



UNIVERSIDADE DE COIMBRA

FACULDADE DE CIÊNCIAS E TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

Modulador de Magnitude em DSP

Irina Filomena Fernandes Varela Robalo

Dissertação de Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Júri:

Doutor Jorge Manuel Miranda Dias (Presidente)
Doutor Fernando Manuel dos Santos Perdigão (Vogal)
Doutor Vítor Manuel Mendes da Silva (Orientador)
Mestre Marco Alexandre Cravo Gomes (Orientador)

Coimbra

Julho de 2010

Modulador de Magnitude em DSP

Irina Filomena Fernandes Varela Robalo

Coimbra

Julho de 2010

Agradecimentos

Durante todo o meu percurso universitário, várias foram as pessoas que me acompanharam e me incentivaram.

Os meus primeiros agradecimentos são dirigidos aos meus orientadores, Professor Doutor Vítor Silva e Professor Mestre Marco Gomes, pela total dedicação, disponibilidade, incentivo e apoio dado ao longo da realização desta dissertação.

Ao Departamento de Engenharia Electrotécnica e de Computadores e sobretudo ao Instituto de Telecomunicações, que disponibilizaram as instalações e todo o material necessário para a elaboração desta dissertação. Aos meus colegas do laboratório de processamento de imagem agradeço todo o espírito de companheirismo e o bom ambiente proporcionado.

Aos meus amigos com quem partilhei momentos únicos e inesquecíveis que vão perdurar por toda a vida.

Aos meus pais e familiares, principalmente a minha avó, pelo apoio e pela confiança que depositaram em mim.

A minha amiga e companheira Margareth pela paciência, pelo incentivo, pela amizade, pelos bons e maus momentos passados juntos ao longo desses anos.

Um agradecimento especial à minha mãe, a razão da minha existência.

Irina Robalo

Resumo

A eficiência espectral e o consumo de potência constituem dois parâmetros fundamentais a ter em conta no projecto de qualquer sistema de comunicação digital. As técnicas de modulação e a filtragem FIR do tipo *root-raised cosine* (RRC) de baixo *roll-off*, assumem um papel cada vez mais importante nas comunicações sem fios com vista à maximização da eficiência espectral dos sistemas. A consequência disto é um aumento da razão entre a potência de pico do sinal a transmitir e a sua potência média (*Peak-to-Average Power Ratio* - PAPR). Elevados valores de PAPR impõem elevados requisitos de linearidade nos amplificadores de potência usados nos transmissores, com consequente aumento do consumo de potência do sistema. Um elevado PAPR dificulta ainda o projecto dos conversores digitais - analógicos (DAC), devido à elevada gama dinâmica do sinal.

A realização desta dissertação tem como objectivo a validação de um método proposto recentemente, controlo de magnitude (*Magnitude Modulation*-MM) do sinal modulado com vista à redução do PAPR do sinal de banda limitada a transmitir. Este sistema foi designado por, *Multistage Polyphase Magnitude Modulation* (MPMM). O sistema foi desenvolvido num processador digital de sinal (DSP) TMS320C6713 (vírgula flutuante) da *Texas Instruments* (TI). Foram exploradas as capacidades de processamento paralelo deste DSP, de forma a maximizar o desempenho do sistema.

Palavras-Chave: *Root-raised cosine filter* (RRC), Modulador de Magnitude (MM), Modulação de Magnitude Polifásica (MPMM), razão entre a potência de pico e a potência média (PAPR), processador digital de sinal (DSP), decomposição polifásica.

Abstract

Spectral efficiency and power consumption are two of the major issues in practical communications system design. Modulation techniques and very low roll-off root-raised cosine filters are starting being used in order to maximize systems' spectral efficiency. However this leads to an increase on the transmit signal Peak-to-Average Power Ratio (PAPR). A high PAPR puts a stringent linearity requirement on the transmitter's high power amplifier (HPA) and a consequent decrease of system's power efficiency. Besides it complicates the project of digital-to-analog converters (DAC), due to the large signal's dynamic range, which allied to HPA's requirements increases system's costs.

Recently, a new efficient Magnitude Modulation (MM) technique was proposed in order to control envelope's power peak of the bandwidth limited modulated transmitted signal. This system was designated by Multistage Polyphase Magnitude Modulation (MPMM). The dissertation consist into the development of a proof-of-concept demonstrator of the MPMM system on digital signal processor (DSP) TMS320C6713 (floating point) from Texas Instruments. The parallel processing capacities of this DSP should be explored in order to maximize system performance.

Keywords: Root-raised cosine filter (RRC), Magnitude Modulation (MM), Multistage Polyphase Magnitude Modulation (MPMM), Peak-to-Average Power Ratio (PAPR), Digital signal processing (DSP), polyphase decomposition.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	v
Glossário	vii
1 Introdução	1
1.1 Organização da Dissertação	2
2 Modulação de Magnitude	5
2.1 Modulador de magnitude	6
2.2 Modulação de magnitude polifásica	8
2.2.1 Decomposição polifásica.....	10
2.2.2 Controlo da envolvente do sinal.....	11
3 Modulador de Magnitude em DSP	17
3.1 Maximização do desempenho do sistema.....	18
3.2 Demonstrador MPMM em tempo real	19
3.2.1 Gerador de bits aleatórios.....	21
3.2.2 Estrutura paralela dos filtros polifásicos	22
3.2.3 Cálculo dos coeficientes MM.....	27
3.2.4 Filtragem polifásica dos filtros RRC.....	32

4 Resultados Experimentais	35
4.1 Tempo de processamento.....	35
4.2 Visualização da saída do sistema MPMM.....	36
5 Conclusões e considerações finais	39
Anexo A – DSP TMS320C6713	41
Anexo B – Representação de números reais	50
Anexo C – Endereçamento circular	52
Bibliografia	55

