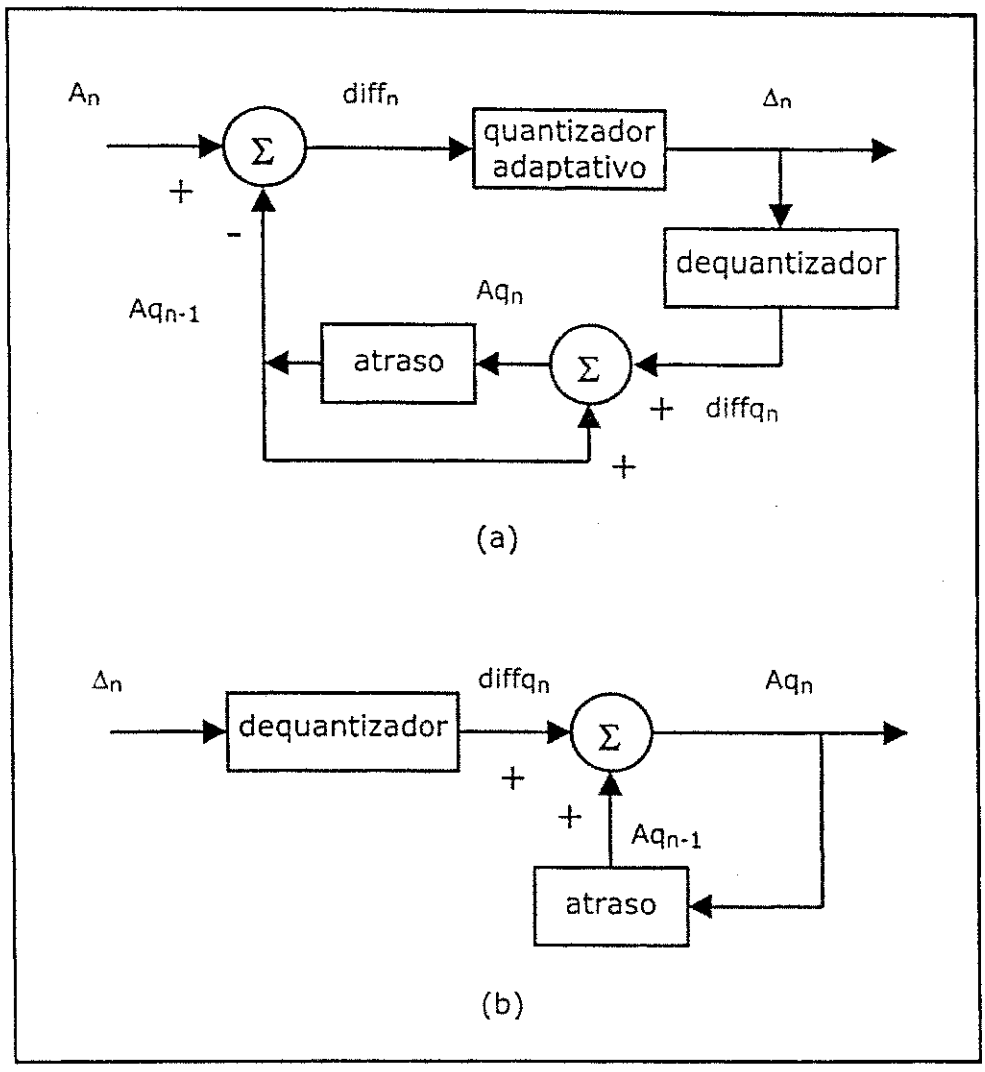


Figura 3.7 Método DPCM adaptativo proposto pela associação IMA:(a) codificador e (b) decodificador. $diff_n$ é a diferença entre a amostra atual A_n e o valor predito para a amostra atual, Aq_{n-1}, que é simplesmente o valor quantizado na última operação ; o decodificador utiliza blocos funcionais iguais aos do codificador.

O valor da última amostra quantizada, presente no registro de atraso, é subtraído da amostra atual de forma a obter-se a diferença entre amostras, chamada na figura 3.7 de $diff_n$. O bloco quantizador adaptativo divide esta diferença pelo passo de quantização para formar a palavra código a ser enviada (Δ_n).

A característica do algoritmo de adaptação às características do sinal sendo quantizado consiste no ajuste do passo de quantização. O ajuste do passo é realizado com base na palavra código resultante da quantização da amostra atual, como mostra a figura 3.5.

A cada palavra código, Δ_n, é associada uma constante de multiplicação para o ajuste do passo.

Estas constantes de multiplicação são próximas às encontradas no estudo de Jayant[12], descritas na tabela 02, sendo o valor do passo limitado tanto inferiormente quanto superiormente.

Bits do quantizador	Fatores Multiplicativos
2	0.8 , 1.6
3	0.9 , 0.9 , 1.25 , 1.75
4	0.9 , 0.9 , 0.9 , 0.9 1.2 , 1.4 , 2.0 , 2.4
5	0.9 , 0.9 , 0.9 , 0.9 0.95 , 0.95 , 0.95 , 0.95 1.2 , 1.5 , 1.8 , 2.1 2.4 , 2.7 , 3.0 , 3.3

Tabela 02. Fatores multiplicativos do passo de quantização para quantizadores diferenciais adaptativos de 2, 3, 4 e 5 bits[12].

Esta implementação permite um bom desempenho tanto na presença de erros inseridos pelo canal de comunicação quanto na falta de sincronização transmissor/receptor inicial, na ocasião em que o receptor é ligado, já que a informação atual do passo de quantização no transmissor não é transmitida. Entenda-se por falta de sincronismo transmissor/receptor o fato do valor, no receptor, de Aq_n , ou Δ_n , ou ambos serem diferentes destes valores no transmissor.

Nestes casos, o erro provoca um nível DC no sinal recuperado podendo provocar ainda diferença de amplitude pico a pico do sinal recuperado em relação ao sinal original. O nível DC será imperceptível, a não ser que o limite inferior ou superior do decodificador seja atingido, o que servirá para corrigir parcial ou totalmente o valor do nível DC. Além disso, um breve intervalo com amostras de pequena amplitude levará o valor do passo necessariamente ao limite inferior, de forma a sincronizar transmissor e receptor [2].

Testes subjetivos de desempenho demonstram que o sistema ADPCM utilizando quantizador de 4 bits adaptativo é mais bem aceito que o sistema PCM logarítmico quantizado com 6 bits[7].

A versão em software do algoritmo IMA ADPCM utiliza busca de valores de passo em tabela, de forma a tornar necessária a alocação de memória de leitura (ROM) na síntese em hardware. Uma solução alternativa seria a implementação das constantes de multiplicação através de lógica combinacional. Em particular, os valores de multiplicação 0.875, 1.25, 1.5, 2.0 e 3.0 podem ser implementados a partir de deslocamentos e adição/subtração de operandos.

3.2. Domínio da frequência

Em geral, os compressores de áudio que utilizam análise espectral oferecem melhor eficiência de compressão em relação aos compressores já vistos, que se restringem ao tratamento das amostras apenas no domínio do tempo. O preço pago por tal desempenho é o aumento de complexidade dos algoritmos que por sua vez se reflete em aumento de área necessária para síntese em circuito integrado e aumento do consumo de energia elétrica [19,20].

DFT (Discrete Fourier Transform, transformada discreta de Fourier), FFT (Fast Fourier Transform, ou transformada rápida de Fourier) e DCT (Discrete Cossine Transform, ou transformada discreta de co-senos) são os métodos mais utilizados na conversão de domínios tempo-frequência. Necessitam de um poderoso recurso computacional para execução de multiplicações, além da alocação de memória de acesso aleatório (RAM) considerável.

3.2.1 MPEG-Áudio

A psico-acústica e o fenômeno do mascaramento de frequências têm sido bastante estudados na busca de compressores de áudio de maior eficiência. A membrana basilar, interna ao ouvido humano, é responsável pela distinção de frequências. Esta membrana possui várias frequências de ressonância ao longo de sua extensão, comumente chamadas de bandas críticas.

Como trata-se de um sistema mecânico, na presença de um forte tom em uma frequência que ressona em uma área que chamemos de Ax, a acentuada vibração em Ax faz com que as áreas vizinhas, que chamemos de Ax+1 e Ax-1, também vibrem nesta frequência, de forma a prejudicar a percepção de possíveis sinais nas bandas adjacentes A_{x+1} e A_{x-1}, a não ser que o tom nestas bandas tenha amplitude suficiente para vencer a ressonância vizinha . A este efeito dá-se o nome de mascaramento[2].

No sistema MPEG-áudio, o sinal de áudio a ser comprimido passa por um banco de filtros, para verificação de suas componentes espectrais dentro das bandas críticas. Estas componentes são então entregues a um modelador de mascaramento que se propõe a distinguir as frequências que serão efetivamente percebidas pelo ouvinte, daquelas que serão imperceptíveis.

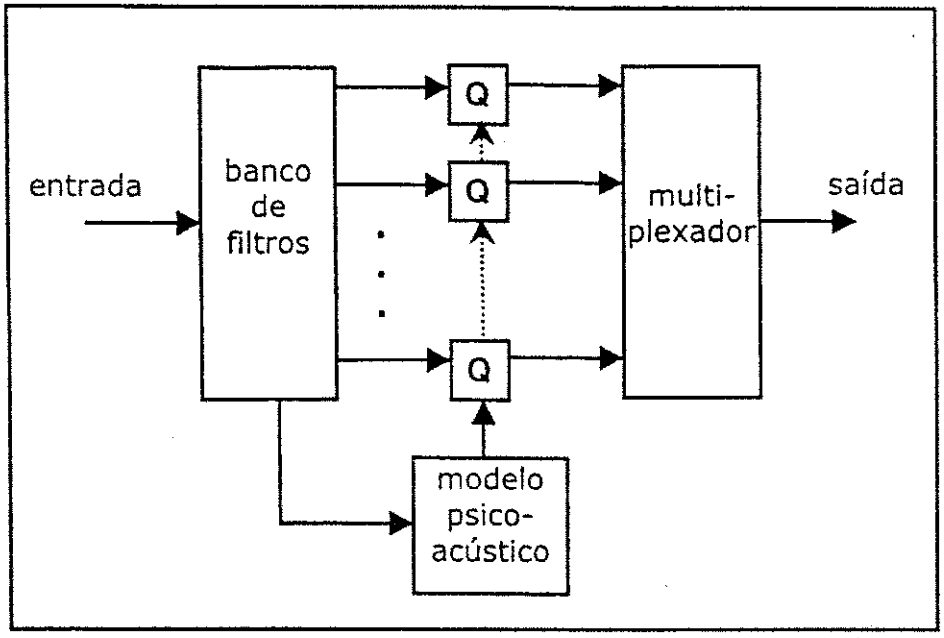


Figura 3.8 Esquema de compressor que utiliza modelamento psico-acústico

Em seguida, a alocação de bits para cada banda é feita de acordo com o resultado do modelador de mascaramento, alocando mais bits para as frequências de maior importância, como mostra a figura 3.8.

O compressor de áudio proposto pelo grupo MPEG[13] baseia-se no fenômeno do mascaramento. Trata-se de um compressor de alta complexidade e alto desempenho. Testes subjetivos demonstram que mesmo utilizando uma taxa de compressão de 6/1, onde o sinal de áudio estéreo amostrado a 44.8 KHz e quantizado com conversores de 16 bits foi comprimido a uma taxa de 256 Kbits/s, ouvintes especializados foram incapazes de distinguir entre o sinal original e o sinal comprimido[2].

3.2.2 Vocoders

Os vocoders formam uma classe especial de compressores que se destinam exclusivamente ao tratamento do sinal de voz. Baseiam-se nos estudos desenvolvidos a respeito do sistema vocal humano. O sistema vocal é modelado como um filtro excitado a partir de um trem de pulsos cujo período equivale ao período do pitch de voz[8].

Os parâmetros do filtro são estimados a partir do sinal de voz recebido, juntamente com o período do pitch, e enviados ao receptor que irá recuperar o sinal de voz a partir da síntese do filtro modelado.

No caso de sinais de voz, os vocoders oferecem uma maior taxa de compressão em relação aos sistemas usuais de quantização (PCM, DPCM, μ-law).

Resumo do capítulo

Os compressores de áudio que utilizam métodos de transformadas de domínios tempo/frequência requerem uma alta complexidade, fazendo com que sejam impraticáveis em sistemas de baixo custo. Os compressores de áudio que trabalham exclusivamente no domínio do tempo exibem menor desempenho em relação aos supracitados, mas podem ser satisfatórios de acordo com a aplicação. Os quantizadores adaptativos demonstram melhor performance que os quantizadores e dependendo da forma de implementação, podem resultar num circuito de baixo consumo e pequena área de silício.

4. Modulação Digital

Uma vez definido o fluxo de bits necessário à transmissão de sinal de áudio estéreo de qualidade desejável, é preciso que o sinal digital passe por um processo de pré-modulação, a fim de se adequar aos requisitos de largura de banda do canal de transmissão. Uma vez feito isto, torna-se possível a utilização de transmissores/receptores faixa-larga de baixa complexidade, que respondam a sinais de até 300KHz, facilmente encontrados no mercado[15] para que sistema final seja de baixo custo.

As transições abruptas do sinal digital original fazem com que este exiba componentes harmônicos de magnitude considerável, bem além do valor de 1Hertz/bit/s, ou seja, um sinal digital com níveis de tensão 0 e 1 e taxa de 100Kbits/s ocupa uma banda que se estende além do valor de 100KHz. Através da pré-modulação do sinal digital, pode-se conseguir um sinal mais compacto em termos de componentes espectrais e que seja detectável de maneira simples e eficaz no receptor.

Existem várias técnicas de modulação digital, cada qual exibindo variados níveis de satisfação aos seguintes requisitos:

- ➔ Aproveitamento de banda (maior taxa de bits alocada)
- ➔ Imunidade ao ruído
- ➔ Potência de transmissão necessária
- ➔ Complexidade do par Modulador/ Demodulador

As modulações digitais equivalentes às analógicas AM, PM e FM são conhecidas como ASK (Amplitude Shift Keying, ou modulação por chaveamento de amplitude), PSK (Phase Shift Keying, ou modulação por chaveamento de fase) e FSK(Frequency Shift Keying, ou modulação por chaveamento de frequência).

4.1 ASK

A figura 4.1 mostra um sinal senoidal modulado por chaveamento de amplitude:

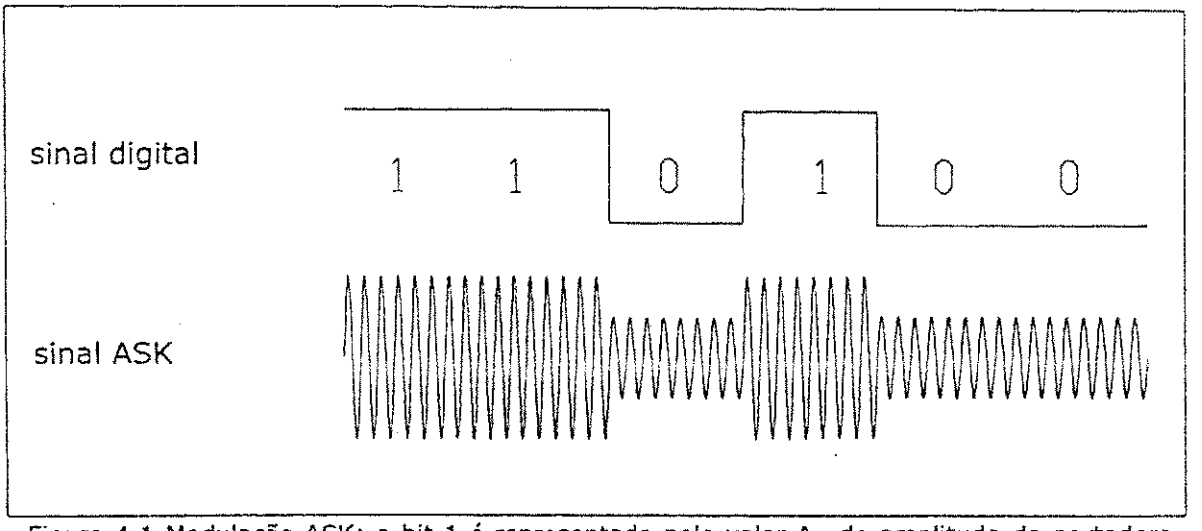


Figura 4.1 Modulação ASK; o bit 1 é representado pelo valor A_1 de amplitude da portadora, enquanto o nível 0 é representado pelo valor A_2 de amplitude da portadora, onde $A_1 > A_2$.

Trata-se de uma modulação bastante simples, podendo ser implementada a partir de um amplificador com ganho variável (modulador) e um detector de envoltória seguido de um comparador de tensão (demodulador). Conjuntos de m bits poderiam ser agrupados formando assim 2^m diferentes níveis de amplitude, definindo assim uma modulação multi-nível.

Alguns modems utilizam modulação por chaveamento de amplitude com os valores m=2, 4,5 e 7, podendo ainda ser encontrado o valor m=10 [16].

A figura 4.2 contém um exemplo de um sinal resultante da modulação ASK com 4 níveis distintos de amplitude, ou 4-ASK :

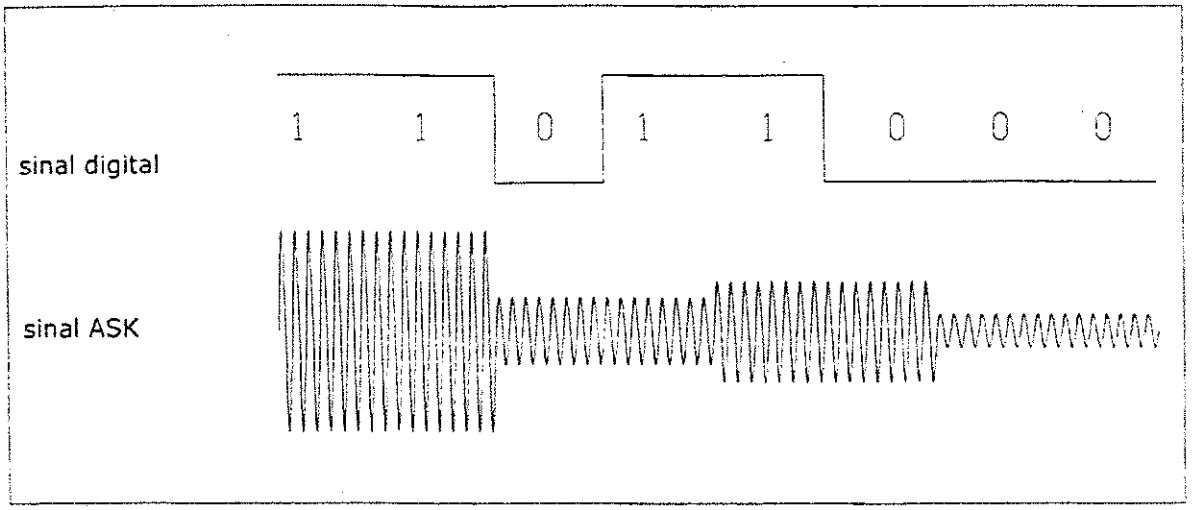


Figura 4.2 Modulação 4-ASK; Agrupamentos de dois bits modulados através de 4 níveis distintos de amplitude da portadora. Desta forma, a variação de amplitude se dá a uma taxa igual à metade da taxa de bits, requerendo assim uma menor banda para transmissão.

Por apresentar baixa imunidade ao ruído aditivo, além de ser sensível à não-linearidade dos transmissores, tornando inviável sua utilização com amplificadores classe C, notoriamente mais eficientes, esta modulação é raramente utilizada nas aplicações de transmissão via Ondas Eletromagnéticas[6,11].

4.2 PSK

A figura 4.3 mostra um sinal senoidal modulado por chaveamento de fase:

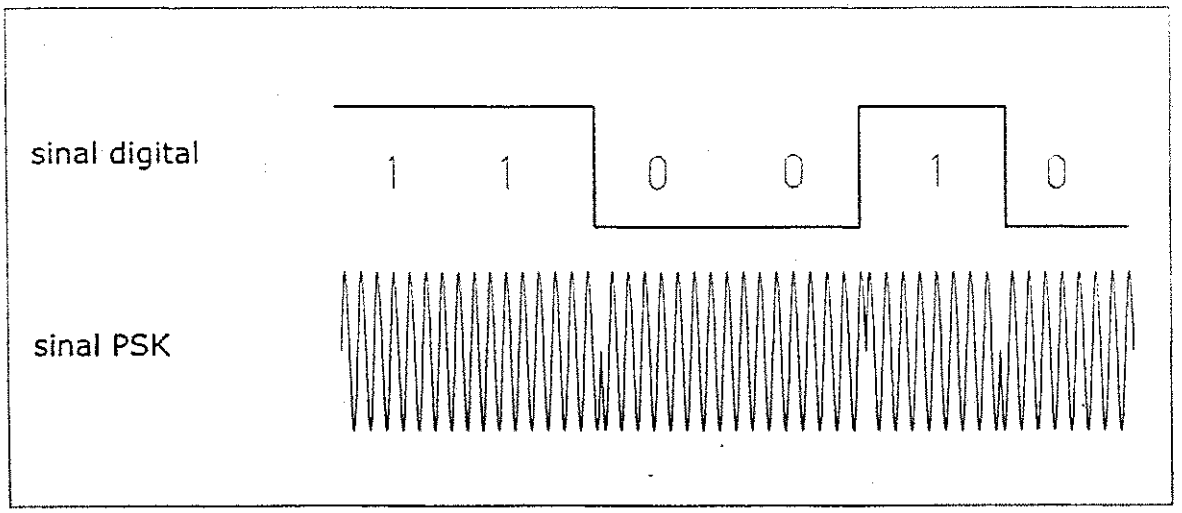


Figura 4.3 Modulação PSK; o bit 0 é representado por uma onda senoidal com fase nula, enquanto o bit 1 é representado por uma onda senoidal com fase 180°.