Lista das Figuras

2.1	Sistema genérico de transmissão incluindo o MM	6
2.2	Sistema MPMM seguido do bloco de filtragem do RRC	9
2.3	Exemplo de um Filtro RRC	11
2.4	Sistema MPMM com único ramo	12
3.1	Diagrama de blocos do sistema desenvolvido	20
3.2	Diagrama de um gerador de 32 bit	22
3.3	Cálculo das sequências $a_i[n]$	23
3.4	Cálculo das sequências $b_i[n]$	23
3.5	Disposição dos dados na memória	24
3.6	Fluxograma da filtragem polifásica para as sequências $a_i[n]$ e $b_i[n]$	25
3.7	Diagrama funcional de apoio à Figura 3.6	26
3.8	Fluxograma do cálculo da função $f(A,a,b)$ para a modulação BPSK	29
3.9	Disposição dos símbolos de entrada para o caso QPSK	30
3.10	Fluxograma do cálculo da função $f(A,a,b)$ para a modulação QPSK	32
3.11	Decomposição da componente polifásica $E_i(z)$	33
3.12	Sistema desenvolvido para a filtragem polifásica do filtro RRC	34
4.1	Segmentos do sinal à saída do filtro RRC para o sistema MPMM_BPSK	37
4.2	Segmentos da componente real do sinal à saída do filtro RRC para o sistema MPMM_QPSK	38

Lista das Tabelas

2.1	PAPR devido à contribuição da constelação	7
2.2	PAPR devido à contribuição do filtro RRC.....	8
4.1	Tempo de processamento	36

Glossário

ADC	Analog-to-Digital Conversion
ALU	Arithmetic Logic Unit
APSK	Amplitude and Phase Shift Keying
BER	Bit Error Rate
BIOS	Basic Input Output system
BPSK	Binary Phase Shift Keying
CCS	Code Composer Studio
CPLD	Complex Programmable Logic Device
CPU	Central Processing Unit
DAC	Digital-to-Analog Conversion
DSK	Digital Signal Processor Starter Kit
DSP	Digital Signal Processor
EDMA	Enhanced Direct Memory Access
EMIF	External Memory Interface
FIR	Finite Impulse Response
GPIO	General Purpose Input Output
HPA	High Power Amplifier
HPI	Host Port Interface
MAC	Multiply – Accumulate

MM	Magnitude Modulation
MFLOPS	Million Floating-point Operation Per Second
MIPS	Millions Instructions Per Second
MPMM	Multistage Polyphase Magnitude Modulation
PAPR	Peak-to-Average Power Ratio
PLL	Phase Locked Loop
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RRC	Root Raized Cosine
SC	Single Carrier
SDRAM	Synchronous Dynamic Random Access Memory
TI	Texas Instruments
VLIW	Very Long Instruction Word
VSAT	Very Small Aperture Terminal

CAPÍTULO 1

Introdução

A razão entre a potência de pico do sinal a transmitir e a sua potência média, PAPR (*Peak-to-Average Power Ratio*) sempre foi uma preocupação importante em sistemas de comunicação que utilizam canais de banda limitada. A limitação da largura de banda do sinal a transmitir é realizada tipicamente com base num filtro *root-raised cosine* (RRC) [1, 2, 3], o que leva a um aumento do PAPR. Um elevado PAPR tem como consequência a diminuição da eficiência do amplificador de saída HPA (*High Power Amplifier*) devido à necessidade de usar *back-off* como forma de evitar a sua saturação. O problema torna-se crítico em sistemas com restrições de baixo consumo de potência, tais como, terminais móveis de satélite VSAT (*Very Small Aperture Terminal*) e outros dispositivos portáteis.

Recentemente, baseado no conceito de *Magnitude Modulation* (MM) [1, 2, 3], foi proposto um método eficiente para o controlo da envolvente do sinal de banda limitada a transmitir. Este método aplicado a sistemas de uma portadora (*single carrier - SC*) tem como principal objectivo a redução da contribuição para o PAPR resultante da filtragem RRC. Este permite uma melhoria significativa na eficiência de potência dos sistemas sem aumentar o BER (taxa de bits errados) [1, 2, 3].

A redução do PAPR, empregando o conceito MM, assenta no ajuste da amplitude de cada símbolo modulado a fornecer ao filtro RRC, a fim de controlar a excursão do sinal transmitido evitando picos indesejáveis no sinal de saída.

Os coeficientes MM aplicados aos símbolos são calculados em tempo real por uma estrutura de filtragem polifásica que só depende da resposta ao impulso do RRC. Este sistema foi designado por *Multistage Polyphase Magnitude Modulation* (MPMM) [1, 2].

O método proposto é independente do tipo de modulação SC, permitindo assim a generalização do conceito MM (inicialmente proposto para modulações de amplitude constante QPSK e 8-PSK) [3, 19].

Nesta dissertação foram desenvolvidos e otimizados sistemas MPMM para modulações BPSK (*Binary Phase Shift Keying*) e QPSK (*Quadrature Phase Shift Keying*). Foi utilizada como plataforma de desenvolvimento o processador digital de sinal DSP (*Digital Signal Processor*) TMS320C6713 (vírgula flutuante) [4, 9, 14] da *Texas Instruments* (TI).

O principal objectivo desta dissertação é a validação em tempo real do método MPMM, que, como já foi referido, tem como finalidade a redução eficiente do PAPR do sinal de banda limitada a transmitir. Na sua realização fez-se um estudo exaustivo das potencialidades do DSP a nível do processamento paralelo de forma a maximizar o desempenho do sistema. No âmbito da decomposição polifásica do filtro RRC, foi utilizado um método que permite realizar varias convoluções em paralelo, aumentando desta forma, a eficiência aritmética do algoritmo (leitura de dados e coeficientes do filtro RRC empacotados em palavras de 64 bits).

1.1 Organização da Dissertação

Esta dissertação é composta por 5 capítulos que abordam o trabalho realizado. Nela são abordadas os pontos essenciais do método MPMM, assim como, a sua validação utilizando um DSP de vírgula flutuante.