Dentre as 3 técnicas de modulação citadas anteriormente (ASK, FSK e PSK), esta modulação propicia a menor taxa de erro de bit em condições

similares de comparação. O processo de demodulação é coerente, ou seja, o demodulador necessita da informação da fase do sinal modulado para uma perfeita recuperação dos bits. Isto requer a sincronização modulador/demodulador, fazendo com que a complexidade do sistema aumente.

Uma maneira de contornar o problema da demodulação coerente é fazer com que a informação do sinal digital seja enviado nas mudanças de fase, e não no valor da fase propriamente dito, esquema conhecido como DPSK (Differential Phase Shift Keying, ou modulação diferencial por chaveamento de fase) [11]. Um exemplo desta técnica de modulação consiste em associar a cada bit 1 (nível lógico alto) do sinal digital, uma variação de fase da onda senoidal de 180°, enquanto que não há variação de fase da onda senoidal para qualquer bit 0 (nível lógico baixo) do sinal digital.

Para a modulação PSK, pode-se separar a sequência de bits do sinal digital, a ser modulado, em duas novas sequências: a sequência dos bits de índice n (bits pares) e a sequência dos bits de índice n+1 (bits ímpares). Estas novas duas sequências apresentam transições entre bits adjacentes a uma taxa igual à metade da sequência original. Fazendo-se com que as novas sequências modulem a fase de sinais em quadratura (seno(x) e co-seno(x)), e somando-se os sinais resultantes, obtém-se um sinal QPSK (Quadrature Phase Shift Keying). O método QPSK apresenta uma maior eficiência de ocupação de banda, com a contrapartida de aumento de complexidade do circuito.

4.3 FSK

A figura 4.4 mostra um sinal senoidal modulado por chaveamento de frequência:

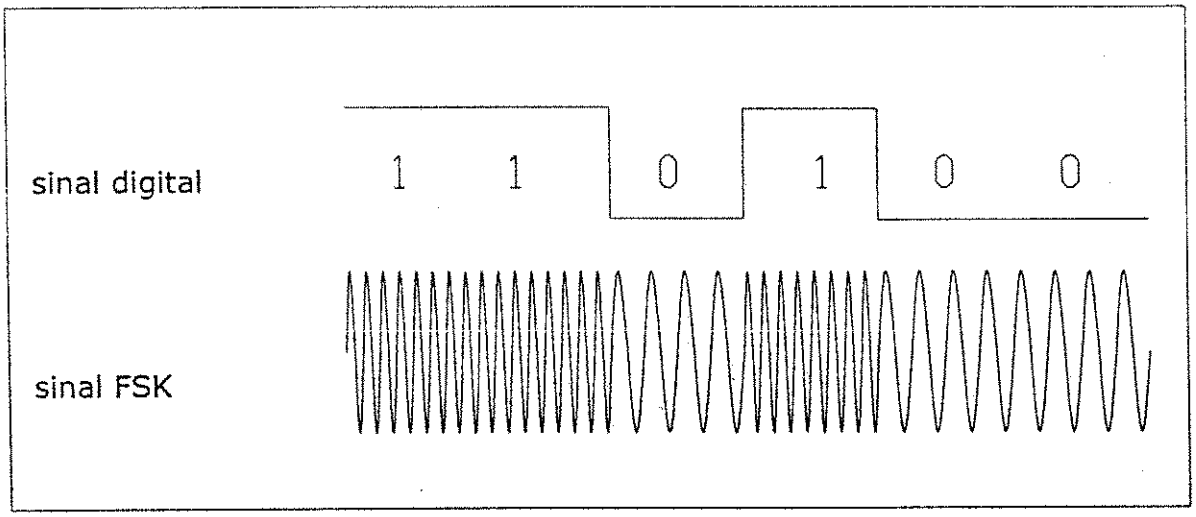


Figura 4.4 Modulação FSK; o bit 0 é representado pela frequência f_1, enquanto o bit 1 é representado pela frequência f_2, onde $f_2 > f_1$

O desempenho desta modulação em termos de taxa de erro de bits se aproxima do que é conseguido através da modulação PSK, podendo ser implementada com uma complexidade relativamente menor. Pode-se usar um método não coerente de demodulação a partir de filtros passa-faixa e detectores de envoltória. A baixa complexidade desta técnica aliada à característica de amplitude constante da portadora fazem com que esta técnica seja amplamente utilizada na transmissão de sinais digitais.

4.4 MSK

A modulação MSK (Minimum Shift Keying) encontra-se definida de várias maneiras. Trata-se de um método de modulação com fase contínua onde a variação da frequência da portadora ocorre sempre na passagem por zero do sinal modulado, ou seja, na transição do bit *n* para o bit *n+1* do sinal digital, há sempre uma passagem por zero do sinal modulado.

Uma característica peculiar do sinal MSK é a relação entre as frequências atribuídas aos bits 1 e 0: a diferença entre a frequência associada ao bit 1 e a frequência associada ao bit zero é sempre igual à metade da taxa de bits. O índice de modulação é dado pela equação 4.1:

$$M = \Delta f \times T \qquad (4.1)$$

onde :
- M é o índice de modulação
- Δf é a diferença entre a frequência do bit 1 e a frequência do bit 0
- T é o tempo de bit

Desta forma, como o valor da diferença entre frequências é sempre a metade do valor da taxa de bits, $\Delta f = | f_1 - f_0 | = \frac{1}{2} \times {}^{1}/_{T} = {}^{1}/_{2T}$, de forma que o índice de modulação do sinal MSK é sempre 0.5 [19].

A figura 4.5 mostra um exemplo de sinal MSK, representando um sinal digital com taxa de 1000bits/s, sendo composto a partir de um sinal senoidal com frequência de 1000Hz representando o bit 0 e um sinal senoidal com frequência de 1500Hz representando o bit 1:

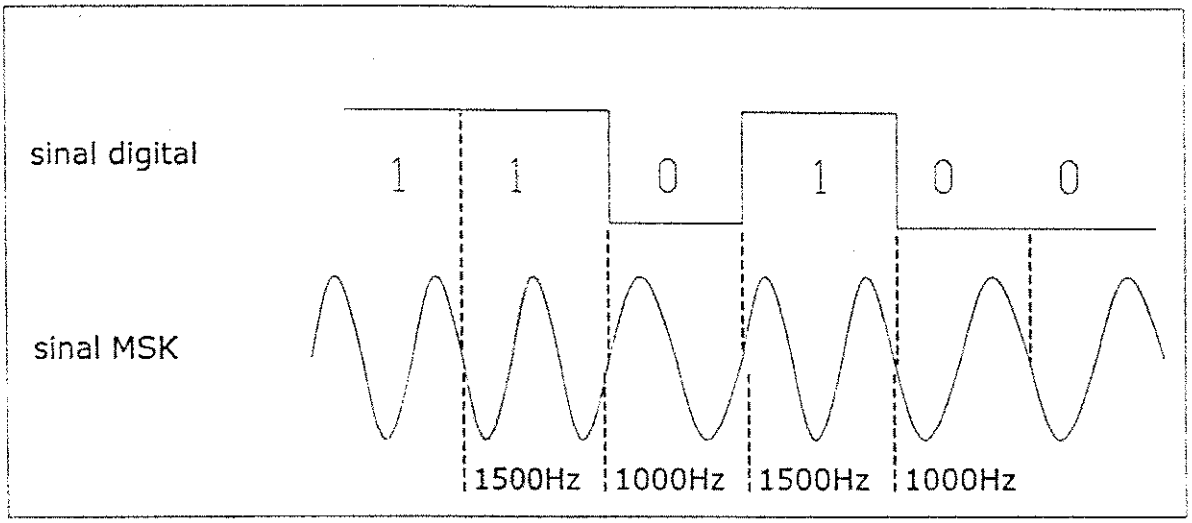


Figura 4.5 Modulação MSK; o bit 0 é representado por um ciclo completo de frequência 1000Hz, enquanto o nível 1 é representado por um ciclo e meio de frequência 1500Hz. Nota-se a ausência de descontinuidade de fase. O valor da frequência que representa o bit 0 é numericamente igual ao valor da taxa de bits, no caso, 1000bits/s , enquanto que o valor da frequência que representa o bit 1 é numericamente igual a 1.5 vezes o valor da taxa de bits.

O sinal da figura 4.5 pode ser escrito da seguinte maneira[11]:

$$X_{MSK}(t) = A_C \times \cos\left[\omega_C t + \int_{-\infty}^{t} \sum_{m} b_m \times p(t - mT)dt\right] \qquad (4.2)$$

onde :

$X_{MSK}(t)$ é o sinal MSK;

A_C é a amplitude de pico do sinal MSK;

ω_C é a frequência da portadora;

b_m é 1 se o bit é 1, e -1 se o bit é 0;

p(t) é um pulso retangular.

O somatório em m representa o sinal digital , de forma que a fase no sinal MSK é a integral do sinal digital, podendo assim ser vista como oriunda do sinal FSK.

Grapes [17] utiliza a modulação MSK em um modem de alta performance com transmissão de taxa de 56KBPS (kilo baud por segundo).

A implementação proposta por Dale A. Heatherington [17] forma o sinal MSK a partir de pontos armazenados em memória, uma máquina de estados e um conversor D/A, podendo assim ser utilizada em aplicações de baixa complexidade.

Esta técnica é particularmente interessante pela baixa complexidade do demodulador: na figura 4.5, nota-se que o período do semiciclo do bit 0 é 1.5 vezes maior que o período do semiciclo do bit 1. Definido um limiar de escolha, como o valor intermediário entre os dois valores de período, por exemplo, o demodulador define o bit recuperado de acordo com o período do semiciclo recebido. Pode-se implementar o demodulador sem a necessidade de um oscilador senoidal local, a partir de um sinal de clock de referência, um comparador de tensão e um contador síncrono, de forma que o circuito analógico necessário à demodulação é reduzido.

A figura 4.5 exemplifica uma das formas do sinal MSK. Variantes deste sinal podem ser encontradas exibindo diferentes índices de modulação, assim como também utilizando intervalos de bit diferentes. Cada uma destas variantes tem características próprias de ocupação de espectro e complexidade de demodulação.

Um melhor aproveitamento da banda do canal de comunicação é conseguido com o método de modulação GMSK, ou Gaussian Minimum Shift Keying, uma derivação do método MSK, com a inserção de um filtro gaussiano para a filtragem do sinal digital, ou seja, precedendo a pré-modulação MSK. Esta filtragem resulta na interferência intersimbólica, ou seja, interferência entre bits adjacentes quando da demodulação, de forma a requerem uma maior complexidade no receptor [19].

Resumo do capítulo

Neste capítulo foram vistos vários tipos de modulação digital. Pela característica de amplitude da portadora constante, os métodos PSK, FSK e MSK apresentam melhor imunidade ao ruído que a técnica ASK, além de tornar viável o uso de amplificadores não-lineares (de superior eficiência de potência de transmissão), apesar desta não ser uma questão preocupante no caso da radiação restrita, onde a potência envolvida é pequena. O método MSK, derivado do método FSK, fazendo-se o índice de modulação igual ao valor 0.5, é particularmente interessante, pois necessita de circuito de baixa complexidade, na sua grande parte, digital, além de oferecer desempenho similar às técnicas apresentadas.

5. Sistema Proposto

5.1 Desempenho desejado

O intuito do projeto maior onde está inserido este trabalho é a concepção de um par de circuitos integrados, transmissor e receptor, visando baixos custo e consumo e apresentando qualidade satisfatória para funções secundárias de entretenimento, ou seja, para sistemas que não requerem alta fidelidade.

A necessidade de baixos custo e consumo do circuito final objetiva a aplicação em sistemas portáteis de grande mercado de consumo, como por exemplo na utilização em fones de ouvido sem fio para transmissão de sinal de áudio de tv. O valor de resposta em freqüência de áudio na faixa de 50-7000 Hz vem sendo utilizado nestes tipos de aplicação, por ser satisfatória para tais sistemas[21]. Adotar-se-á neste projeto este valor de resposta em frequência como base para aferição de desempenho.

Como visto no capítulo 2, para se obter uma resposta em frequência de até 7 KHz é necessário que o sinal de áudio seja amostrado a uma taxa igual ou maior que o dobro de sua maior freqüência, no caso 7 KHz, de forma que a taxa de amostragem deve ser maior ou igual a 14 KHz. Foi definida para este projeto a taxa de amostragem de 20 KHz, de forma a haver uma boa margem de sobre-amostragem, reduzindo, pois, os requisitos de desempenho dos filtros de anti-aliasing e de reconstrução do sinal de áudio no receptor.

Para que se consiga uma relação sinal-ruído maior que 60 dB após o processo de quantização, será previsto o uso de conversores analógico-digitais de 12 bits.

Além disso, o projeto do sistema deve prever o controle do receptor a partir do transmissor. Isto é feito através do envio de palavras de controle embutidas na transmissão de áudio, de forma que a qualidade do som não seja prejudicada.

5.2 Processo de concepção

Um processo bastante utilizado, por sua comprovada eficácia na concepção de circuitos integrados, sugere que o sistema proposto seja primeiro implementado, se possível, a partir de linguagem de alto nível. Desta forma, pode-se analisar a performance de tal sistema de forma rápida e simples, descrevendo-o a partir de uma linguagem de fácil acesso e interface amigável.

Neste trabalho, as linguagens C e MATLAB foram utilizadas para avaliação inicial do sistema proposto. Os resultados de simulações executadas estão descritos nas subseções intituladas *Simulações*.

Em seguida, o sistema proposto foi descrito com o uso da linguagem Verilog HDL (Verilog Hardware Description Language, ou linguagem Verilog de descrição de hardware). A funcionalidade esperada foi descrita de várias maneiras, a fim de que se pudesse avaliar os recursos necessários em termos de área e velocidade de processamento, quando da implementação em silício.

A ferramenta MAX-PLUSII da ALTERA[22] para síntese de FPGAS foi utilizada como base para a comparação entre os vários códigos de descrição de hardware feitos. Esta ferramenta sintetiza o circuito desejado a partir de estruturas básicas configuráveis chamadas de macro-células. Após a compilação do código, a ferramenta cria um arquivo texto onde consta o número de macro-células necessárias à síntese do referido circuito. Foi feita a comparação dos circuitos descritos através do número de macro-células requeridos à síntese de cada um.

Uma das preocupações ao longo da concepção dos circuitos foi a utilização, quando possível, de componentes com valores de mercado, como no caso da escolha do número de bits das conversões A/D e D/A, além de buscar a utilização de componentes comuns para transmissor e receptor.

6. Compressor de áudio

6.1 Algoritmo escolhido

Um dos fatores que ditam o consumo de um circuito digital é a sua complexidade. Pode-se geralmente associar o termo *complexidade de um circuito digital* ao número de transistores que possui. Quanto maior este número, provavelmente maior será o número de chaveamentos ocorridos num ciclo de máquina, o que resulta no aumento de potência dinâmica, ou seja, a potência durante a operação normal do circuito. Além disso, existe a potência estática, que reflete a corrente consumida pelo circuito, devido, entre outras coisas, a correntes parasitas dos transistores. Desta forma, quanto maior o número de transistores, provavelmente maior será a potência estática.

A baixa complexidade requerida por um sistema de baixo consumo para aplicações portáteis limita a escolha do compressor de áudio àqueles que trabalham exclusivamente no domínio do tempo, mencionados na seção 3.1, pois não necessitam da concepção de transformadas e/ou filtros digitais, que requerem uma quantidade bem maior de recurso computacional . Uma outra característica importante do projeto é que deve prever o acionamento aleatório do receptor, ou seja, quando o transmissor já está funcionando e o receptor é acionado.

A técnica de compressão de áudio escolhida para implementação foi o compressor ADPCM, pois apresenta melhor desempenho dentre os algoritmos de compressão no domínio do tempo estudados. Implementou-se o algoritmo proposto pela associação IMA, por apresentar desempenho similar e requerer menos recursos computacionais que outros algoritmos citados em literatura[2].

Como se trata de um algoritmo de compressão com perdas, o valor quantizado para a amostra A_n, que será chamado de Aq_n, pode diferir ligeiramente do valor da amostra A_n propriamente dita.

O algoritmo de compressão de áudio implementado pode ser dividido em duas partes principais: o cálculo da divisão da diferença entre a amostra atual e o último valor enviado pelo passo de quantização atual; e o ajuste do passo de quantização para a próxima amostra, A_{n+1}, com base no resultado da quantização da amostra atual, A_n.

Palavra codificada (Δ, delta)

O compressor implementado trabalha com taxa de compressão de 3:1, de forma que a cada amostra de 12 bits é associada uma palavra código de 4 bits, que será chamada de Δ(delta), como mostra a figura 6.1 :

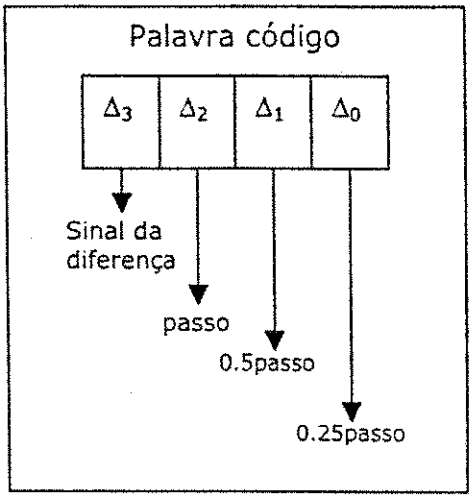


Figura 6.1 Palavra código do compressor ADPCM

O bit mais significativo da palavra código contém a informação de relação da amostra atual com a última amostra quantizada, de forma que e $\Delta_3=0$ a amostra atual é maior ou igual à última amostra quantizada; se $\Delta_3=1$, a amostra atual é menor que a última amostra quantizada.

Os outros três bits da palavra código correspondem ao módulo da diferença entre a amostra atual e a última amostra quantizada, em relação ao passo de quantização atual, $passo_n$. Δ_2 equivale a $passo_n$, Δ_1 equivale a ½ de $passo_n$, enquanto Δ_0 equivale a ¼ de $passo_n$, de modo que $\Delta_{[2:0]}= 111$ equivale a 1,75 do valor do passo atual, $\Delta_{[2:0]}= 101$ equivale a 1,25 do valor do passo atual, e assim por diante. Além disso, uma parcela com valor de $^1/_8$ de $passo_n$ é sempre adicionada ao módulo da diferença computada, para que o erro médio decorrente da compressão seja reduzido. A esta parcela associaremos o acrônimo PREM (Parcela de Redução de Erro Médio).

Desta forma, o algoritmo de quantização consiste no cálculo da diferença entre A_n e Aq_{n-1}, seguido da divisão desta diferença pelo passo de quantização atualizado, p_n e pode ser descrito em 5 passos a seguir:

Etapa 1 : calcular a diferença entre A_n e Aq_{n-1}. O sinal desta diferença será o bit mais significativo da palavra código (Δ_3), conforme explicado anteriormente.

Etapa 2: início da divisão binária da diferença calculada na etapa 1, que chamaremos de $diff_n$, pelo passo atual de quantização. Nesta etapa calcula-se Δ_2, representando este a situação em que $diff_n$ é maior ou igual a $passo_n$. Assim $\Delta_2=1$ se $diff_n \geq passo_n$, e $\Delta_2=0$ caso contrário. Como este é um processo de divisão binária, sendo $diff_n \geq passo_n$, $diff_n$ será atualizado com o valor $diff_n$-$passo_n$.