No capítulo 2 são apresentados os conceitos fundamentais sobre o método MPMM, nomeadamente, o seu objectivo e suas características.

No capítulo 3 é apresentado o desenvolvimento do método MPMM no DSP. De realçar as principais características do DSP, a implementação dos sistemas de filtragem multi-ritmo para

o módulo MPMM e RRC, realização do método numérico a utilizar na resolução da equação de segundo grau com raízes reais pertencentes a $]0,1[$.

No capítulo 4 são apresentados os resultados experimentais obtidos.

Por fim, no capítulo 5 é feita uma análise global do trabalho desenvolvido e são apresentadas as principais conclusões.

CAPÍTULO 2

Modulação de magnitude

A eficiência espectral e o consumo de potência constituem dois parâmetros fundamentais a ter em conta na concepção de qualquer sistema de comunicação digital. Face aos requisitos de elevados débitos colocados pela crescente proliferação e uso de serviços multimédia, as técnicas de modulação do tipo QAM (*Quadrature Amplitude Modulation*) e APSK (*Amplitude and Phase-shift keying*) de ordem elevada e a filtragem FIR do tipo RRC de baixo *roll-off* assumem um papel cada vez mais importante nas comunicações sem fios com vista à maximização da eficiência espectral dos sistemas. No entanto, tal tem como consequência um aumento do PAPR.

Elevados valores de PAPR impõem requisitos adicionais de linearidade nos amplificadores de potência usados nos transmissores, com consequente aumento do consumo de potência do sistema. Um elevado PAPR dificulta ainda o projecto dos conversores digitais - analógicos (DAC), devido à elevada gama dinâmica do sinal, com o inerente aumento dos custos dos sistemas [1, 2, 3].

2.1 Modulador de Magnitude

A figura 2.1 apresenta um diagrama de blocos que ilustra um sistema de transmissão digital genérico utilizando o conceito MM.

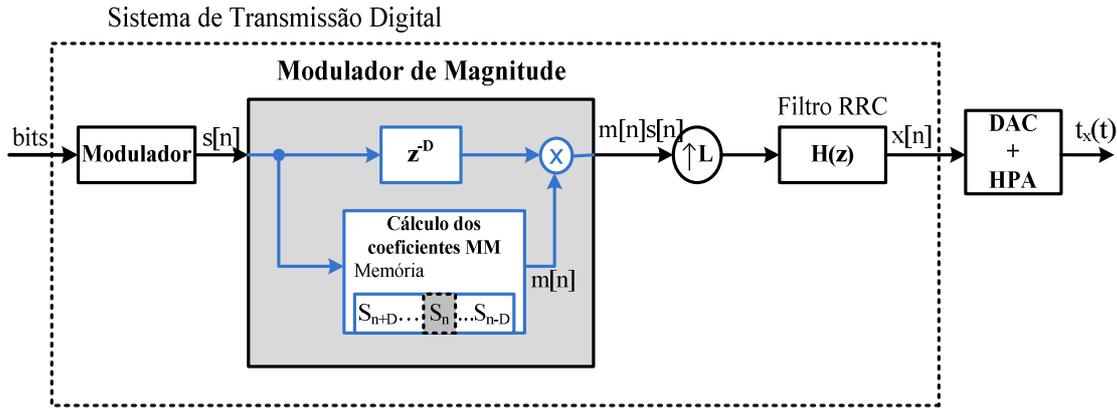


Figura 2.1 - Sistema genérico de transmissão incluindo o MM.

Em sistemas de transmissão de uma portadora, SC, a principal contribuição para o PAPR do sinal a transmitir, $x[n] \in \mathbb{C}$, resulta da filtragem RRC com vista à limitação a largura de banda do sinal modulado $s[n]$, o que dá origem a variações indesejáveis no sinal $x[n]$. O bloco MM ajusta a amplitude de cada símbolo, $s[n]$, antes da filtragem RRC, tendo em conta os símbolos na sua vizinhança (passado e futuro), numa tentativa de evitar a ocorrência de picos de amplitude elevada no sinal $x[n]$.

O equivalente em banda base do sinal digital complexo é dado por,

$$x[n] = \left[\sum_k m[k] s[k] \delta[n - kL] \right] * h[n], \quad (2.1)$$

onde $h[n]$ é a resposta a impulso do filtro RRC e $L \in \mathbb{N}$ é a taxa de sobre-amostragem.

O PAPR do sinal transmitido $x[n]$ é definido por,

$$PAPR = 10 \log_{10} \left[\frac{\max |x[n]|^2}{E(|x[n]|^2)} \right] \text{ (dB)}, \quad (2.2)$$

que é a soma de duas contribuições: $PAPR_{const}$ e $PAPR_{rrc}$, devido à constelação e ao filtro RRC, respectivamente. Estas podem ser obtidas por,

$$PAPR_{const} = 10 \log_{10} \left[\frac{\max_{i=1, \dots, M} |S_i|^2}{\frac{1}{M} \sum_{i=0}^M |S_i|^2} \right] \text{ (dB)}, \quad (2.3)$$

$$PAPR_{rrc} = 10 \log_{10} \left[\frac{\max_{i=0, \dots, L-1} \left(\sum_{n=0}^{2N} |h[nL+i]| \right)^2}{\frac{1}{L} \sum_{n=0}^{2NL} |h[n]|^2} \right] \text{ (dB)}, \quad (2.4)$$

onde S_i com $i=1, \dots, M$ representa os M símbolos da constelação. Assume-se sem perda de generalidade, que o filtro RRC é um filtro com resposta a impulso finita, FIR (*finite impulse response*) do tipo I e com $2NL + 1$ coeficientes, isto é, a sua resposta a impulso $h[n]$ estende-se ao longo de $2N$ símbolos. Nas Tabelas 2.1 e 2.2 estão representadas as contribuições da constelação e do filtro RRC para o PAPR, respectivamente.

Constelação	M-PSK	16-APSK	16-QAM	32-APSK	32-QAM
$PAPR_{const}$	0 dB	1.1 dB	2.6 dB	2.1 dB	2.3 dB

Tabela 2.1 - PAPR devido à contribuição da constelação.

Roll-off	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
PAPR _{RRC} (dB)	6.45	5.93	5.15	4.53	4.07	3.64	3.41	3.40

Tabela 2.2 - PAPR devido à contribuição do filtro RRC.

A maior contribuição para o PAPR tem origem no filtro RRC, especialmente quando se consideram os valores *roll-off* na gama de interesse, $\alpha \leq 0,3$ (ver tabela 2.2).

Seja A a amplitude máxima dos símbolos da constelação e G_{rrc} o ganho em potência do filtro RRC. Daqui em diante assume-se sem perda de generalidade um filtro normalizado, isto é, $G_{rrc} = 1$. Considerando um DAC ideal, assume-se que A é a amplitude máxima admissível na entrada do HPA que não causa distorção. Assim, o MM garante que $|x[n]| \leq A$, a fim de evitar a saturação do HPA [1, 2].

O principal objectivo da técnica de MM é, portanto, cancelar a contribuição indesejável do RRC para o PAPR, eliminando a necessidade de *back-off* do sinal à entrada do HPA. Este é dado por,

$$Back_off = 10 \log_{10} \left[\frac{\max |x[n]|^2}{A_{\max}^2 G_{rrc}} \right] \text{ (dB)}. \quad (2.5)$$

2.2 Modulação de Magnitude Polifásica

Originalmente, a técnica de MM foi proposta tendo por base uma arquitectura baseada em tabelas de consulta (LUT – *Look-Up-Table*) [3, 19]. Os factores de MM eram calculados à priori para cada estado (definido pelo símbolo a modular e seus vizinhos para uma dada profundidade de memória pré-definida) e guardados na posição da tabela correspondente a esse estado.

O método foi inicialmente proposto para modulações de amplitude constante (QPSK - BPSK) [19]. No entanto, a extensão a modulações de ordem superior apresentava problemas devido ao crescimento exponencial do número de estados a considerar [3].

O método de Modulação de Magnitude Polifásica proposto recentemente permitiu ultrapassar as limitações da técnica LUT-MM inicialmente proposta [1, 2]. O seu nome *Multistage Polyphase Magnitude Modulation* (MPMM) resulta do facto de o cálculo em tempo real dos factores MM ter por base a estrutura polifásica do filtro RRC (filtro de limitação da largura de banda do sinal a transmitir).

Um sistema MPMM de k - estágios seguido da representação polifásica do filtro RRC é representado na figura 2.2. O MPMM pode usar vários estágios. Em cada estágio ajusta-se incrementalmente os símbolos, a fim de alcançar um melhor controlo na excursão do sinal a saída do RRC.

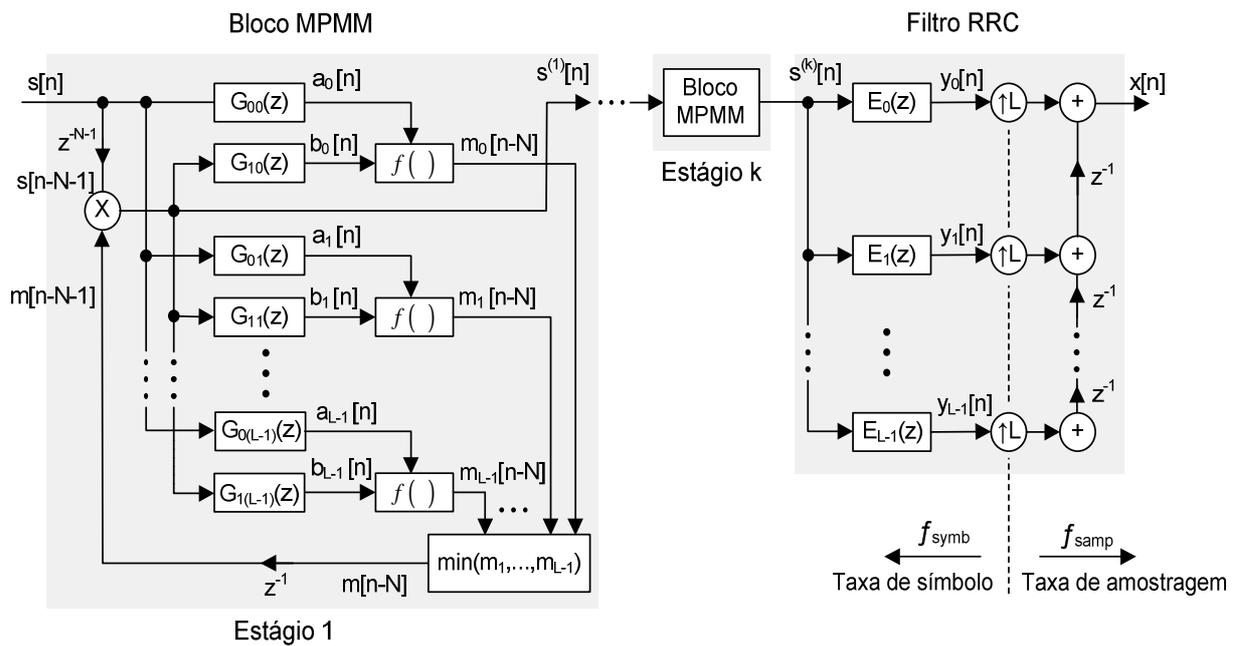


Figura 2.2 - Sistema MPMM seguido do bloco de filtragem do RRC.

Sem perda de generalidade e para uma melhor compreensão, será analisado o sistema MPMM constituído por apenas 1-estágio.

2.2.1 Decomposição polifásica

Como já referido a ideia de calcular os coeficientes do MM em tempo real assenta na decomposição polifásica da função de transferência $H(z)$ do filtro RRC. Com vista a garantir o teorema da amostragem ou de Nyquist e facilitar o projecto do filtro de reconstrução passa – baixo usado na conversão Digital – Analógico, a filtragem RRC é realizada a um ritmo superior ao débito simbólico à saída do modulador.