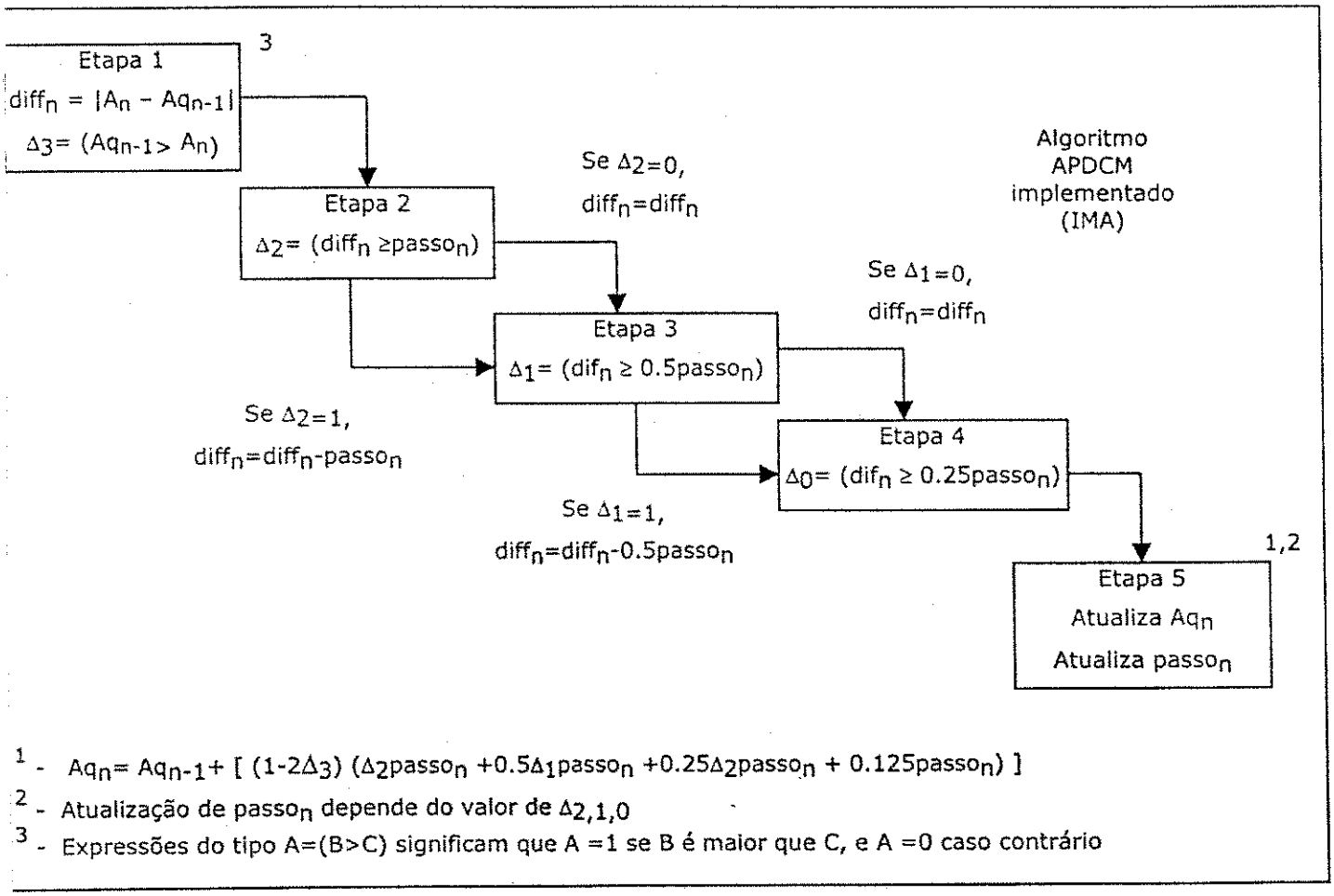
Etapa 3: cálculo de Δ_1. Aqui se repete o processo da etapa 2 para definir Δ_1. Se $diff_n$, atualizado na etapa 2, for maior ou igual a $0.5passo_n$, $\Delta_1=1$, e $\Delta_1=0$ caso contrário. Sendo $\Delta_1=1$, $diff_n$ será atualizado com o valor $diff_n - 0.5passo_n$.

Etapa 4: cálculo de Δ_0. Se $diff_n$, atualizado na etapa 3, for maior ou igual a $0.25passo_n$, $\Delta_0=1$, e $\Delta_0=0$ caso contrário. Ao final desta etapa, a palavra a ser enviada ao receptor já está definida.

Etapa 5: atualização do passo de quantização ($passo_{n+1}$) a ser utilizado na amostra futura (A_{n+1}), e do valor da amostra quantizada enviada Aq_n, verificando possíveis estouros. A amostra atual quantizada calculada, Aq_n é dada pela equação 6.1:

$$Aq_n = Aq_{n-1} + \{ (1-2\Delta_3) [\Delta_2 passo_n + 0.5\Delta_1 passo_n + 0.25\Delta_0 passo_n + 0.125 passo_n] \} \quad (6.1)$$

Estes passos podem ser descritos a partir do gráfico de estados na figura 6.2 :



gura 6.2 Descrição das etapas envolvidas no algoritmo de compressão de áudio implementado. Há várias formas de implementação de tal algoritmo em hardware, inclusive com a execução de duas ou mais etapas em paralelo.

Passo de quantização

A cada amostra, o passo de quantização, $passo_n$, é atualizado de forma a adaptar-se às características do sinal a ser quantizado. O valor de $passo_n$ é ajustado a partir da utilização de duas tabelas de alocação de valores. O passo de quantização $passo_{n+1}$ a ser utilizado na quantização da amostra A_{n+1} depende do valor de $passo_n$ e do módulo de $\Delta_{[2:0]}$ referente à amostra A_n.

Um registro chamado índice de passo armazena um valor entre 0 e 59, que será usado para endereçar uma tabela com os possíveis valores do passo de quantização. A modificação deste registro se dá através do uso da tabela 03, que contém os possíveis desvios a serem somados ao valor do índice de passo referente à amostra A_n para cálculo do índice para a amostra A_{n+1}. O valor deste desvio depende do módulo de $\Delta_{[2:0]}$ de acordo com a tabela 03 [18]:

Tabela 03

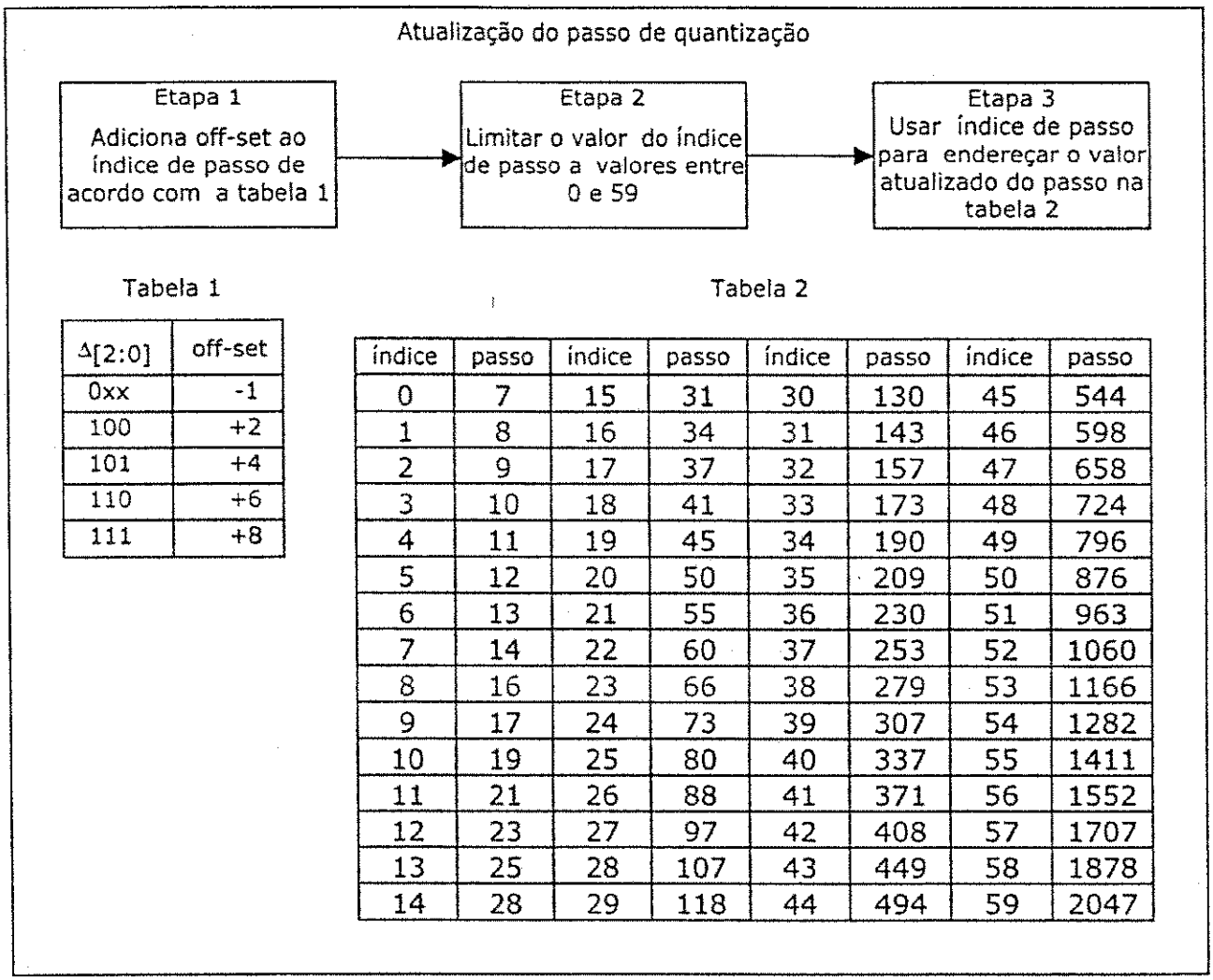
$\Delta[2:0]$	desvio
000	-1
001	-1
010	-1
011	-1
100	+2
101	+4
110	+6
111	+8

Tabela 03 :tabela com o valor dos possíveis desvios para o índice de passo

Assim, se o índice de passo atual contiver o valor 30 e, ao final da codificação da amostra atual, $\Delta_{[2:0]} = 110$, então o valor do desvio a ser somado ao índice de passo atual é +6. Após esta soma, o valor do índice é sempre limitado entre 0 e 59. No exemplo não haverá estouro e o índice de passo para a próxima amostra terá o valor de 36.

O índice assim atualizado representa o endereço a ser usado na tabela 02 da figura 6.3, que contém todos os possíveis valores do passo de quantização. Assim sendo, a partir da atualização do índice de passo, tem-se a atualização do passo de quantização através de valor armazenado em ROM. Este novo valor do passo de quantização é então utilizado no processo de quantização da próxima amostra, A_{n+1}. No exemplo, o índice de passo, que contém o valor 36, endereça o valor 230 na tabela 01 da figura 6.3. Este valor (230) será o valor do passo de quantização para a amostra A_{n+1}.

A figura 6.3 mostra as etapas envolvidas no ajuste do passo de quantização:



Δ[2:0]	off-set
0xx	-1
100	+2
101	+4
110	+6
111	+8

índice	passo	índice	passo	índice	passo	índice	passo
0	7	15	31	30	130	45	544
1	8	16	34	31	143	46	598
2	9	17	37	32	157	47	658
3	10	18	41	33	173	48	724
4	11	19	45	34	190	49	796
5	12	20	50	35	209	50	876
6	13	21	55	36	230	51	963
7	14	22	60	37	253	52	1060
8	16	23	66	38	279	53	1166
9	17	24	73	39	307	54	1282
10	19	25	80	40	337	55	1411
11	21	26	88	41	371	56	1552
12	23	27	97	42	408	57	1707
13	25	28	107	43	449	58	1878
14	28	29	118	44	494	59	2047

Figura 6.3 Algoritmo de atualização (adaptação) do passo de quantização, a partir da utilização de tabelas contendo valores de ajuste

Percebe-se assim que a palavra código, Δ_n, referente à amostra A_n depende do valor de A_{n-1} e de $passo_n$, que por sua vez depende do valor do Δ_{n-1} e de $passo_{n-1}$. Seguindo este raciocínio, pode-se mostrar que são necessárias apenas as palavras código calculadas desde o início da quantização do sinal para a recuperação da amostra quantizada atual. Desta forma, apenas a palavra codificada, Δ, será transmitida, sem haver necessidade de transmissão do passo de quantização.

Byte de dados

O byte de dados é assim formado por Δ^R referente ao canal direito em conjunto com Δ^L referente ao canal esquerdo, como mostra a figura 6.4:

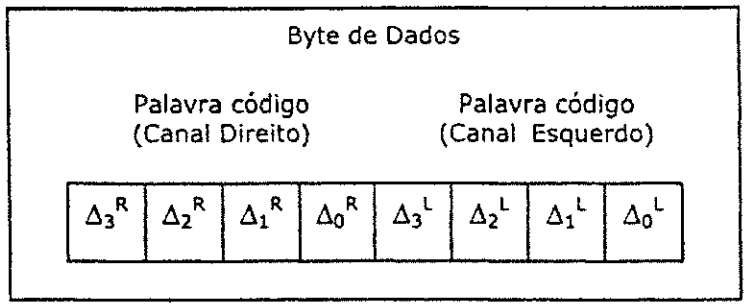


Figura 6.4 Byte de dados formado pela palavra código dos canais direito e esquerdo.

O algoritmo de recuperação de amostras quantizadas utiliza alguns blocos a serem sintetizados no compressor. A amostra atual recuperada, Ar_n, é calculada a partir da soma (ou subtração, dependendo do valor de Δ_3 recebido) do valor da amostra anterior recuperada, Ar_{n-1}, com o resultado da multiplicação do passo atual de quantização, $passo_n$, pelo módulo da palavra recebida, $\Delta_{[2:0]}$. Assim como o processo de compressão supracitado, este processo pode ser dividido em 5 etapas:

Etapa 1: início do cálculo do produto do passo atual pelo $\Delta_{[2:0]}$ recebido. Se $\Delta_2=1$, $Ar_n= Ar_{n-1}+\{[1- (2\Delta_3)]passo_n\}$ caso contrário $Ar_n= Ar_{n-1}$.

Etapa 2: continua cálculo. Se $\Delta_2=1$, $Ar_n= Ar_n+\{[1-(2\Delta_3)]0.5passo_n\}$, caso contrário $Ar_n= Ar_n$;

Etapa 3: continua cálculo.Se $\Delta_0=1$, $Ar_n= Ar_n+\{[1-(2\Delta_3)]0.25passo_n\}$, caso contrário $Ar_n= Ar_n$;

Etapa 4:adiciona ajuste de $passo_n/8$ (PREM). $Ar_n= Ar_n+\{[1-(2\Delta_3)]0.125passo_n\}$.

Etapa 5: atualiza valor do passo de quantização da mesma forma que é feito no codificador, com verificação de estouro do índice de passo.

Obs: em todos os passos de 1 a 4, é verificado se houve estouro na atualização da amostra. As 4 primeiras etapas estão contidas na equação 6.2 :

$$Ar_n=Ar_{n-1}+\{ [1-(2\Delta_3)] (\Delta_2 passo_n + 0.5\Delta_1 passo_n + 0.25\Delta_0 passo_n + 0.125 passo_n)\} \quad (6.2)$$

A figura 6.5 ilustra as 5 etapas envolvidas na recuperação das amostras, descritas anteriormente :

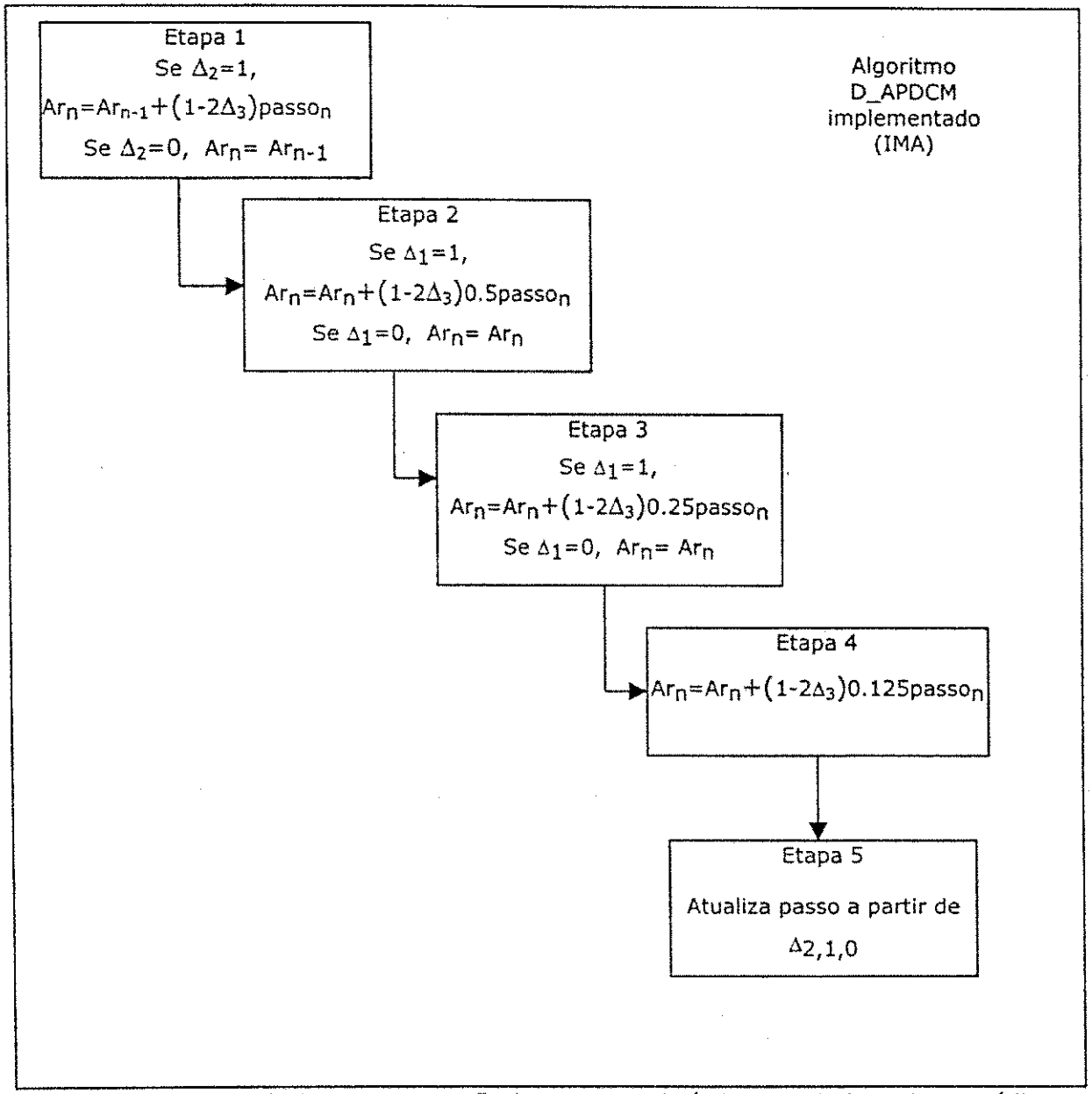


Figura 6.5 Etapas envolvidas na recuperação das amostras de áudio a partir das palavras código recebidas

Sincronismo

Quando do acionamento inicial do receptor, o índice de passo estará com o valor 0(zero), proveniente do reset do circuito, de forma que o valor do passo de quantização será igual a 7, e possivelmente diferirá do valor do passo no transmissor. Haverá portanto uma falha de sincronismo transmissor/receptor, já definida anteriormente como a situação em que o valor, no receptor, do passo de quantização, da amostra quantizada, ou de ambos, diferem do respectivo valor no transmissor.

Esta diferença no valor do passo de quantização e amostra quantizada entre transmissor e receptor poderá ocasionar dois problemas: nível DC e amplitude pico a pico errôneos no sinal recuperado.

A alteração na amplitude pico a pico do sinal recuperado em relação ao sinal quantizado no transmissor acontece na ocasião em que o passo de quantização no receptor for diferente do valor no transmissor. Desta forma, a multiplicação da palavra código detectada pelo passo de quantização, no receptor, formará uma diferença entre amostras maior ou menor de acordo com o valor do passo no receptor, se maior ou menor que o valor do passo no transmissor. Portanto, esta alteração na diferença entre amostras resultará na alteração de amplitude pico a pico do sinal detectado.

Já o nível DC surge quando, apesar de ter o mesmo valor do passo de quantização do transmissor, o valor da amostra recuperada é diferente do valor correspondente no transmissor. Nesta ocasião, as diferenças entre amostras do sinal recuperado terão a mesma amplitude pico a pico do sinal quantizado no transmissor, mas terão como referência um valor diferente, surgindo assim o nível DC.

Uma característica do algoritmo implementado que colabora para o estabelecimento do sincronismo transmissor/receptor é o fato dos valores do passo de quantização e do sinal detectado serem limitados tanto inferiormente quanto superiormente.

Estas limitações fazem com que o passo de quantização e o sinal detectado no receptor sejam gradativamente corrigidos, na medida em que se detecte o estouro de um e/ou outro. Havendo estouro, o valor detectado é limitado aos possíveis extremos, tornando-se mais próximo do valor correto.

Além disso, a limitação inferior faz com que qualquer intervalo de 88 amostras, ou seja, em um intervalo de tempo de 88 x (período de amostragem) = 88 x ($^1/_{\text{frequência de amostragem}}$) = (88 x $^1/_{20.000}$ = 4.4ms) com pequena amplitude pico a pico do sinal de áudio, ou mesmo um intervalo de silêncio, leva necessariamente o passo de quantização a seu valor mínimo. Isto significa que, na pior das hipóteses, o receptor sincronizará no primeiro intervalo de 4.4ms de silêncio após ser ligado.

Erros

Similar ao caso do acionamento do receptor, erros introduzidos ao longo do canal de comunicação podem provocar a perda de sincronismo transmissor/receptor. O sincronismo será gradativamente recuperado, como já explicado.

Simulação de erros/start

A introdução de erros aleatórios em bits ao longo do canal de comunicação foi simulada utilizando linguagem C. Um arquivo de som foi

comprimido e os bytes a serem enviados ao decodificador (4 bits por cada canal) foram aleatoriamente corrompidos a fim de verificarmos as consequências do ruído, quando na utilização do circuito real.

Como já foi explicado para o caso do acionamento do receptor, o sincronismo será gradativamente recuperado. Simulações de perda de sincronismo transmissor/receptor a partir de erros nas palavras código foram feitas e o sinal de áudio detectado foi analisado subjetivamente. A rotina para simulação inverte, aleatoriamente, todos os bits de um byte de dados, a uma taxa de 1 byte a cada 100,1000,10.000,..., simulando uma taxa de erro de byte de 10^{-2}, 10^{-3}, 10^{-4},...., respectivamente. Nos testes executados, não foi houve influência audível no sinal de áudio corrompido por uma taxa de erro de byte menor que 10^{-3}.

6.2 Implementação do compressor de áudio

A partir da análise do algoritmo de ajuste do passo de quantização, foi proposta uma nova forma de implementação que não utiliza memória ROM para armazenagem de valores, sendo sintetizada apenas com somadores, um subtrator e alguma lógica de controle. O esquema elétrico do circuito proposto pode ser encontrado no seguinte endereço: http://www.dee.ufpb.br/~asouza. Não foi encontrada forma similar de implementação na literatura estudada.

Após a compilação deste circuito utilizando-se a ferramenta MaxplusII, verificou-se que tal circuito exige para a síntese mais macro-células que o circuito que utiliza tabelas de armazenamento de valores. Desta forma, esta implementação foi descartada.

Ao analisar o algoritmo de compressão de áudio, percebe-se que alguns blocos utilizados no codificador são requeridos no decodificador, como exemplo, somadores, subtratores, tabelas de armazenagem de valores e lógica de ajuste de passo. Isto faz com que sub-módulos desenvolvidos sejam compartilhados entre transmissor e receptor.

O processo de descrição do hardware do compressor ADPCM foi bastante exaustivo, pois existem diversas formas de se descrever sua funcionalidade. Assim, buscou-se ao longo de várias implementações e simulações conseguir o circuito que exigisse o menor número de macro-células. A seção 6.1, *algoritmo escolhido*, sugere a implementação do compressor de áudio a partir de uma máquina de 5 estados, necessitando, pois, de 5 ciclos de clock para sua execução (por canal).

Célula ADPCM

O compressor foi concebido a partir de uma máquina de 5 estados executando as 5 etapas descritas na seção 6.1.

Compressor estéreo

Duas configurações de circuitos foram comparadas :

Circuito 1: utilizou duas células de compressor ADPCM, uma para executar a compressão do sinal do canal R e outra para a compressão do canal L. Estas duas células, completamente independentes, funcionam concomitantemente, ou seja, realizam as mesmas operações ao mesmo tempo. O circuito necessita de 5 ciclos de clock para operação completa, ou seja, um ciclo de clock para executar cada uma das 5 etapas descritas na seção 6.1. O código de descrição do circuito 1 pode ser encontrado no seguinte endereço: http://www.dee.ufpb.br/~asouza. A figura 6.6 ilustra a arquitetura do circuito 1 :

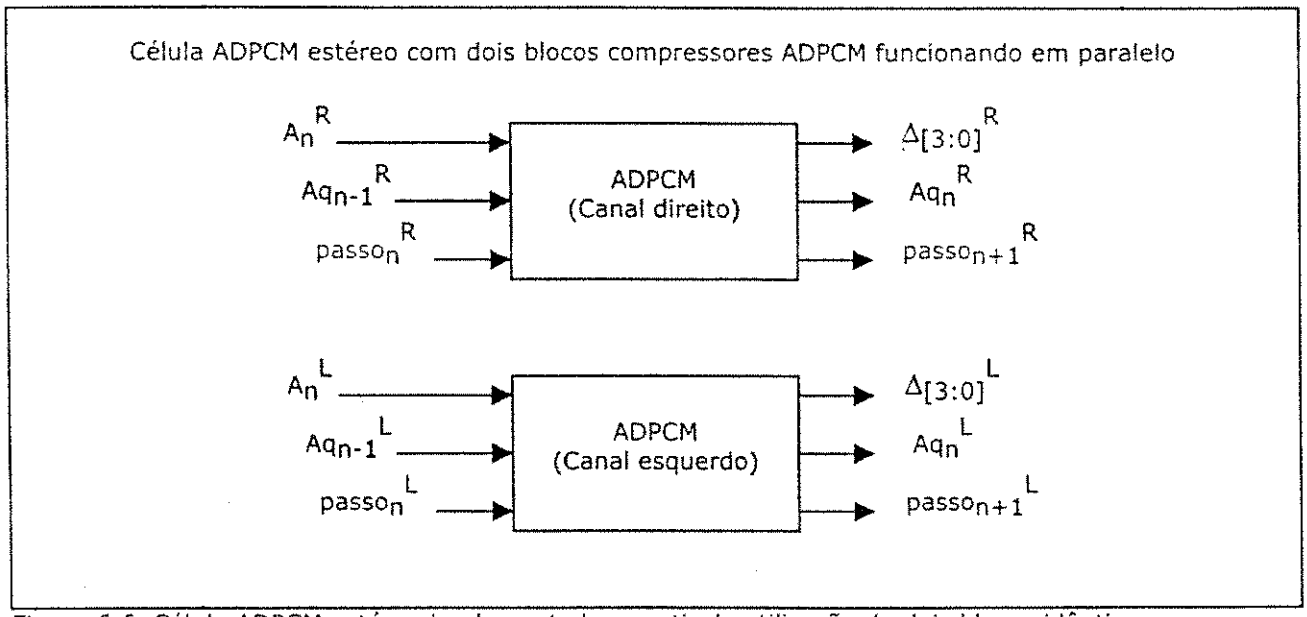


Figura 6.6 Célula ADPCM estéreo implementada a partir da utilização de dois blocos idênticos e independentes, cada qual executando a compressão de um canal de áudio.

Circuito 2: utilizou apenas uma célula de compressor ADPCM, realizando a compressão dos canais R e L em tempos diferentes. Um circuito especial foi sintetizado de modo a carregar, com os valores corretos ,os registros de cada canal no tempo adequado. A operação completa é executada em 9 ciclos de clock, pois um ciclo de clock pode ser aproveitado para execução de duas etapas :a etapa 5 do canal direito é executada no mesmo ciclo de clock da etapa 1 do canal esquerdo. O código de descrição do circuito 2 pode ser encontrado no seguinte endereço: http://www.dee.ufpb.br/~asouza. A figura 6.7 ilustra a arquitetura do circuito 2 :

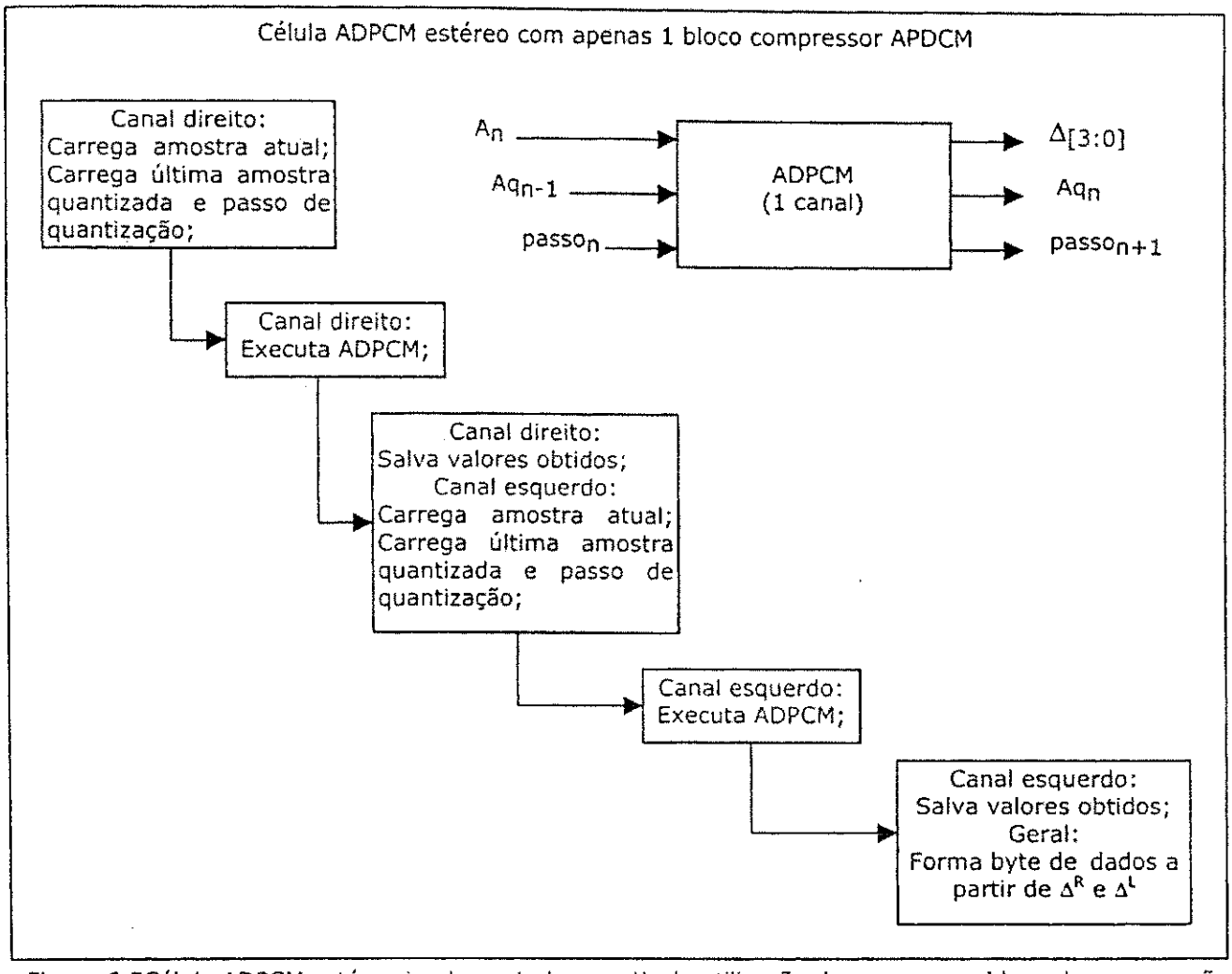


Figura 6.7Célula ADPCM estéreo implementada a partir da utilização de apenas um bloco de compressão ADPCM, executando a compressão dos canais direito e esquerdo em tempos diferentes.

Após a confirmação do correto funcionamento dos dois circuitos, examinou-se a quantidade de macro-células requeridas na síntese de cada um deles. O circuito da figura 6.6 requereu 25% a mais de macro-células que o circuito da figura 6.7. Assim sendo, o circuito final escolhido contém apenas uma célula de compressor ADPCM executando a compressão dos canais R e L em tempos diferentes.

6.3 Simulações do compressor

A verificação funcional dos circuitos propostos para o compressor de áudio foi feita através da comparação de sua funcionalidade com a funcionalidade do algoritmo descrito em linguagem de alto nível, neste caso a linguagem C.

Inicialmente, um rotina escrita em linguagem C gera um arquivo texto com várias palavras aleatórias de 4 dígitos hexadecimais, como visto na figura 6.8:

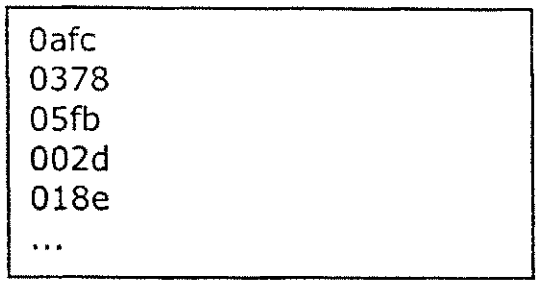


Figura 6.8 arquivo de amostras aleatórias de áudio

Este arquivo serve de fonte de amostras de áudio de 12 bits (os 4 bits mais significativos da palavra serão descartados, de forma que a primeira palavra do arquivo, no exemplo 0afc, fornece a amostra afc, com 12 bits de precisão. Este arquivo serve de fonte de amostras de áudio tanto para o hardware descrito em Verilog quanto para a rotina em linguagem C que implementa o algoritmo ADPCM.

A rotina ADPCM em linguagem C então abre este arquivo de amostras e realiza a compressão de duas em duas amostras (uma associada ao canal L e outra ao canal R), formando assim bytes de dados resultantes da compressão estéreo. Os bytes de dados formado são agrupados e escritos em um arquivo que servirá de referência de funcionalidade para o hardware descrito em verilog. A figura 6.9 ilustra o arquivo de bytes de dados gerado:

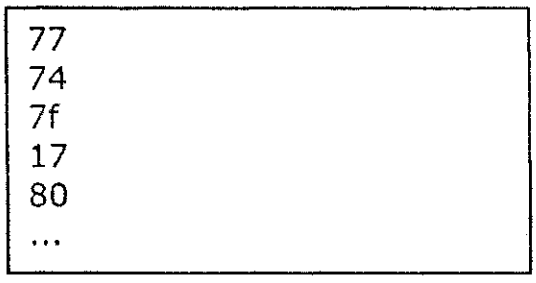


Figura 6.9 arquivo com dados gerados pela rotina ADPCM implementada em linguagem C

Feito isto, uma rotina de teste escrita em linguagem Verilog abre o mesmo arquivo de amostras (figura 6.8), e fornece pares de amostras ao circuito ADPCM descrito em linguagem Verilog, além de fornecer outros sinais de estímulos necessários ao funcionamento do circuito, como sinal de clock e reset.

A simulação do circuito é feita através da compilação do hardware descrito em Verilog pelo software VLOGCMD[23]. O circuito simulado gera, pois, os bytes de dados referentes às amostras utilizadas na compressão. Estes bytes de dados são agrupados e a rotina de teste gera um arquivo

com estes bytes de dados, como ilustra a figura 6.10 :

```
77
74
7e  <- erro
17
80
...
```

Figura 6.10 arquivo com bytes de dados gerados pelo compressor ADPCM em linguagem Verilog. Note-se um erro no terceiro byte de dados do arquivo, pois o valor esperado no exemplo é 7f (vide figura 6.9)

O arquivo com bytes de dados proveniente do código verilog simulado é comparado, via comando "fc" do DOS que compara dois arquivos binários, com o proveniente da rotina em linguagem C.

Nas simulações iniciais, que levaram apenas alguns segundos para sua realização, o arquivo de amostras utilizado continha 100 pontos por canal. Ao comparar os dois arquivos de bytes de dados gerados, os primeiros erros de concepção do circuito digital foram descobertos (figura 6.10). A partir da análise da discrepância entre os resultados, foram identificados os erros a serem corrigidos.

Após a correção dos erros detectados, um arquivo com 1000 amostras foi gerado, sendo utilizado o mesmo processo descrito acima. A simulação durou cerca de 3 minutos e resultou na ocorrência de erros mais sutis, em casos não cobertos durante a primeira fase da simulação (arquivo com 100 amostras).

Corrigidos mais uma vez os erros, vários arquivos com 30.000 amostras foram gerados, para utilização em diversas simulações, que duraram cerca de 30 minutos. Obtiveram-se as mesmas saídas que as resultantes do código em linguagem C.

Uma vez feito isto, garantiu-se que os circuitos descritos em linguagem Verilog executavam a mesma função do algoritmo ADPCM proposto. O arquivo de teste em Verilog, assim como o algoritmo em linguagem C, podem ser encontrados no seguinte endereço: http://www.dee.ufpb.br/~asouza.

6.4 Implementação do Descompressor de áudio

A célula D_ADPCM (decodificador ADPCM) foi implementada a partir de uma máquina de 5 estados, executando os 5 passos descritos na seção 6.1. Circuitos combinacionais foram utilizados para executar funções de soma, subtração e comparação, em conjunto com flip-flops para armazenamento dos resultados de cada passo.

Foram sintetizados dois circuitos decodificadores, da mesma forma que no caso do compressor, contendo uma célula D_ADPCM executando a descompressão dos canais direito e esquerdo em intervalos diferentes, e um circuito com duas células D_ADPCM trabalhando em paralelo. A análise dos resultados de compilação dos dois circuitos mostra que o circuito com apenas uma célula D_ADPCM consome cerca de 26% a menos de macro-células. Desta forma, o circuito escolhido para implementação em hardware foi o descompressor estéreo que utiliza apenas uma célula D_ADPCM. Os arquivos de descrição da célula decodificador e dos circuitos para comparação, assim como os que contém os resultados de compilação (número de macro-células requeridas) podem ser encontradas no seguinte endereço : http://www.dee.ufpb.br/~asouza. A figura 6.11 ilustra o circuito descompressor ADPCM estéreo que utiliza apenas 1 célula D_ADPCM, enquanto a figura 6.12 ilustra a utilização de duas células D_ADPCM trabalhando em paralelo :

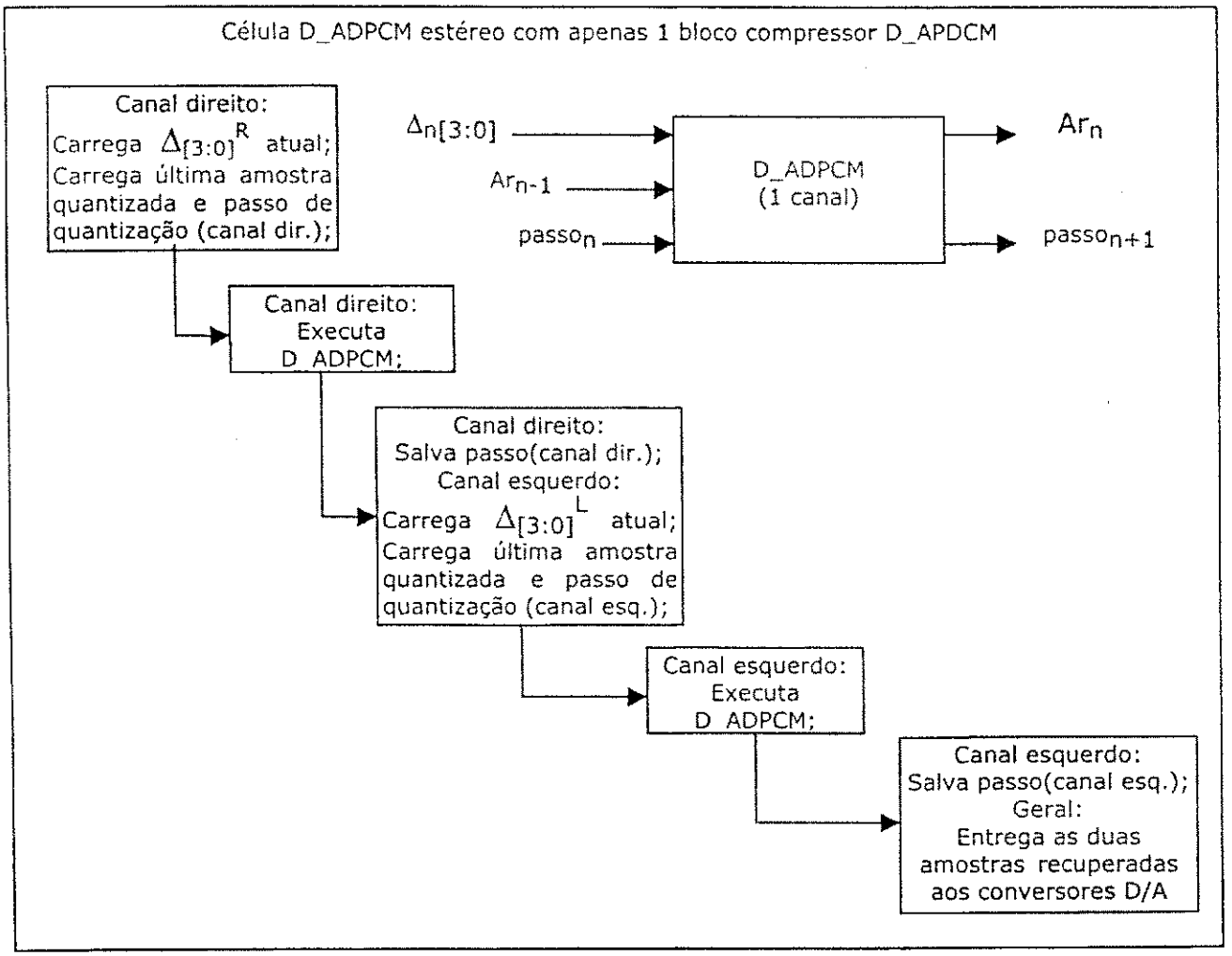


Figura 6.11 Célula D_ADPCM estéreo implementada a partir da utilização de apenas um bloco de-compressor D_ADPCM, executando a recuperação de amostras dos canais direito e esquerdo em tempos diferentes.