Seja T_{samp} o período de amostragem para o qual opera o filtro RRC. T_{samp} relaciona-se com o tempo de símbolo, T_{symp} , por: $L = T_{symp} / T_{samp}$, em que $L \geq 2$ e $L \in \mathbb{N}$.

Seja o filtro RRC do tipo FIR real de fase linear com um atraso N , medido em intervalos de símbolos de dados, T_{symp} , cuja função de transferência é dada por,

$$H(z) = \sum_{n=0}^{2NL} h[n] z^{-n}, \text{ com } h[n] = h[2NL - n]. \quad (2.6)$$

A decomposição polifásica de $H(z)$, (figura 2.2) é dada por,

$$E_i(z) = \sum_{n=0}^{2N} e_i[n] z^{-n} = \sum_{n=0}^{2N} h[nL + i + \lambda] z^{-n}, \quad i = 0, \dots, L-1, \quad (2.7)$$

sendo $\lambda \in \mathbb{Z}$ o *offset* de fase usado na decomposição. A partir de $e_i[n]$ calculam-se directamente as respostas a impulso dos filtros $G_{0i}(z)$ e $G_{1i}(z)$ de cada estágio do MPMM, dadas, respectivamente, por,

$$g_{0i}[n] = e_i[n] \times (u[n] - u[n - N - 1]), \quad (2.8)$$

onde $u[n]$ é a sequência degrau unitário e,

$$g_{li}[n] = e_i[n + N + 1] \times (u[n] - u[n - N]). \quad (2.9)$$

A título de exemplo, a figura 2.3 apresenta a decomposição polifásica de um filtro RRC de fase linear onde são representados os coeficientes dos filtros $G_{0i}(z)$ e $G_{1i}(z)$.

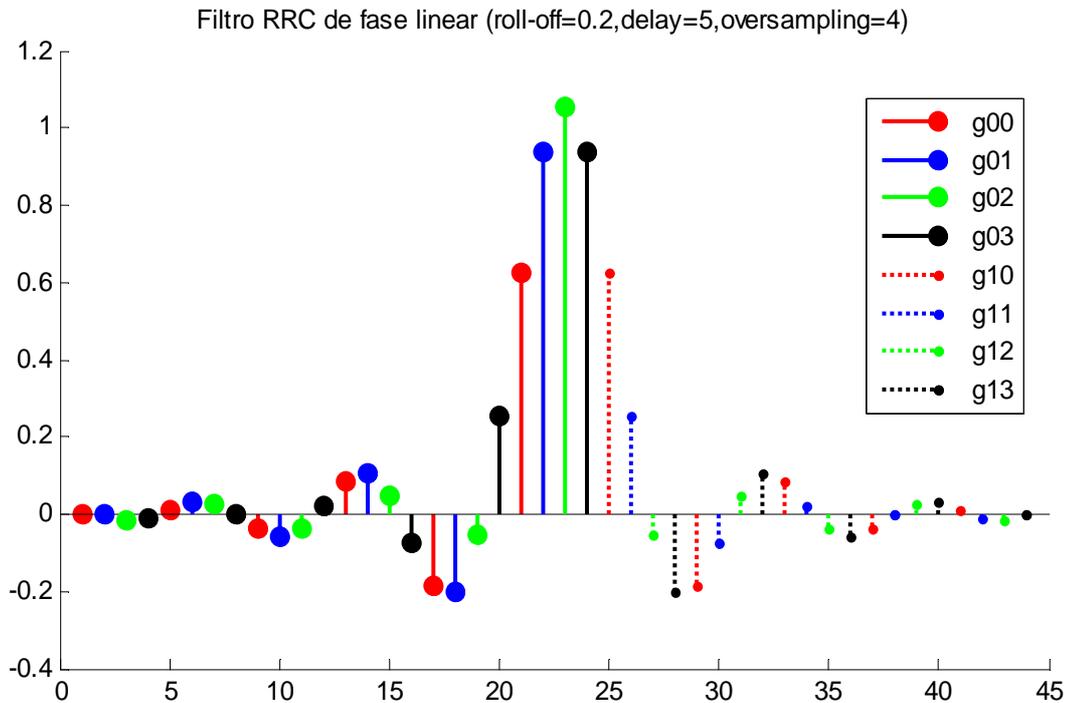


Figura 2.3 - Exemplo de um Filtro RRC.

2.2.2 Controle da envolvente do sinal

O princípio do método MPMM consiste em calcular o melhor coeficiente de MM a aplicar a cada símbolo $s[n_0]$ que garanta $|x[n]| \leq A$, tendo em conta o conjunto de amostras de saída do filtro RRC para o qual $s[n_0]$ mais contribui [1, 2], isto é, no intervalo $[(n_0 + N - 1/2)L; (n_0 + N + 1/2)L]$.

A saída $x[n]$ (figura 2.2) é dada por uma das saídas polifásicas do filtro, no instante $n_0 = \lfloor n/L \rfloor$, atendendo a que,

$$x[n] = y_{(n \bmod L)}[\lfloor n/L \rfloor]. \quad (2.10)$$

Todas as saídas $y_i[n_0]$ com, $i=0, \dots, L-1$, dependem do mesmo conjunto de símbolos de entrada $s[k] \in \mathbb{C}$, com $k = n_0 - 2N, \dots, n_0$. Assim sendo,

$$y_i[n] = \sum_{k=0}^{2N} e_i[k] m_i[n-k] s[n-k], \quad (2.11)$$

onde $m[n] \in]0,1]$ é a sequencia dos coeficientes MM reais que controlam os picos de amplitude indesejáveis no sinal de saída do filtro polifásico $E_i(z)$.

Para uma melhor compreensão do funcionamento do sistema apresenta-se na figura 2.4 um único ramo do sistema MPMM.