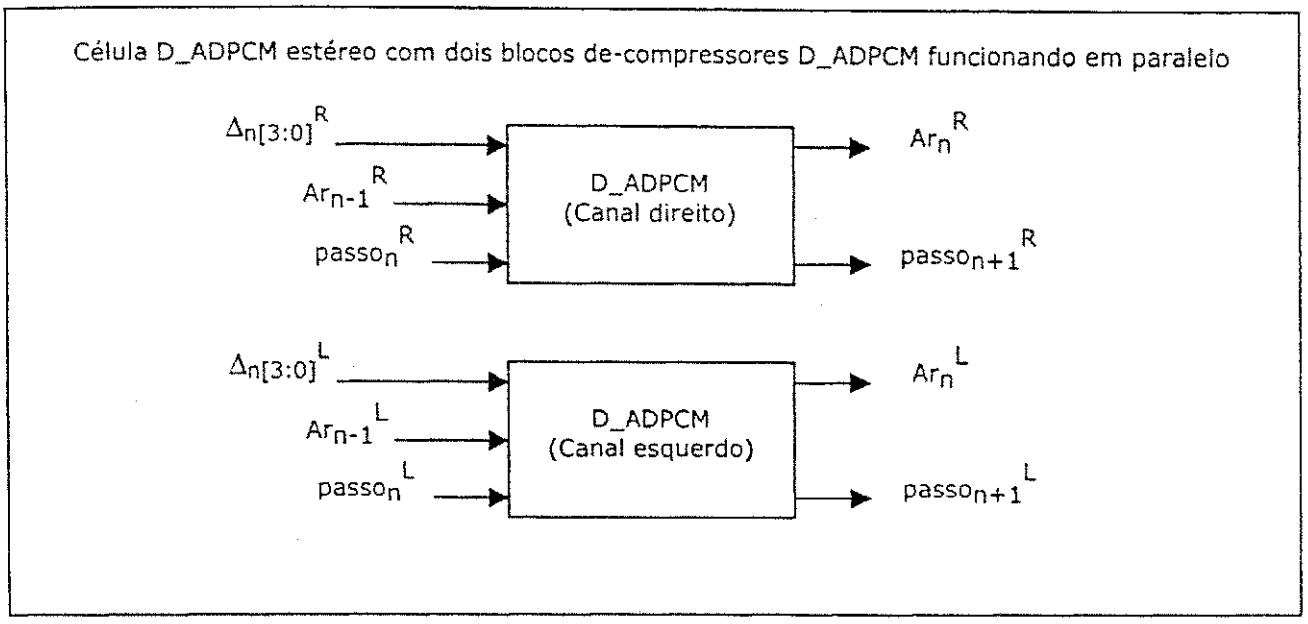


Figura 6.12 Célula D_ADPCM estéreo implementada a partir da utilização de dois blocos idênticos e independentes, cada qual executando a recuperação de amostras de um canal de áudio.

6.5 Simulações do Descompressor

Uma vez descrito em linguagem Verilog, o descompressor foi simulado utilizando o mesmo procedimento descrito na seção 6.3, *Simulações do compressor*. Mesmo antes da simulação deste bloco, alguns erros de concepção foram encontrados, decorrentes dos resultados da simulação do compressor de áudio.

Erros mais sutis foram encontrados a partir das simulações de 1000 amostras. Uma vez corrigidos, o código em Verilog foi submetido à extensiva verificação funcional, com vários arquivos de 30.000 amostras. Os resultados obtidos estão de acordo com os resultados gerados em linguagem de alto nível. Uma vez feito isto, garantiu-se que o circuito descompressor funciona de acordo com o algoritmo proposto.

6.6 Conclusões

O algoritmo ADPCM proposto é um algoritmo de baixa complexidade e desempenho satisfatório para sistemas secundários de entretenimento, onde não há necessidade de alta fidelidade. Os circuitos compressor e descompressor foram sintetizados de várias maneiras a fim de se comparar a quantidade de macro-células requeridas por cada implementação. Circuitos bastante compactos foram conseguidos, implementáveis na família MAX9400.

O algoritmo apresenta deficiência de robustez na presença de ruído. Foi observado ruído audível nas simulações feitas taxa de inversão de byte de 10^{-2}, comportamento insatisfatório ao usuário. Isto indica que o algoritmo requer uma modulação digital de boa eficiência na presença de ruído.

7. Modulador digital

7.1 Modulação digital escolhida

Um das preocupações deste trabalho foi a quantidade de componentes externos necessários ao processo de modulação/demodulação em banda-básica. Circuitos osciladores para geração de sub-portadoras no modulador e no demodulador, no caso de demodulação coerente exigida pela modulação PSK, exigem alguns requisitos de implementação. No caso de serem implementados com circuitos que utilizam capacitores e indutores, tais componentes variam de acordo com a temperatura. Uma das formas de contornar este problema seria a implementação de um circuito adicional que compense este efeito térmico.

Levando isto em consideração, alguns tipos de modulação digital foram estudadas, como visto no capítulo 4. Em seguida, foram simuladas duas formas de modulação em fase (a modulação em amplitude foi descartada por motivos já citados no capítulo 4): a modulação DPSK e a modulação MSK, em conjunto com uma análise das possíveis formas de implementação. O motivo da seleção da modulação DPSK para simulação foi a possível demodulação não coerente, já citada no capítulo 4.

Nas análises feitas, o modulador MSK proposto obteve um desempenho em ocupação de banda (vide gráfico 7.3) levemente superior ao modulador DPSK proposto. Além disso, no caso do demodulador MSK o hardware necessário à demodulação dos bits é de baixa complexidade, como será visto.

Inicialmente, foi feita uma rotina para análise do sinal MSK. Esta rotina, implementada em linguagem Matlab, mostra graficamente as diversas variantes do sinal MSK, onde o usuário pode ajustar tanto o número de ciclos associados a cada bit, quanto a amplitude de cada ciclo, a fim de simular o canal de comunicação. Além disso, a rotina ainda executa análise espectral do sinal MSK, mostrado a ocupação de banda do sinal em questão, em relação à frequência de 1Hz/bit/s.

O gráfico (a) da figura 7.1 mostra um sinal digital qualquer. No gráfico (b), aos bits 1 do gráfico (a) é associado meio ciclo de uma onda senoidal de frequência $^{tx}/_{2}$, onde tx é a taxa de bits; já aos bits 0 é associado um ciclo de onda senoidal de frequência tx; as amplitudes de picos são iguais para as duas formas de onda(1V). Estes dois gráficos são repetidos, em menor resolução, nos gráficos (c) e (d). O gráfico (e) da figura 7.1 mostra o espectro do sinal MSK dos gráficos (b) e (d), em unidades de $^{Hz}/_{bits/s}$, ou seja, para uma taxa de 100Kbits/s , o valor 1 do eixo horizontal do gráfico (e) representa o valor de 100KHz:

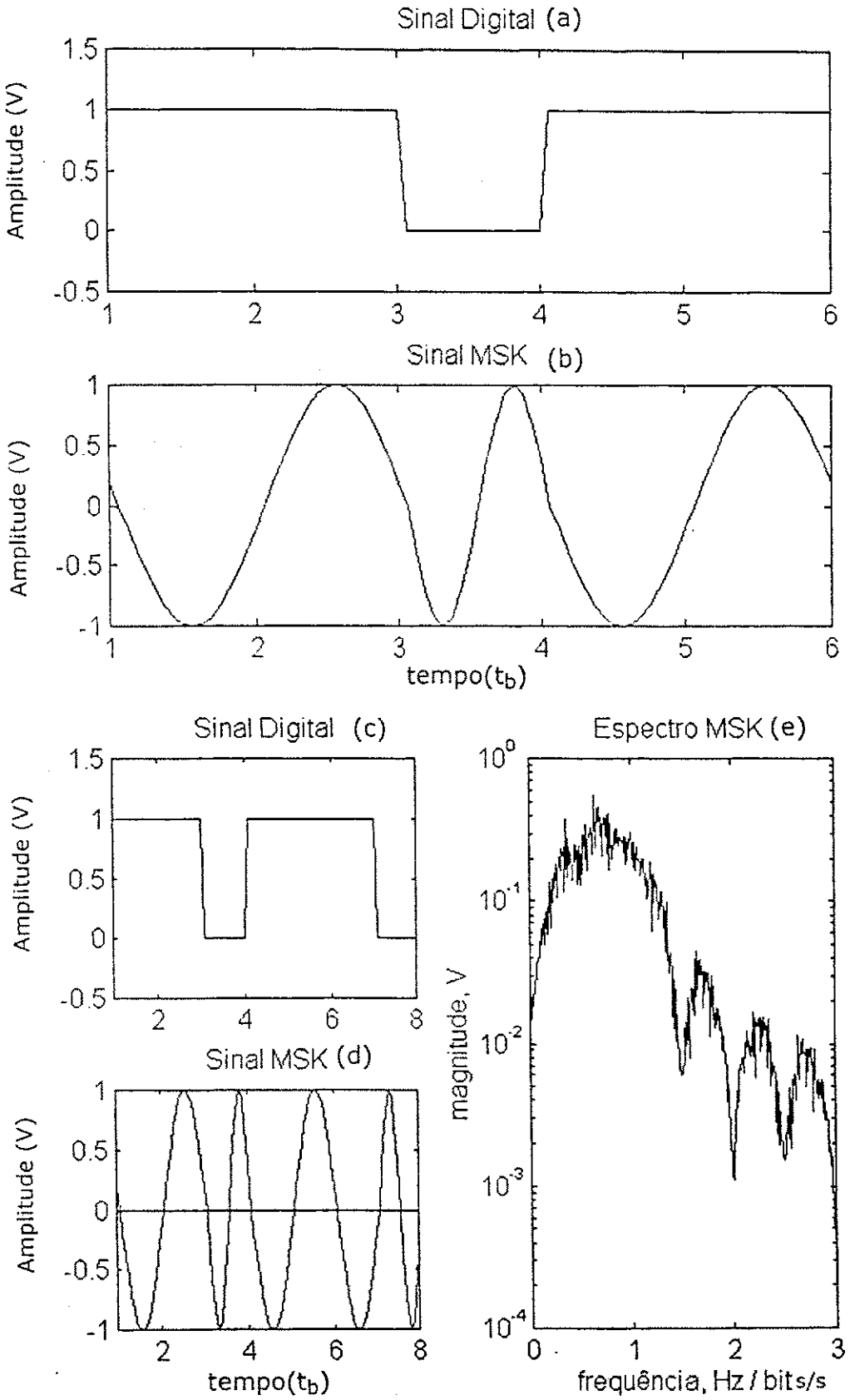


Figura 7.1 Geração do sinal MSK implementado com as seguintes especificações:
Amplitude de pico do sinal correspondente ao bit 1, AMP1 = 1v;
Amplitude de pico do sinal correspondente ao bit 0, AMP0 = 1v;
Número de ciclos do sinal correspondente ao bit 1, CICLO1 = 0.5;
Número de ciclos do sinal correspondente ao bit 0, CICLO0 = 1;
Nota-se ainda a análise espectral do sinal MSK puro. Escala horizontal

No gráfico (b) da figura 7.2, aos bits 1 do sinal digital do gráfico (a) é associado um ciclo completo de onda senoidal de frequência tx, enquanto aos bits 0 é associado um ciclo e meio de onda senoidal de frequência 1.5tx. O gráfico (c) mostra o espectro do sinal MSK da figura (b):

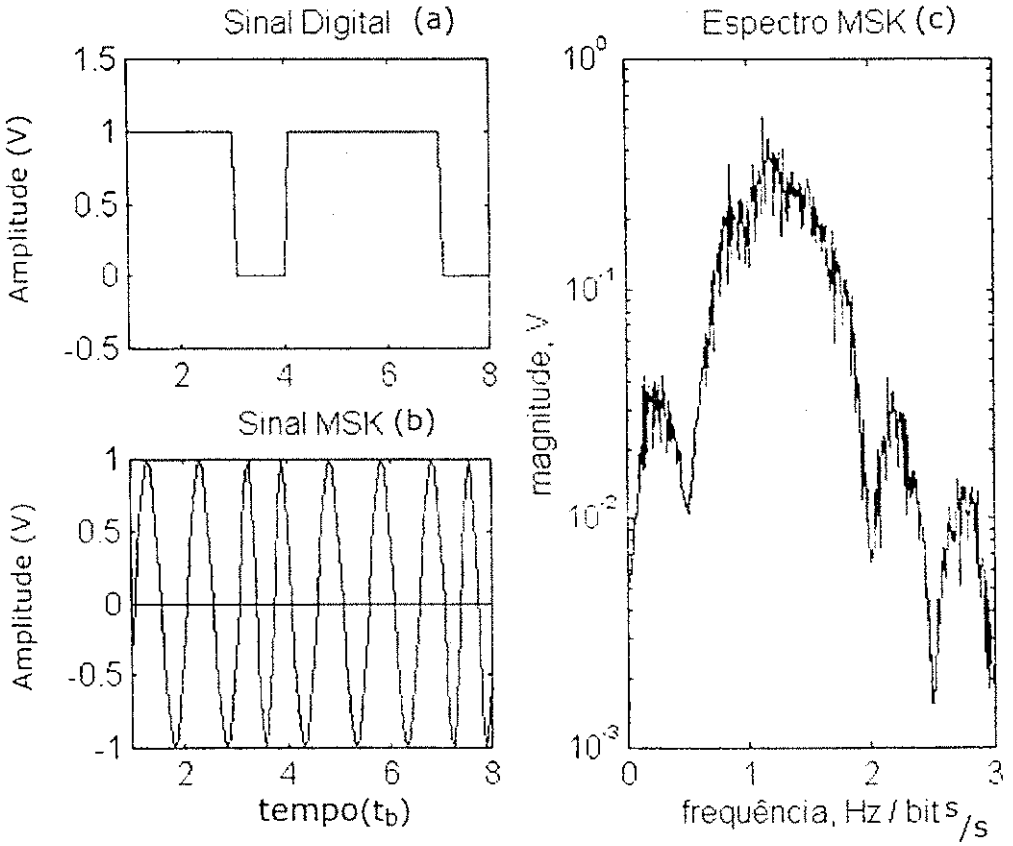


Figura 7.2 Geração do sinal MSK implementado com as seguintes especificações:
Amplitude de pico do sinal correspondente ao bit 1, AMP1 = 1v;
Amplitude de pico do sinal correspondente ao bit 0, AMP0 = 1v;
Número de ciclos do sinal correspondente ao bit 1, CICLO1 = 1;
Número de ciclos do sinal correspondente ao bit 0, CICLO0 = 1.5;
Nota-se ainda a análise espectral do sinal MSK . Escala horizontal em unidades de Hz/bits/s .

Uma outra rotina em linguagem MATLAB mostra o desempenho, em termos de espectro ocupado, do sinal DPSK, com variação de fase de 180° associada ao bit 1, enquanto não há variação de fase associada ao bit 0. Na figura 7.3, o gráfico (b) representa o sinal DPSK correspondente ao sinal digital do gráfico (a). Este sinal apresenta uma variação de fase de 180° a cada bit 1 do sinal digital. No gráfico (c), tem-se o espectro do sinal DPSK, com a escala horizontal em unidades de Hz/bits/s . Já no gráfico (d) estão expostos os espectros dos sinais MSK(oriundo da figura 7.1) e DPSK, para uma análise comparativa de ocupação de banda:

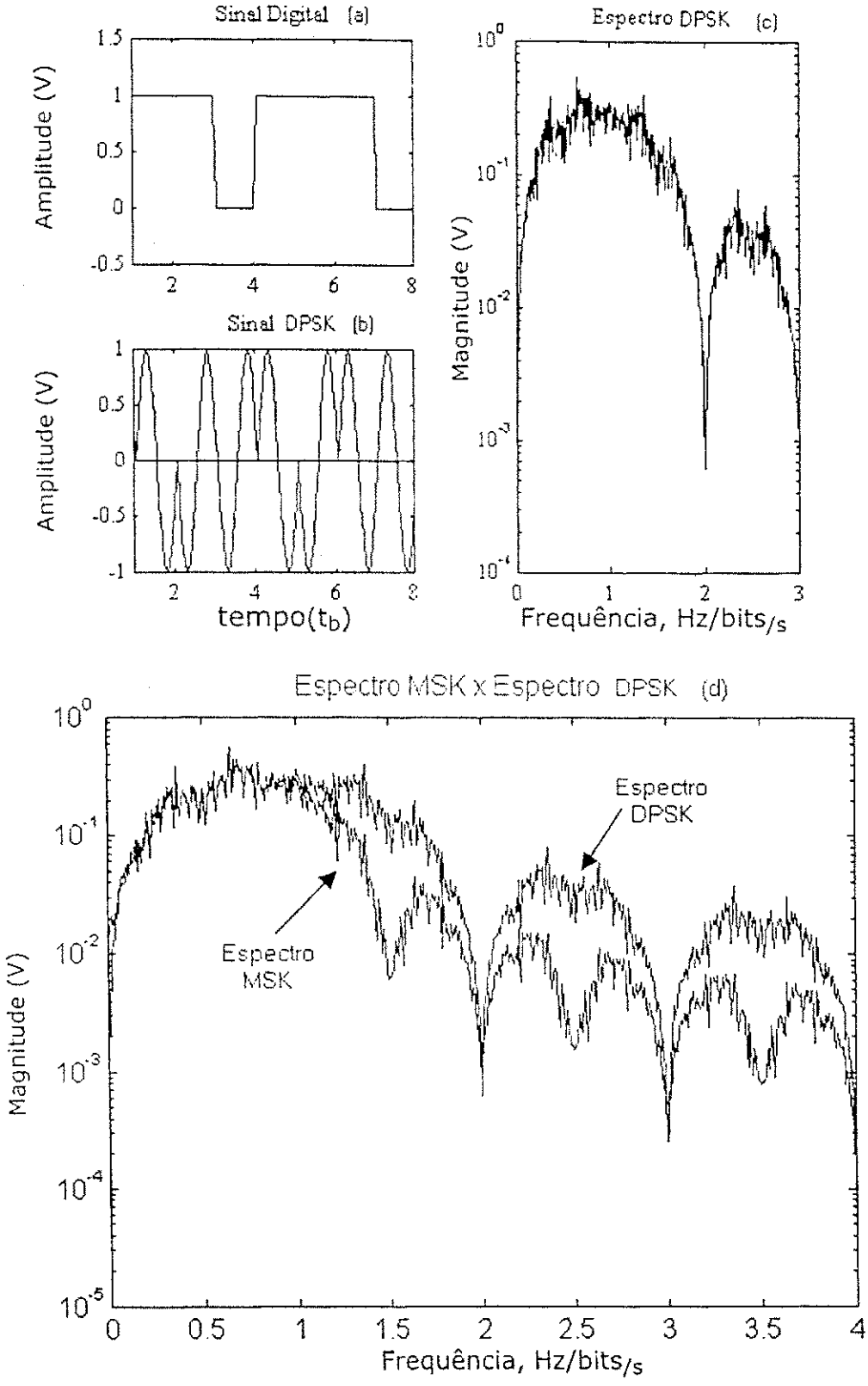


Figura 7.3 Gráfico superior com análise do sinal DPSK. Variação de fase de 180° associada ao bit 1 e variação nula de fase associada ao bit 0. Note-se o espectro resultante, em unidades de Hz/bit/s(eixo horizontal).Gráfico inferior: comparação de espectro de freqüências dos sinais MSK e DPSK. O sinal MSK proposto exibe maior atenuação que o sinal PSK para freqüências maiores que 1.25 Hz/bit/s.

A partir da análise destes resultados, mostrando uma ocupação menor de largura de banda por parte da modulação MSK, em conjunto com a complexidade de implementação do par modulador/demodulador, definiu-se para este trabalho a modulação MSK com a configuração da figura 7.1.

7.2 Implementação do Modulador

No transmissor, o sinal correspondente a cada bit é armazenado em memória ROM com precisão de 8 bits. A modulação é concebida a partir de um contador síncrono de 16 estados que endereça tais pontos em ROM e os entrega a um conversor D/A de 8 bits, construindo assim o sinal modulado. A figura 7.4 mostra os pontos armazenados em ROM que formam o sinal MSK com 16 pontos por bit :

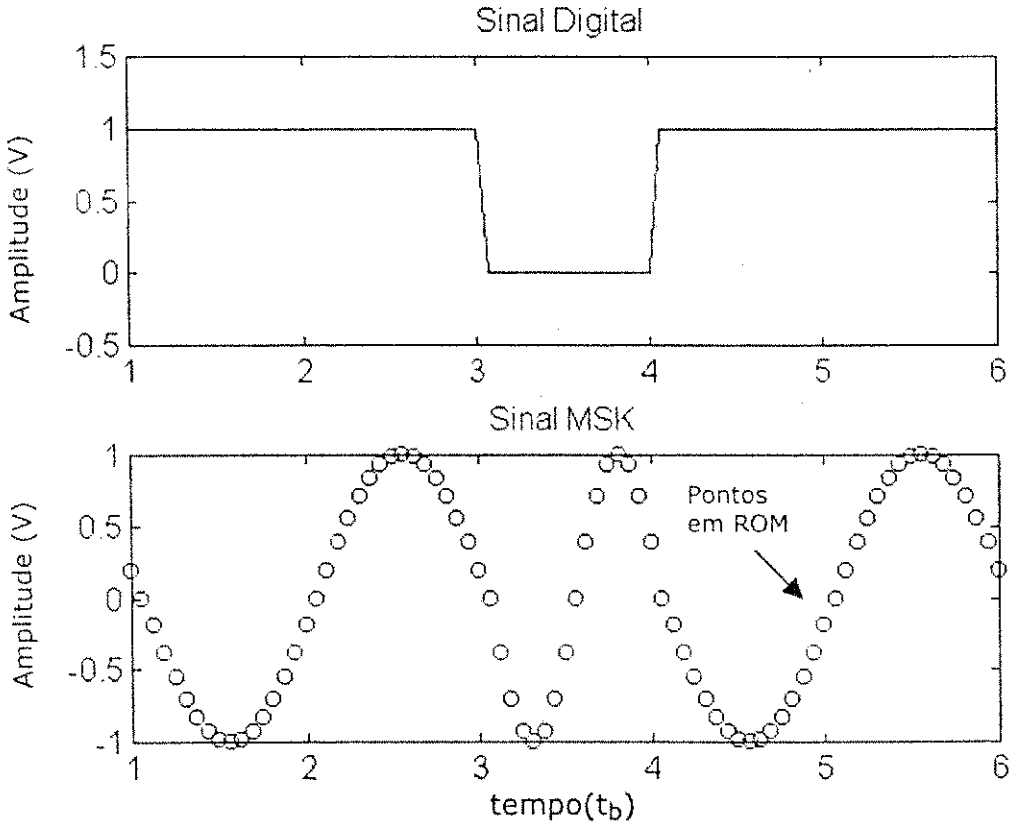


Figura 7.4 Sinal MSK construído a partir de pontos armazenados em memória ROM a serem entregues a um conversor $D/_A$ (16 pontos por bit).

A vantagem da utilização deste método de concepção do modulador consiste em sua simplicidade e eficácia: não são necessários osciladores na modulação, ao mesmo tempo em que o sinal gerado tem a precisão dos pontos armazenados em ROM, em conjunto com o conversor D/A necessário na saída do modulador.

Na representação do sinal MSK, o número de pontos por bit é o número de amostras do sinal MSK que serão entregues ao conversor D/A para a formação da forma de onda associada a cada bit. Inicialmente, a construção do sinal MSK no transmissor com apenas 8 pontos por bit é satisfatória, pois temos assim uma frequência de amostragem cerca de 5 vezes maior que o valor de 1.5 Hz por bit do sinal MSK (sendo portanto aproximadamente 2.5 vezes maior que a frequência de Nyquist para a banda em questão).

Mas buscando a utilização do mesmo cristal tanto pelo transmissor como pelo receptor, para aumentar o número de componentes comprados de mesmo valor de frequência (reduzindo assim o custo de cada peça), o circuito foi redesenhado de forma a compor o sinal MSK com 16 pontos por bit (este é o valor de clock requerido pelo demodulador, como será visto a seguir na seção 7.4), de modo a ter-se uma frequência de amostragem aproximadamente 10 vezes maior que o valor de 1.5 Hz por bit. Este aumento de pontos por bit acarretou aumento do tamanho da ROM necessária.

No caso da precisão do sinal MSK, o mesmo raciocínio foi seguido. O valor de 8 bits de precisão levou em consideração a facilidade de aquisição de células de conversores D/A de 8 bits, que geralmente são disponibilizadas pela tecnologia a serem futuramente implementados os circuitos aqui descritos.

De qualquer forma, a utilização de 6 bits de precisão forma um sinal MSK relação sinal-ruído, referente à quantização, de aproximadamente 30 dB, o que ainda resulta na margem de 22 dB até que se chegue ao valor de 8 dB de relação sinal-ruído necessários para que o demodulador recupere corretamente os bits (este valor é resultado das simulações feitas com o circuito demodulador proposto, e encontra-se descrito na seção 7.5). Por margem de 22 dB entenda-se o valor da redução da relação sinal-ruído do sinal MSK, pela inserção de ruído ao longo do canal, de forma que ainda se consiga recuperação correta dos bits. Um conversor comercial com precisão maior que 6 bits facilmente encontrado no mercado é o de 8 bits.

Para a concepção do contador de 16 estados, foram utilizados 4 flip-flops . A cada pulso de clock, este contador incrementa sua saída, até que esta atinja o valor 15. Quando isto acontecer, no próximo pulso de clock a saída do contador passará ao valor 0(zero), e repetirá o processo de contagem. Seus 4 bits de saída são utilizados para endereçar a memória ROM, para que se construa o sinal MSK, como mostra a figura 7.5:

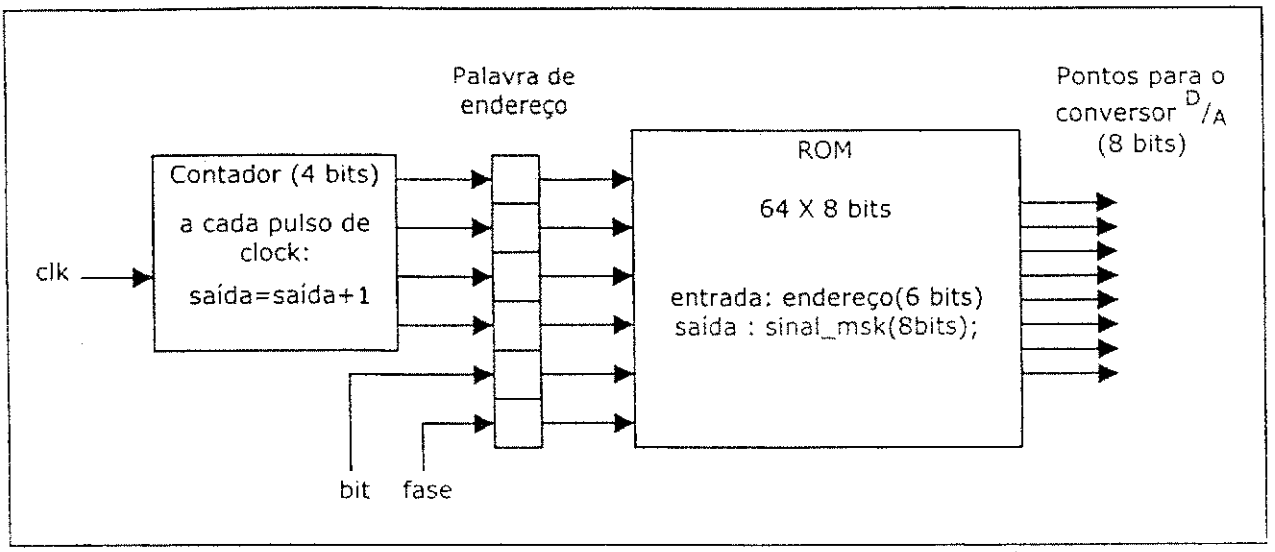


Figura 7.5 Endereçamento da ROM com pontos do sinal MSK. Os sinais *bit* e *fase* decidem uma entre 4 formas de onda a serem compostas, enquanto a saída do contador de 16 estados (4 bits) endereça os 16 pontos de cada uma delas

Além dos 4 bits do contador, mais 2 sinais são necessários para decidir uma entre 4 formas de onda. São eles o sinal *fase* e o próprio bit a ser transmitido. Desta forma, as 4 combinações possíveis entre estes dois bits irão formar as formas de ondas que constam na figura 7.6:

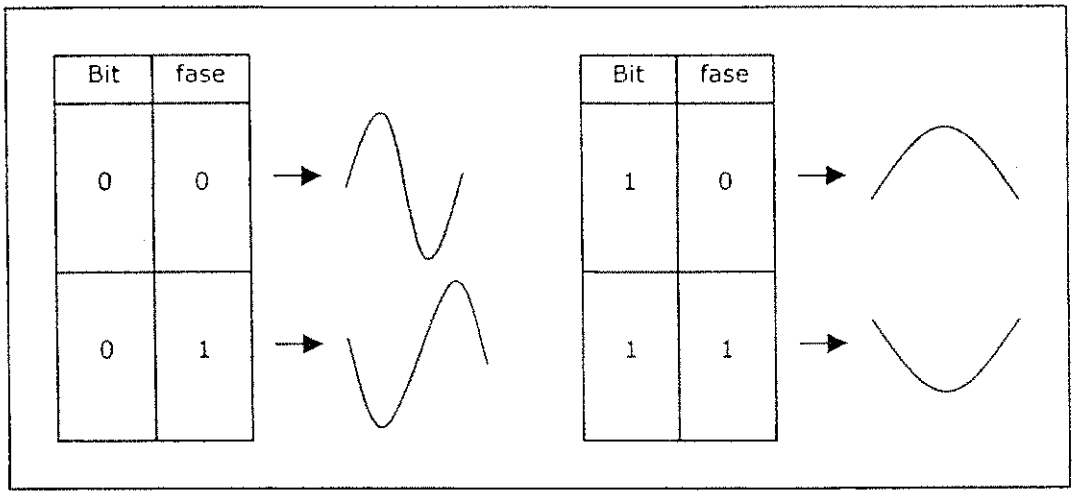


Figura 7.6 As 4 formas de onda determinadas pelos sinais *bit* e *fase*

A ROM foi implementada de forma que o primeiro ponto da forma de onda, quando a saída do contador tem valor nulo, é sempre 8'b10000000 (nível intermediário da excursão de saída da ROM, que vai de 8'b00000001 a 8'b11111111) e o último ponto pode ser maior ou menor que zero, dependendo do bit a ser transmitido e da fase atual.

A partir da observação do sinal MSK percebe-se que haverá uma inversão de fase da forma de onda a ser transmitida, sempre que o último bit transmitido foi 1, não havendo inversão de fase associada ao bit 0. Pode-se assim executar a escolha da fase para o próximo bit. Sempre que o bit atual for 1, o próximo bit será transmitido com uma forma de onda com fase inversa à última transmitida para tal bit. O circuito que executa a operação de decisão da fase do próximo bit faz uso de uma porta XOR cujas entradas são o valor da fase e bit atuais, e a saída é válida para a transmissão do próximo bit.

Sincronismo de Frame

Ao byte de dados é adicionado um bit de sincronismo de frame, para que haja sincronização do byte de dados entre modulador e demodulador MSK. Desta forma, tem-se um frame de 9 bits por amostra dos canais R e L. Este bit de sincronismo é um bit 1(nível lógico alto). Assim, pode-se esperar que a partir da aleatoriedade do byte de dados vindo de compressor de áudio, numa sequência de alguns frames o único bit que se mantém sempre em 1 seja o bit de sincronismo.

Resta então ao demodulador testar a permanência no estado 1(nível lógico alto) a cada 9 bits, que é o número de bits em um frame como mostra a figura 7.7. Na ausência de permanência, ou seja, quando o bit recebido, que espera-se que seja o sincronismo, for 0 (nível lógico baixo), deve-se então mudar a referência do bit de sincronismo no frame, testando assim o próximo bit.

A escolha do bit de sincronismo para o frame MSK como sendo o bit 1 levou em consideração o seguinte fato: para que o passo de quantização passe de seu menor valor(7) ao seu maior valor(2048), são necessárias 8 palavras consecutivas do tipo 4'bx111. Exemplificaremos com o caso 4'b1111. Isto significa que dificilmente um sinal de áudio resulte na transmissão de mais de 8 palavras consecutivas do tipo 4'b1111, já que o valor máximo de passo de quantização cobre aproximadamente 94% da faixa dinâmica do sinal de entrada, o que indicaria uma possível anormalidade do sinal de áudio de entrada.

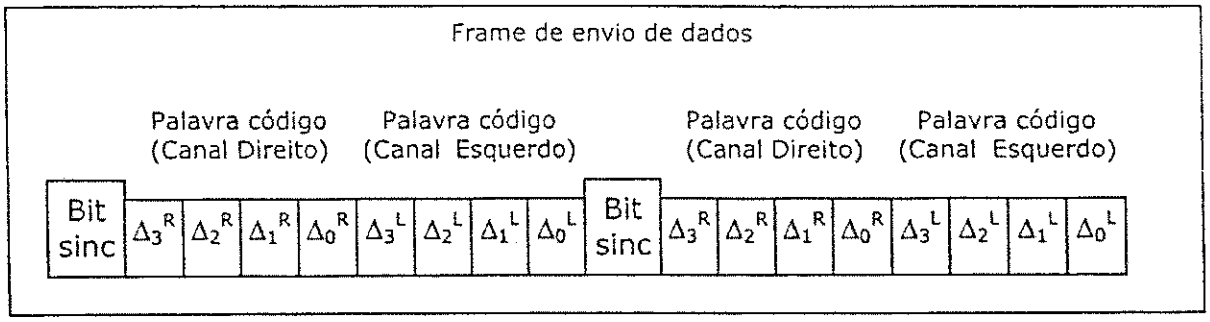


Figura 7.7 Frame de envio de dados. Note-se a inclusão do bit de sincronismo a cada byte de dados com informação dos dois canais de áudio. O bit 1 foi escolhido para bit de sincronismo.

Por outro lado, para que o passo de quantização passe de seu maior valor(2048) a seu menor valor(7), são necessárias 63 palavras do tipo 4'bx0xx. Exemplificaremos com o caso 4'b0000. Em intervalos de silêncio a presença de 0s na palavra quantizada é bem mais provável que a presença de 1s. Concluímos assim que, utilizando o bit 1 como bit de sincronismo, chega-se mais rapidamente ao sincronismo de frame no receptor.

Percebe-se assim que o circuito completo de modulação MSK é de baixa complexidade, sendo concebido a partir de uma pequena lógica combinacional, um contador de 4 bits e uma ROM de 64x8. Este circuito consumiu 32 macro-células, quando compilado para a síntese em família MAX7128.

7.3 Simulações do Modulador

Uma rotina de teste funcional para o modulador MSK foi escrita em linguagem Verilog. Esta rotina abre um arquivo de bytes aleatórios (arquivo texto), insere o bit de sincronismo para formar o frame de envio e escreve em outro arquivo os pontos, com precisão de 8 bits, resultantes da modulação MSK com 16 pontos por bit.

Os resultados no arquivo gerado forma visualizados no MATLAB e também comparados com os resultados gerados por um arquivo escrito em linguagem de alto nível (MATLAB). Todos os resultados encontrados a partir da simulação do circuito modulador MSK descrito em linguagem Verilog estão de acordo com o esperado.

7.4 Implementação do Demodulador

O demodulador MSK executa a recuperação dos bits a partir da comparação do tempo entre passagens por zero. Já foi dito anteriormente que uma das grandes vantagens deste demodulador é o fato de necessitar apenas de um circuito analógico: o comparador de tensão. Este comparador de tensão compara o sinal MSK detectado com o nível médio deste sinal, gerando portanto pulsos com largura igual ao período entre passagens por zeros do sinal MSK.

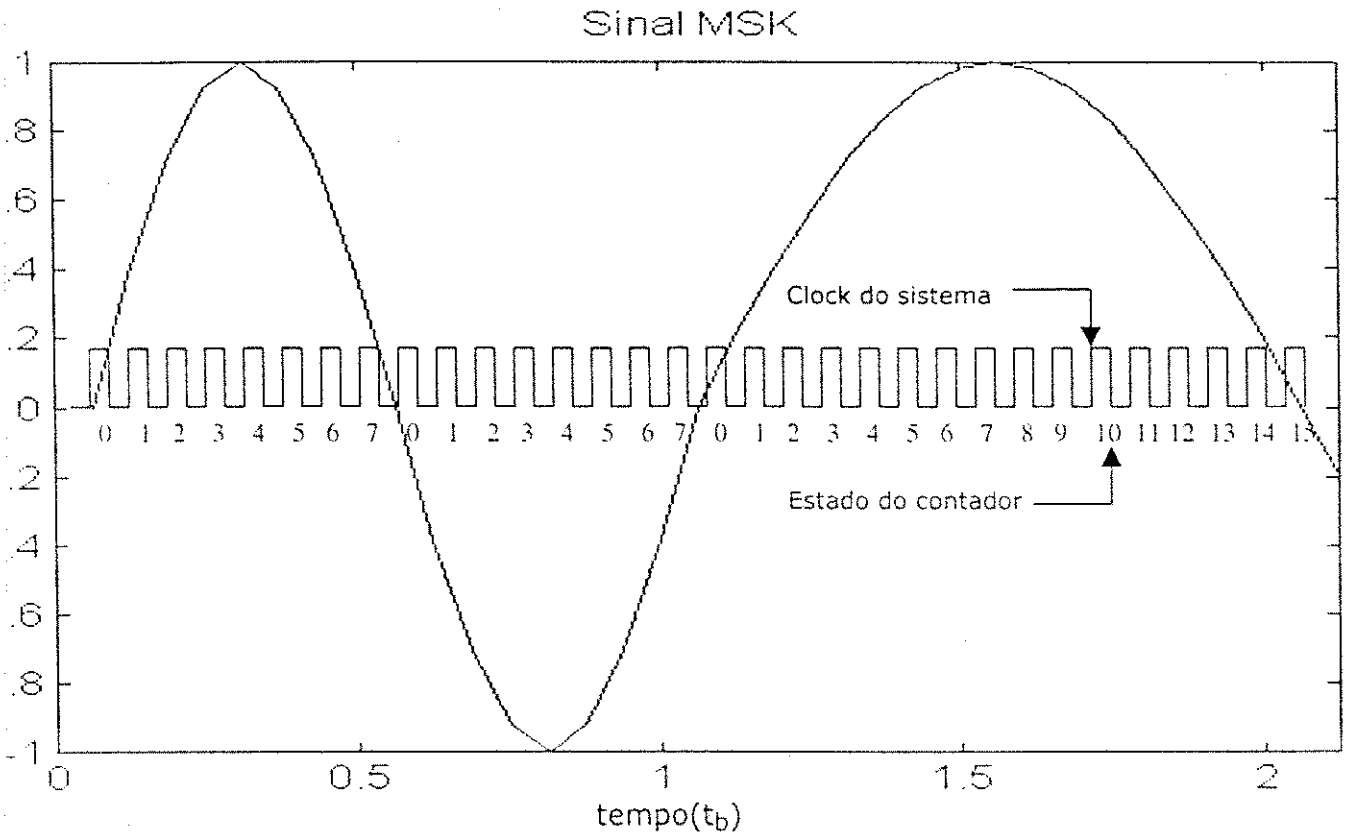
O circuito digital então irá comparar os tempos entre transições, que efetivamente representam as detecções de zeros. Um contador de 4 bits (16 ciclos de clock) foi utilizado para medir os tempos entre transições.

O número médio de ciclos clock entre zeros do sinal MSK foi escolhido de forma a não ser tão pequeno que dificultasse a decisão entre bit 0 e 1, quando na presença de sinal corrompido por ruído, nem ser tão grande que tornasse necessária a utilização de um clock de alta frequência, gerando aumento de consumo e interferência em outros sistemas.

Primeiro analisou-se a utilização de uma frequência de clock que resultasse em 8 contagens entre zeros para o bits 1 e 4 contagens entre zeros do sinal referente ao bit 0, o que se mostrou um número insatisfatório quando o sinal MSK corrompido foi simulado. Então se partiu para a análise de 16 ciclos de clock entre zeros para o bit 1. Conseguiu-se um melhor desempenho de demodulação quando da presença de ruído no sinal MSK que a obtida para o valor de 8 ciclos de clock para o referido bit.

A partir deste valor, o acréscimo de ganho de desempenho na demodulação não chega a ser compensador para o aumento significativo da frequência de clock. Desta forma, foi escolhido o valor de 16 ciclos de clock para cada bit no receptor.

Como o receptor utiliza a mesma frequência de clock do transmissor, para o bit 1 tem-se 16 ciclos de clock entre transições, enquanto que o bit 0 apresenta duas transições de 8 ciclos de clock cada (supondo recepção sem ruído), como mostra a figura 7.8. Desta forma, o limiar de decisão foi escolhido como sendo o período equivalente a 12 ciclos de clock, ou seja, sempre que o contador atingir o 12° estado, o bit recebido é necessariamente 1, enquanto que quando houver 2 transições onde o contador não atingiu 12, o bit recebido é 0(zero).



gura 7.8 Demodulação não-coerente do sinal MSK proposto. Ao longo da forma de onda associada ao bit 1, o contador utilizado no demodulador alcança um valor maior que 12 (limiar de escolha), enquanto que na forma de onda associada ao bit 0 o maior valor alcançado é 7 (sinal sem ruído).

Detector de transição

O contador deve ser zerado a cada transição. A partir de um registro (flip-flop) para armazenar o valor do sinal MSK disponível na saída do comparador de tensão e uma porta XOR, concebeu-se um detector de transição. Este detector irá gerar um pulso sempre que o sinal na saída do comparador de tensão mudar de valor, isto é, quando houver uma transição de 0 a 1 ou vice-versa.