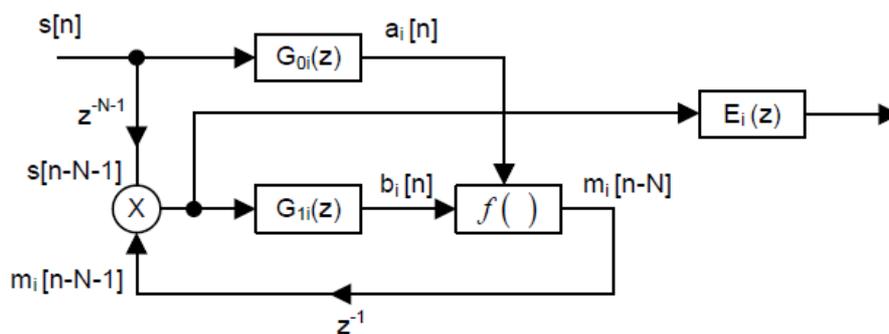


Figura 2.4 - Sistema MPMM com único ramo [2].

O sistema ajusta a amplitude do símbolo $s[n_0]$ de forma a controlar a excursão do sinal a saída de todos os filtros polifásicos $E_i(z)$ no instante $n_0 + N$ de forma a garantir que $|y_i[n_0 + N]| \leq A$.

Uma vez que a energia da resposta a impulso do filtro RRC, $h[n]$ está concentrada em torno da amostra NL (assume-se que $H(z)$ é causal), então, cada resposta a impulso polifásica $e_i[n]$ tem a sua energia centrada ao redor da amostra N e, sendo assim, de acordo com (2.11), $s[n_0]$ é o símbolo que mais contribui para a excursão indesejável à saída do filtro RRC, $x[n]$, durante o intervalo $[(n_0 + N - 1/2)L; (n_0 + N + 1/2)L[$, isto é, para cada valor de saída $y_i[n_0 + N]$ [1, 2], considerando uma decomposição polifásica com $\lambda = -L/2$.

Os filtros $G_{0i}(z)$ e $G_{1i}(z)$ seguidos do bloco $f(A, a, b)$, (função baseada numa equação quadrática, não negativa, cujos parâmetros são obtidos a partir da filtragem do sinal modulado do sistema FIR multi-ritmo), (figura 2.4), tentam antecipar a saída de $E_i(z)$, a cada instante $n_0 + N$, e assim sendo, determinam-se os coeficientes MM, $m_i[n_0]$, a aplicar a $s[n_0]$ de forma a garantir que $|y_i[n_0 + N]| \leq A$.

Um problema que se coloca é o facto de se conhecer os coeficientes $m_i[n_0 - k]$, com $k = 1, \dots, N$, (símbolos passados relativo a $s[n_0]$), quando se calcula $m_i[n_0]$, mas não são conhecidos os valores de MM que irão modular a amplitude dos símbolos $s[n_0 - q]$, com $q = -N, \dots, -1$ (símbolos futuros).

A fim de resolver este problema, assume-se que os símbolos futuros são modulados em amplitude pelo mesmo factor, isto é,

$$m_i[n_0 - q] \approx m_i[n_0], \text{ com } q = -N, \dots, -1. \quad (2.12)$$

A sequência $m_i[n_0]$ controla por multiplicação os símbolos à saída do modulador, $s[n_0]$, de modo a garantir que $|y_i[n_0 + N]| \leq A$. Sendo assim tem-se que:

$$|y_i[n_0 + N]| \leq A \Leftrightarrow \left| \sum_{k=0}^{2N} e_i[k] m_i[n_0 + N - k] s[n_0 + N - k] \right| \leq A \quad (2.13)$$

$$\Leftrightarrow \left| m_i[n_0] \sum_{k=0}^N e_i[k] s[n_0 + N - k] + \sum_{k=N+1}^{2N} e_i[k] m_i[n_0 + N - k] s[n_0 + N - k] \right| \leq A \quad (2.14)$$

$$\Leftrightarrow \left| m_i[n_0] \sum_{k=0}^N g_{0i}[k] s[n_0 + N - k] + \sum_{k=0}^{N-1} g_{1i}[k] m_i[n_0 - k - 1] s[n_0 - k - 1] \right| \leq A \quad (2.15)$$

$$\Leftrightarrow \left| m_i[n_0] a_i[n_0 + N] + b_i[n_0 + N] \right| \leq A \quad (2.16)$$

$$\Rightarrow m_i[n_0] = f(A, a_i[n_0 + N], b_i[n_0 + N]) \quad (2.17)$$

onde $a_i[n_0]$ e $b_i[n_0]$ são as saídas dos filtros $G_{0i}(z)$ e $G_{1i}(z)$, respectivamente (ver figura 2.4). A função $f(A, a, b)$ é baseada numa equação quadrática, não negativa, cujos parâmetros são obtidos a partir da filtragem do sinal modulado do sistema FIR multi-ritmo que é usada com objectivo de controlar a excursão do sinal à saída do RRC. Esta é definida por,

$$f(A, a, b) = \begin{cases} 1 & , |a + b| \leq A \\ \frac{-\operatorname{Re}\{ab^*\} + \sqrt{\operatorname{Re}\{ab^*\}^2 - |a|^2(|b|^2 - A^2)}}{|a|^2} & , |a + b| > A \end{cases} \quad (2.18)$$