A cada borda de subida do sinal de clock, o valor atual do sinal na saída do comparador de tensão é armazenado no registro. Desta forma, a saída deste registro estará sempre defasada do valor real do pulso, ou seja, qualquer mudança de nível neste sinal só será atualizada no registro na próxima borda de subida do clock. Haverá pois uma diferença entre o valor do registro e o valor real do sinal na saída do comparador sempre que este mudar de valor. Se estes dois sinais (registro e saída do comparador) alimentarem as entradas de uma porta XOR, esta irá gerar um pulso nesta ocasião, como mostra a figura 7.9 :

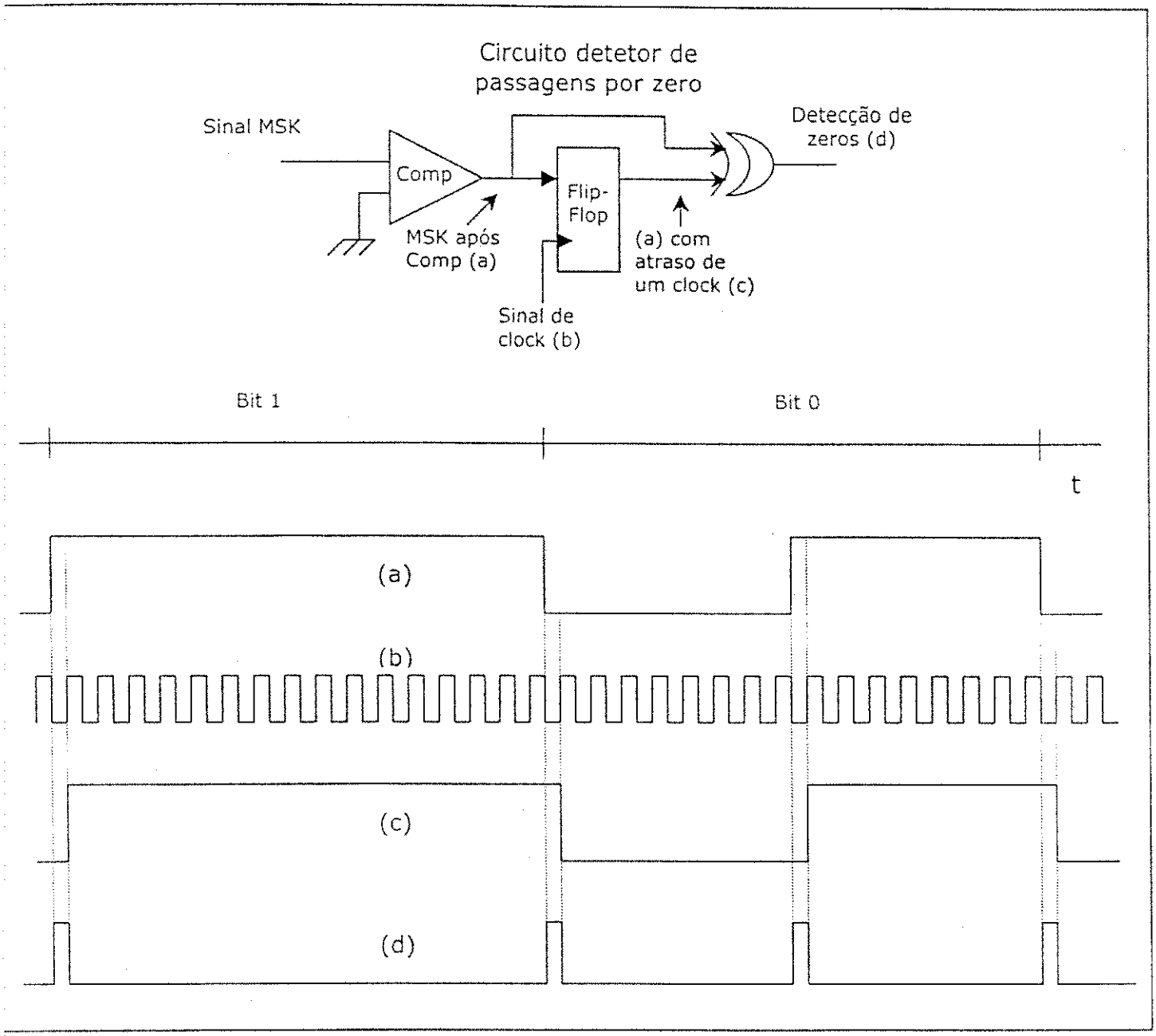


Figura 7.9 Circuito detetor de passagens por zero. O comparador de tensão "Comp" compara o sinal MSK com seu valor médio, gerando o sinal (a). Este sinal alimenta um flip-flop (registro) sensível à borda de subida do sinal de clock do sistema(b). Desta forma, o sinal na saída do flip-flop (c) é defasado de um ciclo de clock em relação ao sinal (a). Uma porta XOR é alimentada com os sinais (a) e (c), gerando pois os pulsos de detecção de zeros (d).

Imunidade ao ruído

A presença de ruído no sinal MSK tem 2 principais consequências do ponto de vista de demodulação: alterações de período entre transições e a formação de transições errôneas. A maior influência do ruído na demodulação, portanto, se dá nas proximidades da passagem por zero. O demodulador foi desenvolvido prevendo estas duas situações.

Variação do período entre transições

Como o limiar de escolha é o período correspondente a 12 ciclos de clock, variações no período entre transições inferiores a 4 ciclos de clock não ocasionarão erros. Um problema decorre do fato de haver aumento do período correspondente ao bit 1 (16 ciclos clock), pois sem haver controle especial do contador, quando este receber um pulso com período maior que 16 ciclos de clock, passaria do estado 15 para o estado 0, e começaria a contar novamente. Para evitar esta situação, utilizou-se um circuito de "congelamento" do contador, que faz com que toda vez que o contador atinja o limiar de escolha, seu valor permaneça inalterado até que um nova passagem por zero seja detectada.

Transições errôneas

No caso das transições errôneas, um circuito especial de proteção foi gerado. Este circuito faz com que sempre que haja uma transição e o contador estiver num estado menor que 3, a passagem por zero anterior seja descartada, passando a ser referência a passagem por zero atual.

O circuito demodulador MSK é bastante robusto em termos de imunidade a ruído, como será visto na seção *simulações do demodulador MSK*. Além disso, requer o uso de apenas um componente analógico para recuperação dos bit, o comparador de tensão, e é de baixa complexidade.

7.5 Simulações do demodulador MSK

No caso do demodulador MSK, a simulação inicial foi feita em linguagem C. O código escrito recupera os bits enviados a partir da análise de pontos do sinal MSK contidos em um arquivo previamente gerado.

Em seguida, estes resultados foram comparados com os obtidos a partir da simulação do circuito demodulador descrito em VERILOG, como descrito nas seções acima. Não houve discrepância entre resultados.

Feito isto, a rotina analisa o número de pontos entre passagens por zero e constrói um histograma com todos os valores encontrados ao longo do sinal MSK. Obviamente, para o sinal MSK puro (sem ruído e sem passagem por filtros) este histograma contém apenas dois valores para o número de pontos entre passagens por zero, 8 e 16, como mostra a figura 7.10:

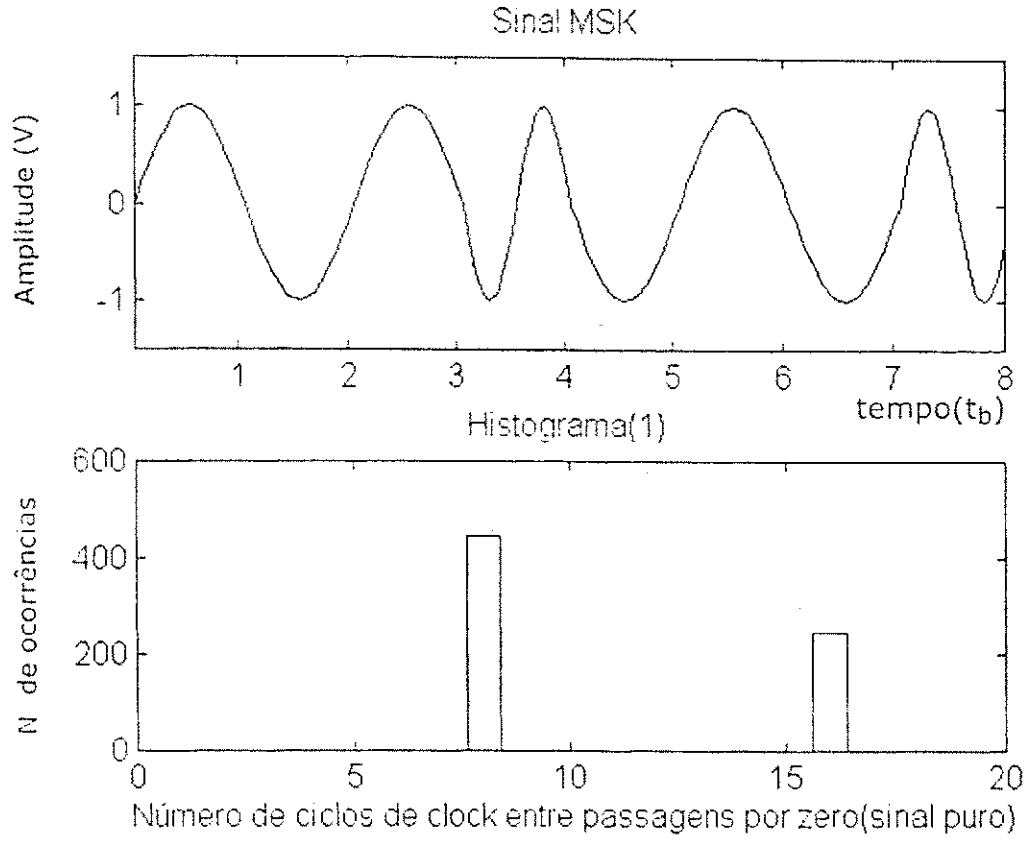


Figura 7.10 Análise do número de ciclos de clock entre passagens por zero do sinal MSK puro (sem ruído). Nota-se que se observam apenas dois valores distintos: o valor de 16 ciclos de clock referente a um bit 1, e o valor de 8 ciclos de clock, referente ao bit 0.

A partir do sinal MSK que foi gravado, prossegue-se assim na análise de casos mais próximos da realidade. Filtros passa-baixas foram implementados, para simular o comportamento de um canal de transmissão escolhendo-se a ordem do filtro e a frequência de corte em relação à frequência de amostragem. Foram utilizados filtros butterworth.

Em seguida o sinal puro foi filtrado, e o resultado foi gravado em outro arquivo. Analisa-se novamente o número de pontos entre passagens por zero ao longo do sinal, a fim de avaliar o efeito de filtragem na recuperação dos bits. Observa-se que a filtragem promove o espalhamento dos valores de número de ciclos de clock entre passagens por zero, sendo encontrados valores diferentes dos encontrados anteriormente, 8 e 16, como mostra a figura 7.10:

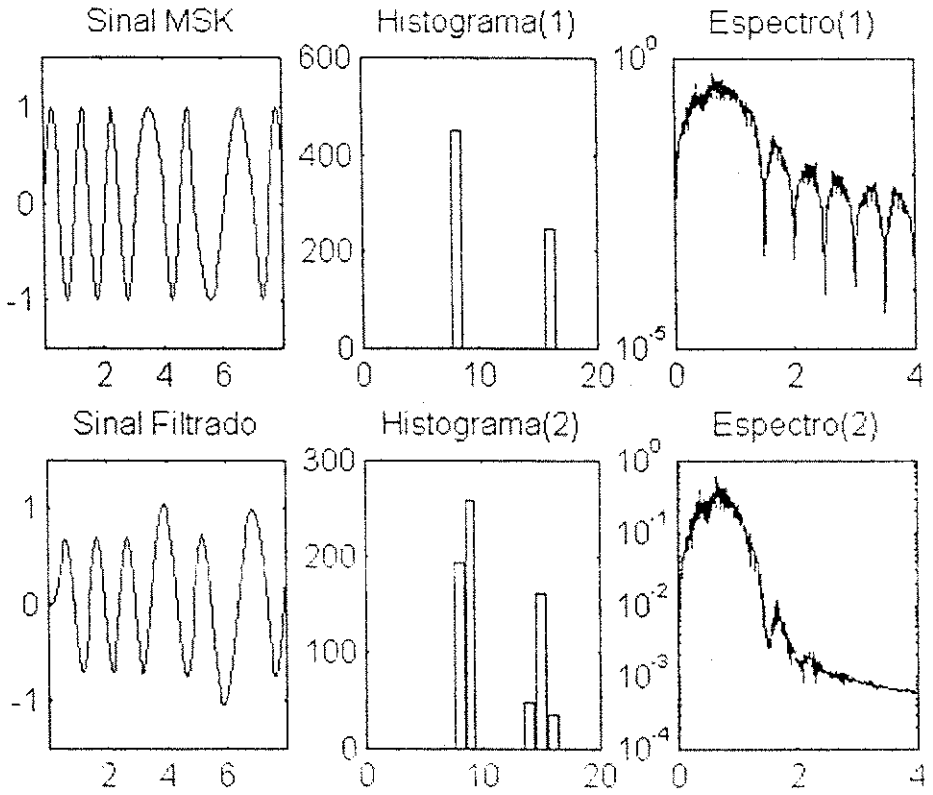


Figura 7.11 Efeito da filtragem do sinal MSK (filtro butterworth de 3ª ordem e freqüência de corte em 1 $Hz/_{bit/s}$) no número de ciclos de clock entre passagens por zero. Note-se os novos valores encontrados para o número de ciclos de clock entre passagens por zero. Escala horizontal dos gráficos "Espectro(1)" e "Espectro(2)" em $Hz/_{bit/s}$

Uma vez feito isso, procedeu-se à análise da influência de ruído aditivo na demodulação MSK. Para isso, implementou-se uma rotina de geração de ruído gaussiano branco aditivo AWGN. A figura 7.12 mostra distribuição do ruído gerado pela rotina:

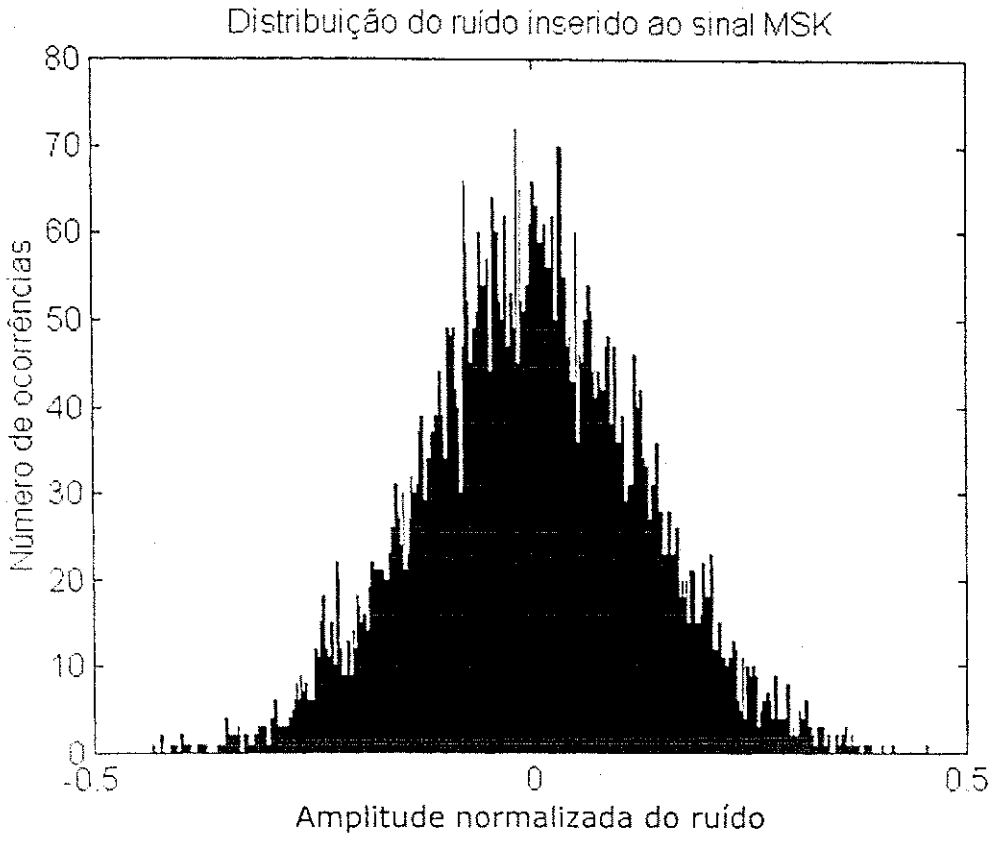


Figura 7.12 Distribuição do ruído branco aditivo gaussiano, introduzido no sinal MSK para a simulação do canal de transmissão.

Esta rotina abre um arquivo com pontos do sinal MSK e baseado numa relação sinal-ruído desejada para simular o canal de transmissão, gera o ruído AWGN correspondente, introduz o ruído no sinal original e grava um arquivo de saída com o sinal corrompido.

A relação sinal-ruído é a relação entre o valor RMS de tensão do sinal MSK e o valor RMS de tensão do ruído, de forma que uma relação sinal-ruído de 20dB equivale a dizer que o valor RMS de tensão do sinal MSK é 10 vezes maior que o valor RMS da tensão do ruído.

A figura 7.13 mostra o sinal resultante da inserção de ruído AWGN ao sinal MSK (relação sinal-ruído de 15 dB) :

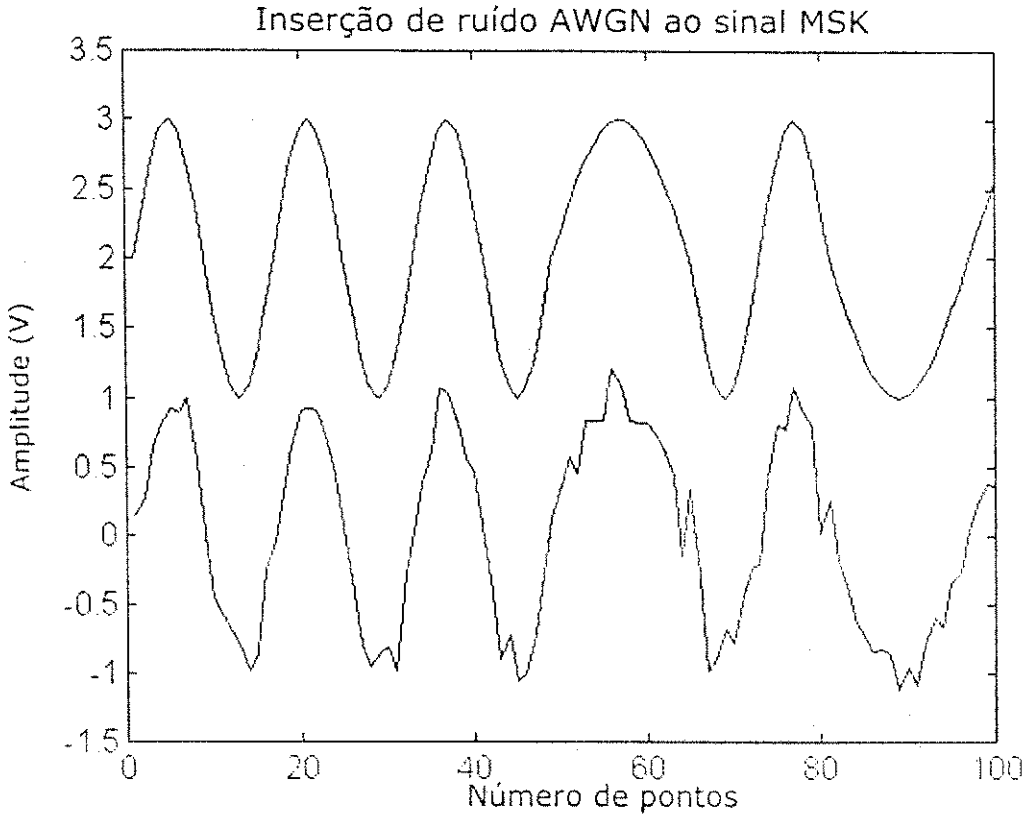


Figura 7.13 Inserção de ruído AWGN ao sinal MSK. O sinal resultante apresenta relação sinal-ruído de 15 dB. Obs: 16 pontos por bit; para efeitos de visualização, foi adicionado um nível DC de +2V ao sinal MSK puro.

Outra rotina implementada constrói um histograma com os valores de número de ciclos de clock entre passagens por zero do sinal MSK puro e do sinal MSK corrompido com ruído AWGN. Os resultados estão expostos na figura 7.14 para um sinal corrompido com relação sinal ruído de 15 dB.

Já a figura 7.15 mostra o efeito da filtragem do sinal MSK corrompido com ruído AWGN (relação sinal ruído de 8dB), enquanto a figura 7.16 mostra o histograma com o valor de número de ciclos de clock entre passagens por zeros dos dois sinais da figura 7.15.

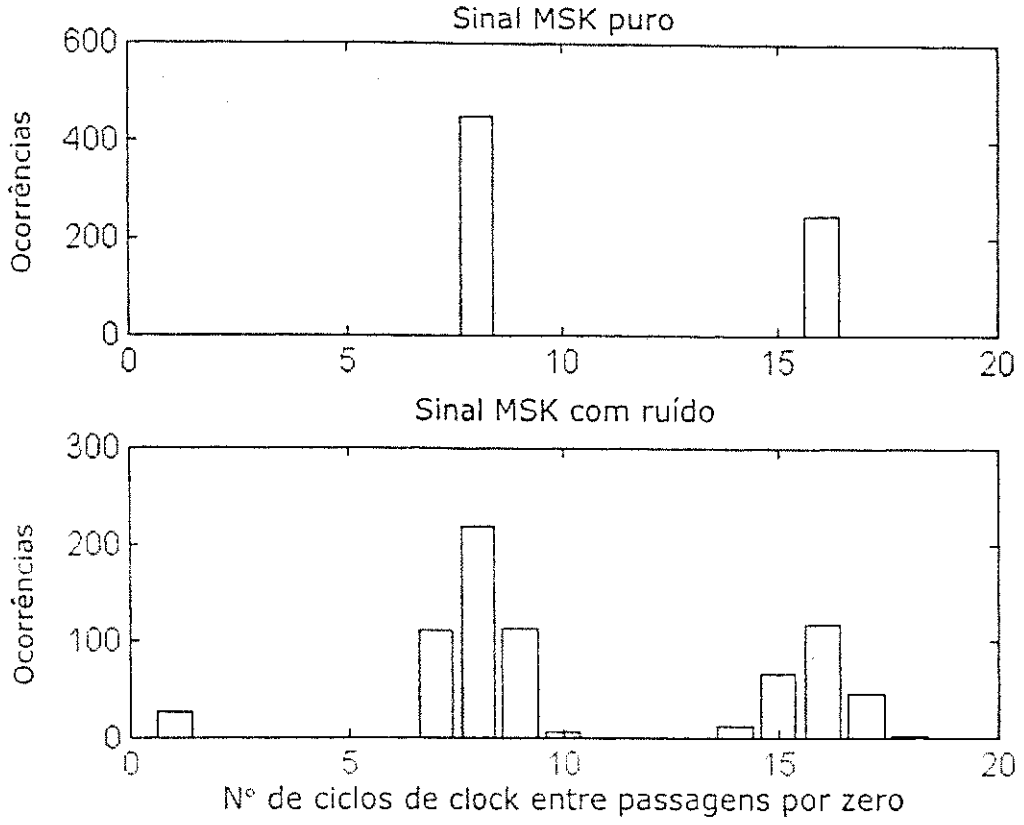


Figura 7.14 Histograma do número de ciclos de clock entre passagens por zero do sinal MSK. No gráfico superior é analisado o sinal MSK sem ruído. Mostra-se no gráfico inferior como a adição de ruído provoca o espalhamento dos valores encontrados no gráfico superior. Além disso, provoca também o aparecimento de novas passagens por zero com período de 1 ciclo de clock.