Para garantir que $|y_i[n_0 + N]| \leq A$ para $i = 0, \dots, L - 1$, a sequência MM $m[n]$ deve ser escolhida de forma que

$$m[n] = \min(m_0[n], m_1[n], \dots, m_{L-1}[n]) . \quad (2.19)$$

Assumindo que a condição (2.12) é verdadeira apenas quando se calculada $m[n_0]$, é preciso efectuar uma melhoria de $m[n]$. Quando o coeficiente MM, $m[n_0 + 1]$ é maior que o coeficiente MM, $m[n_0]$, a condição $|y_i[n_0 + N]| \leq A$ deixa de ser válida. Contudo, dado que o símbolo $s[n_0 + 1]$ é mais relevante para a saída $y_i[n_0 + N + 1]$, deve-se controlar $m[n_0 + 1]$ de

forma a não atenuar muito $y_i[n_0 + N + 1]$ (o que seria indesejável) e garantir assim, que a restrição $|y_i[n_0 + N]| \leq A$ só é ligeiramente ultrapassada. Esse objectivo é facilmente obtido, se for adoptado [1, 2] o seguinte procedimento:

$$m[n+1] > m[n] \Rightarrow m'[n+1] = \left(\frac{m[n+1] + m[n]}{2} \right). \quad (2.20)$$

CAPÍTULO 3

Modulador de Magnitude em DSP

Neste capítulo pretende-se descrever os sistemas MPMM desenvolvidos no DSP TMS320C6713. Este é um processador de 32 bits com aritmética de vírgula flutuante (pode também operar com aritmética de vírgula fixa), capaz de executar 8 instruções por ciclo de relógio. É baseado na arquitectura *Very Long Instruction Word* (VLIW) [4, 9, 14]. Possui 32 registos de 32 bits e oito unidades funcionais independentes – dois multiplicadores e seis aritméticas lógicas, ALU (*Arithmetic Logic Unit*) com extensões VelociTI.2 (a segunda geração da VelociTI, desenvolvida pela TI). A arquitectura VelociTI.2 permite melhorar o desempenho do processador e também aumentar o paralelismo. Dispõe de 264 Kbyte de memória interna e de 4 Gbyte de espaço de endereçamento externo. Uma descrição mais detalhada sobre o processador C6713 encontra-se disponível no Anexo A.

Foram exploradas as capacidades de processamento paralelo e de endereçamento deste DSP com vista a maximizar o desempenho do sistema. De seguida são apresentadas as técnicas de programação utilizadas na realização deste trabalho que permitiram tirar o máximo proveito dessas capacidades.

3.1 Maximização do desempenho do sistema

Através do processo de optimização dos algoritmos pretende-se atingir dois objectivos importantes: execução mais rápida e uma redução do código. Para atingir estes objectivos é necessário conhecer a arquitectura do processador e o seu conjunto de instruções.

O DSP utilizado permite utilizar diversas metodologias de optimização dos algoritmos, entre as quais:

- Uso de instruções em paralelo, o que significa que várias unidades irão operar no mesmo ciclo;
- Preenchimento das unidades de atraso (NOP's – *delay slots*) com código útil;
- Realização de leituras de memória em palavras de 32 bits (*words*) e 64 bits (*double word*);
- *loop unrolling*, ou seja redução ao máximo de ciclos repetindo blocos de instruções;
- Programação *pipeline*.

Os três primeiros métodos referidos possibilitam um aumento do desempenho e uma diminuição do tamanho de código. Pelo contrário, o método *loop unrolling* permite evitar a quebra do mecanismo de *pipeline* associado à execução de condições de teste à custa de um aumento do tamanho do código [6].

O método *software pipelining* tem como principal objectivo optimizar o código associado aos ciclos (*loops*). Esta tarefa é complexa, pelo que deve ser utilizado o procedimento descrito de seguida [5, 6]:

1. O código é escrito em *assembly* linear, permitindo uma visão clara do algoritmo uma vez que não é necessário ter cuidado com as unidades funcionais, instruções NOP, etc;

2. Estudar o fluxo óptimo de dados;
3. Listar os recursos requeridos para determinar o tamanho mínimo necessário para cada bloco iterativo;
4. Criar uma tabela que mostra as instruções executadas nas unidades apropriadas numa base de ciclo-por-ciclo de processador de forma a identificar possíveis incompatibilidades;
5. Gerar o código em linguagem máquina.

3.2 Demonstrador MPMM em tempo real

Na figura 3.1 é ilustrado o diagrama funcional do sistema desenvolvido. O bloco gerador de símbolos simula o tipo de modulação utilizado, nomeadamente, BPSK e QPSK. O bloco modulador de magnitude calcula os coeficientes MM, $m_i[n]$, através dos filtros $G_{oi}(z)$ e $G_{ii}(z)$ seguidos do bloco $f(A, a, b)$, (figura 2.4) de forma a garantir que a saída $|y_i[n]| \leq A$, isto é, com objectivo de cancelar a contribuição indesejável do RRC para o PAPR. Após o ajuste da amplitude de cada símbolo este é fornecido ao filtro RRC. De seguida é realizada a filtragem RRC sem a aplicação do método MM. Isso deve-se ao facto de querermos visualizar o efeito do método MM sobre o sinal modulado. Os resultados da filtragem são enviados para o DAC do processador C6713. Ao conector *line-out* do DSK é ligado um osciloscópio digital com a finalidade de ver o sistema em funcionamento em tempo real.

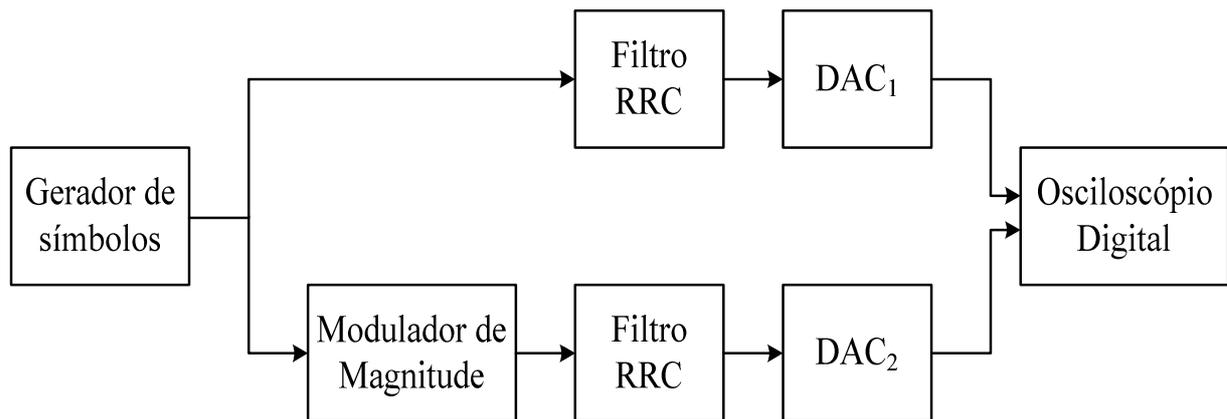


Figura 3.1 - Diagrama de blocos do sistema desenvolvido.