A partir disso, a adição de ruído ao sinal MSK foi prevista na funcionalidade do circuito descrito em VERILOG. O processo de análise de resultados foi inicialmente gráfico, com a utilização do editor de formas de onda do ambiente MAXPLUSII. Com este editor é possível construir graficamente os valores encontrados após a comparação do sinal MSK com seu valor médio. Desta forma, variações de tempo entre passagens por zero e geração de passagens errôneas foram simuladas, e alguns ajustes de concepção, como o caso do circuito de proteção contra transições errôneas do sinal MSK já citado na seção 6.4, tiveram de ser implementados.

O processo de iteração na busca de melhores resultados de desempenho foi bastante exaustivo. Ao final deste processo, o arquivo de descrição em Verilog foi compilado tendo como entrada pontos do arquivo contendo sinal MSK corrompido por ruído gaussiano aditivo limitado em banda.

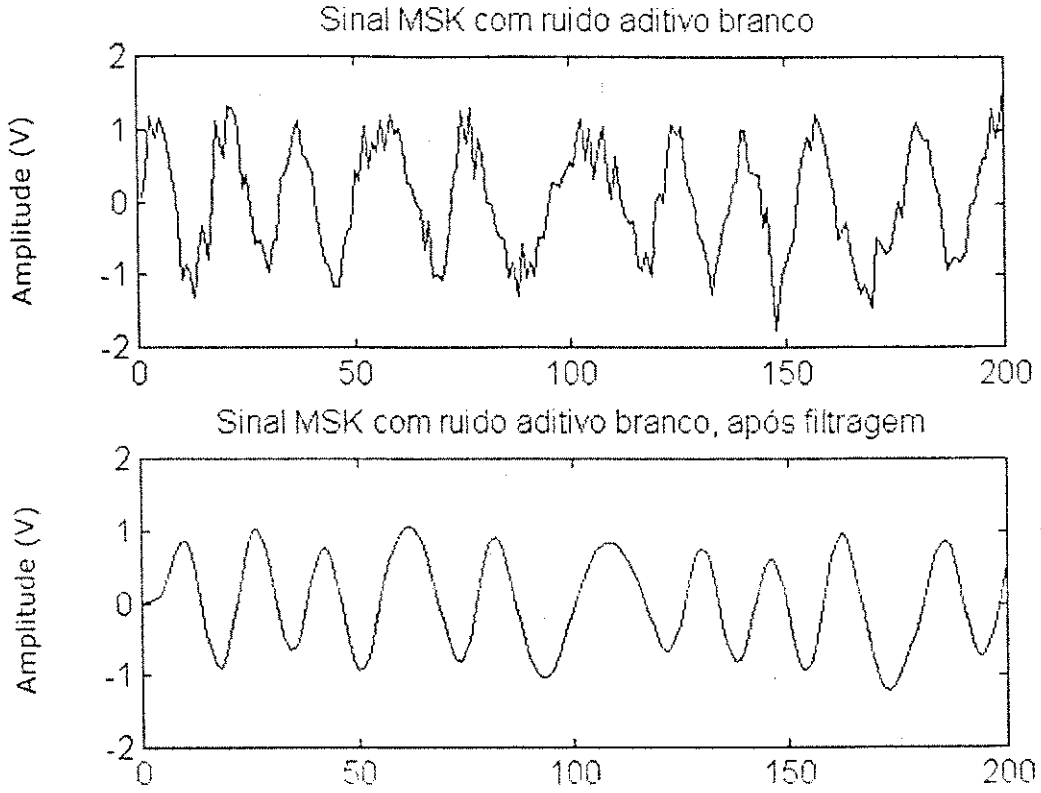


Figura 7.15 Gráfico superior: sinal MSK corrompido com ruído AWGN, apresentando relação sinal ruído de 8 dB. Gráfico inferior : resultado da filtragem do sinal MSK corrompido, com um filtro butterworth de terceira ordem, com freqüência de corte de 1.25 $Hz/_{bit}$. Escala horizontal : 16 pontos/bit.

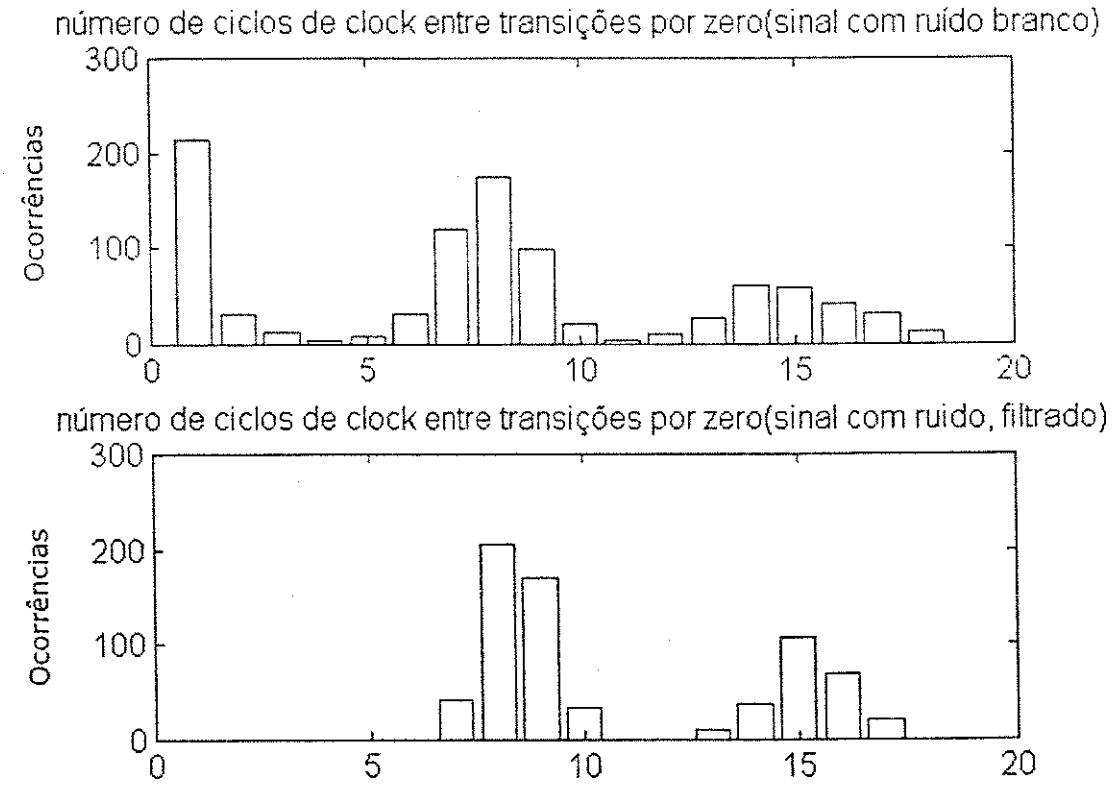


Figura 7.16 Gráficos do número de pulsos de clock entre os zeros dos sinais da figura 7.15. Note-se que no gráfico superior bits errôneos serão detectados, por conta da ocorrência de pulsos com período de 3 ciclos de clock

Percebeu-se ao longo das simulações, que para o ruído em questão, o demodulador recupera corretamente os bits para valores de relação sinal-ruído maiores que 8 dB do sinal MSK (após filtragem). Decrescendo esta relação a partir deste valor, a correta recuperação já não é mais alcançada.

Um problema grave ocorre quando o bit errôneo recuperado é o bit de sincronismo de frame. Nesta situação, haverá perda de sincronismo de frame, ou seja, como o bit de sincronismo foi invertido, o demodulador mudará a referência para o início do byte de dados. Os dados subsequentes (bytes) recuperados serão errôneos e causarão perda de sincronismo transmissor/receptor do sistema ADPCM. O sincronismo de frame só será readquirido após alguns frames, como descrito na seção 7.1. Após isso, haverá também um intervalo de tempo variável para a aquisição de sincronismo transmissor/receptor do sinal ADPCM.

7.6 Conclusões

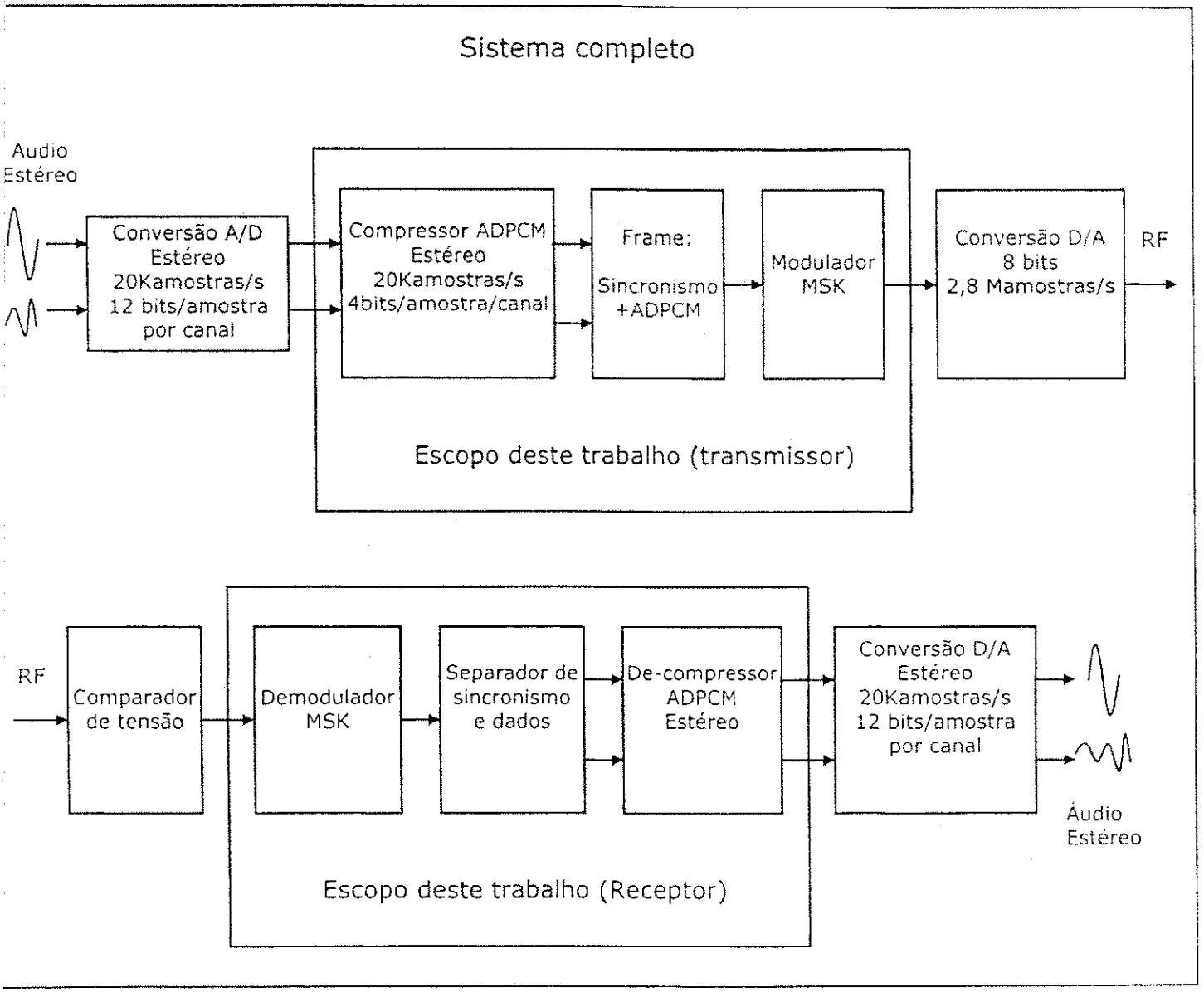
A escolha da modulação digital em banda básica teve como base os requerimentos já explícitos no início da seção. Além disso, o modo de implementação teve como principais objetivos a robustez do conjunto modulador/demodulador na presença de ruído aditivo e a diminuição de processamento analógico de sinais.

Desta forma, ao invés do modulador ser sintetizado a partir de um oscilador controlado (modulador em frequência com índice de modulação 0.5), o sinal MSK foi gerado a partir de pontos armazenados em ROM. No caso do demodulador, a característica de diferentes períodos entre zeros do sinal MSK foi aproveitada.

Os resultados de simulação mostram que o demodulador consegue recuperar corretamente os bytes transmitidos, a partir de um sinal MSK com relação sinal ruído de 8 dB (sinal com ruído aditivo branco, que é posteriormente filtrado com filtro passa-baixas). Este valor de relação sinal-ruído no áudio recuperado nos moduladores analógicos (p.ex. Moduladores FM) torna-o intolerável pelo usuário.

Tanto o circuito modulador MSK quanto o demodulador MSK consumiram menos que 35 macro-células no dispositivo MAX7128, demonstrando quão compacta foi a implementação.

8. Circuito Completo



ura 8.1 Gráfico do sistema completo de transmissão digital de áudio estéreo proposto. Note-se o escopo deste trabalho já implementado em linguagem Verilog. Por RF entenda-se um par transmissor/receptor que garanta uma relação sinal-ruído de 8 dB do sinal MSK a ser entregue ao demodulador MSK(após filtragem).

8.1 Detalhes de Implementação

Uma vez de posse dos sub-módulos ADPCM, D_ADPCM, MSK e D_MSK, o transmissor e o receptor foram sintetizados (vide escopo deste trabalho na figura 8.1). Um sinal de carregamento foi é gerado pelo transmissor assim que compressão ADPCM termina, a fim de indicar o início de um novo processo de conversão $A/_D$. Este sinal será usado pelo conversor $A/_D$. Da mesma forma, no receptor foi implementado um sinal para indicar um novo carregamento com os 12 bits de cada canal de áudio nos conversores $D/_A$.

8.2 Simulação

ADPCM+MSK

De posse dos módulos ADPCM e MSK simulados separadamente, passou-se à simulação do conjunto ADPCM+MSK. Nesta etapa foram utilizados os arquivos de descrição dos circuitos em questão e códigos escritos também em linguagem Verilog para geração de estímulos aos circuitos. Os resultados gerados por estes circuitos foram comparados com os resultados gerados por rotinas em linguagem C através de comando "fc"do DOS, usando o mesmo procedimento já citado na seção 6.3, simulações do compressor de áudio.

Um arquivo de amostras aleatórias de áudio foi usado para transmissão. Este arquivo continha valores com precisão de 12 bits (formato de 4 dígitos hexadecimais, um deles descartado).

Uma rotina de teste escrita em Verilog fornece as duas primeiras amostras deste arquivo (uma amostra do canal L e outra do canal) à entrada do circuito ADPCM+MSK, que efetuou a compressão ADPCM e modulou o frame de dados gerado. O próprio circuito gera um sinal de carregamento para o conversor A/D, como já foi explicado, que é entendido por este arquivo de estímulo como o sinal para carregamento de novo par de amostras. Este processo é repetido até que o arquivo contendo amostras seja completamente modulado.

D_MSK+ADPCM

O arquivo gerado pelo circuito supracitado contém os pontos do sinal MSK referente à modulação do arquivo de amostras. De posse deste arquivo, o circuito D_MSK+ADPCM foi testado.

O circuito D_MSK+ADPCM tem como sinais de entrada o sinal resultante da comparação do sinal MSK com seu valor médio, e o sinal de clock. Como saída, tem-se os sinais de áudio R e L, com precisão de 12 bits.

Um arquivo de estímulo foi escrito em linguagem Verilog. Este arquivo compara os pontos do arquivo do sinal MSK com um valor constante (média do sinal MSK) e insere o sinal digital resultante desta comparação na entrada do bloco D_MSK+ADPCM.

A cada pulso de clock, um novo ponto do sinal MSK é comparado e entregue ao bloco. A saída deste circuito (amostras R e L com 12 bits) é gravada em um arquivo de saída.

Comparamos então este arquivo de saída do demodulador com um arquivo gravado por rotina em C, que apenas comprimiu o as amostras do arquivo inicial, e gerou um arquivo de saída com os valores a serem efetivamente recuperados pelo receptor. Alguns ajustes foram necessários no que diz respeito a correções de escrita de arquivo devido ao processo de reset, mas a funcionalidade estava de acordo com o esperado.

8.3 Conclusões

Com isto concluímos que a descrição comportamental do circuito digital proposto já está pronta para a integração com os blocos analógicos necessários à concepção do transmissor e receptor. Após as simulações, concluiu-se que os circuitos compilados funcionam corretamente desde que a relação sinal/ruído na transmissão MSK seja superior a 8 dB. O circuito usa um sinal de relógio relativamente lento(2.8 MHz) e tem baixa complexidade, indicando um baixo consumo de energia elétrica e aplicabilidade num sistema móvel que utiliza pilhas.

9. Trabalhos futuros

Ao longo do desenvolvimento dos circuitos propostos neste trabalho, alguns testes de especificação não puderam ser executados por falta de tempo hábil à finalização do trabalho. Um exemplo disto é a quantização da relação sinal-ruído de áudio na saída do demodulador ADPCM, quando da recuperação correta dos bits por parte do circuito de demodulação MSK. Outro teste de especificação pode ser feito na busca da curva de taxa de erro de bit versus relação sinal-ruído do sinal MSK. Estas especificações podem ser geradas em trabalhos futuros, para a especificação do circuito integrado final.

Além disso, deve-se proceder à síntese de leiaute do circuito digital descrito em linguagem Verilog, para que se saiba com precisão a área em silício necessária à sua implementação. Isto deve ser feito com ferramenta profissional e utilizando tecnologia atual, para que possam ser feitas cotações de preço junto a fábricas de semicondutores.

Para a concepção da parte analógica do circuito, duas metodologias de projeto podem ser seguidas: a utilização de macro-células já prontas e disponibilizadas pelos fabricantes de circuitos integrados, ou ainda o projeto *full-custom*, com o design particular desde o início da concepção.

A implementação do controle do receptor: este controle pode ser facilmente implementado a partir da inserção de 1 bit ao frame de envio de dados, totalizando assim 10 bits no frame, ou a partir da utilização de um novo símbolo da modulação MSK.

10. Referências Bibliográficas

1. Projeto EUREKA 147-DAB. Eletronic Engineering. Abril de 1998

2. Pan,Davis yen. Digital Audio Compression. Digital Technical Journal Vol.5 N°2 1993

3. Ministério da Comunicações. Portaria 211 de 10 de Novembro de 1983

4. http://www.rohm.com/products/databook/com/pdf/bu8710aks.pdf

5. BA1404 FM Transmitter. http://www.rohm.com/

6. Haykin, Simon S. Digital communications. 1ª edição. Ie-Wiley. 1988

7. Rabiner, Lawrence R. e Schafer, Ronald W. Digital Processing of Speech Signals. Prentice Hall. 1978

8. Proakis, John G. e Deller, John R. Discrete Time Processing of Speech Signals. 1ª edição. MacMillan Books. 1993

9. P. NOLL, "Non adaptative and adaptative DPCM for speech signals" Polytech. Tijdschr. Ed. Elektrotech/Elektron (The Netherlands), N°19,1972

10. Bellanger, Maurice. Digital Processing of Signals. 2° edição. John Wiley Professio. 1989

11. Razavi, Behzad. RF Microeletronics. Prentice Hall. 1998

12. Jayant, N. S. Adaptive quantization with one word memory. Bell System Tech Journal. Setembro de 1973

13. Motion Picture Experts Group, www.mpeg.org

14. http://www.owlnet.rice.edu/~engi202/capacity.html

15. SA626 Receptor FM Faixa-Larga Philips, www.philips.com

16. http:\\robotics.eecs.berkeley.edu/~sastry/ee20/modulation.node6.html

17. 56Kbaud MSK modem , www.grapes.org/paper.html

18. IMA ADPCM, Interactive Multimedia Association

19. Practical GMSK data transmission, www.mxcom.com

20. Weste, Neil H. E. Principles of CMOS VLSI Design. Second edition. 1992

21. Santivañez, Javier e Alcaim, Abraham. ADPCM-MQ em sub-banda para a codificação digital de áudio na faixa de 7KHz

22. www.altera.com

23. http://www.iaik.at/tec/Materialien.htm

24. P. Noll, "A Comparative Study of Various Schemes for Speech Encoding", Bell System Tech Journal, Vol 54, N° 9, p.1597-1614. Novembro de 1975