Foram desenvolvidos os seguintes sistemas:

- **MPMM_BPSK** – aplicação do conceito de modulador de magnitude utilizando modulação BPSK;
- **MPMM_QPSK** – aplicação do conceito de modulador de magnitude utilizando modulação QPSK.

Durante o desenvolvimento destes sistemas foram utilizadas as diversas técnicas de otimização já mencionadas. Fez-se uso de instruções em paralelo, preenchimento dos *delays slots* (NOP's) com código útil, realização de leituras de memória em palavras de 64 bits e utilização da técnica *branch delay* (técnica que consiste no preenchimento dos cinco ciclos de relógio que a instrução *branch* consome, com código útil). Desta forma foi possível executar várias instruções num mesmo ciclo. Através do uso do mecanismo *software pipelining*, conseguiu-se otimizar os blocos de código associados aos *loops*, utilizando o máximo de unidades funcionais do processador em paralelo.

Pode-se decompor o desenvolvimento dos sistemas já mencionados nas seguintes etapas:

1. Desenvolvimento de um gerador de símbolos aleatórios para simular a técnica de modulação utilizada.

2. Realização dos filtros polifásicos: $G_{0i}(z)$ e $G_{1i}(z)$.
3. Cálculo dos coeficientes do MM, $m[n]$.
4. Filtragem polifásica com o filtro RRC a ser aplicado ao sinal MM e filtragem polifásica do sinal proveniente do gerador de símbolos, isto é sem MM.

De seguida é apresentada uma explicação mais detalhada de cada uma destas etapas.

3.2.1 Gerador de Bits aleatórios

O método de geração de bits aleatórios baseia-se na utilização de registos de deslocamento realimentados por " polinómios binários primitivos " e de aritmética módulo 2 [15].

Foi escolhido o seguinte polinómio gerador [15]:

$$f(x) = x^{32} + x^7 + x^5 + x^3 + x^2 + x^1 + x^0. \quad (3.1)$$

Cada polinómio primitivo de módulo 2 e de ordem n (32) define uma relação de recorrência para a obtenção do novo bit aleatório. Esta produz uma sequência de comprimento máximo, ou seja, percorre-se todas as sequências possíveis de n bits (excepto todos os zeros) antes que ela se repita. Obtêm-se $2^n - 1$ bits aleatórios antes que a sequência volte a repetir-se. A figura 3.2 ilustra a gerador implementado [15]. Este consiste apenas num único registo de deslocamento de n bits e um XOR ("ou exclusivos " ou adição módulo 2). Para o polinómio primitivo dado acima, a fórmula de recorrência é

$$a_0 = a_{32} \wedge a_7 \wedge a_5 \wedge a_3 \wedge a_2 \wedge a_1. \quad (3.2)$$

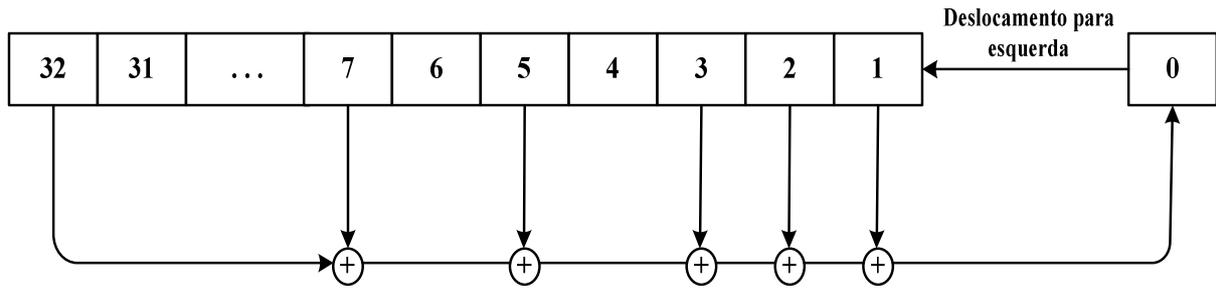


Figura 3.2 - Diagrama de um gerador de 32 bit.

3.2.2 Estrutura paralela dos filtros polifásicos

O filtro RRC utilizado é um filtro FIR de fase linear, do tipo I com $2NL + 1$ coeficientes, em que, $N = 18, L = 4$ e um *roll-off* de 0.2 (faixa de interesse).

Aplicando a decomposição polifásica a esse filtro obtêm-se os filtros $G_{0i}(z)$ e $G_{1i}(z)$. Os $G_{0i}(z)$ são constituídos pelos coeficientes do lado esquerdo do filtro RRC e os $G_{1i}(z)$ são os coeficientes do lado direito do filtro RRC. A saída de cada um desses filtros produz $a_i[n]$ e $b_i[n]$ (figura 2.4) em que

$$a_i[n_0] = \sum_{k=0}^N g_{0i}[k] s[n_0 - k], \quad i = 0, \dots, L-1, \quad (3.3)$$

e

$$b_i[n_0] = \sum_{k=0}^{N-1} g_{1i}[k] m_i[n_0 - k - N - 1] s[n_0 - k - N - 1]. \quad (3.4)$$

Um simples análise de (3.3) e (3.4) permite concluir que os dados são os mesmos para $i = 0, \dots, L-1$ e o que varia são os coeficientes dos filtros.

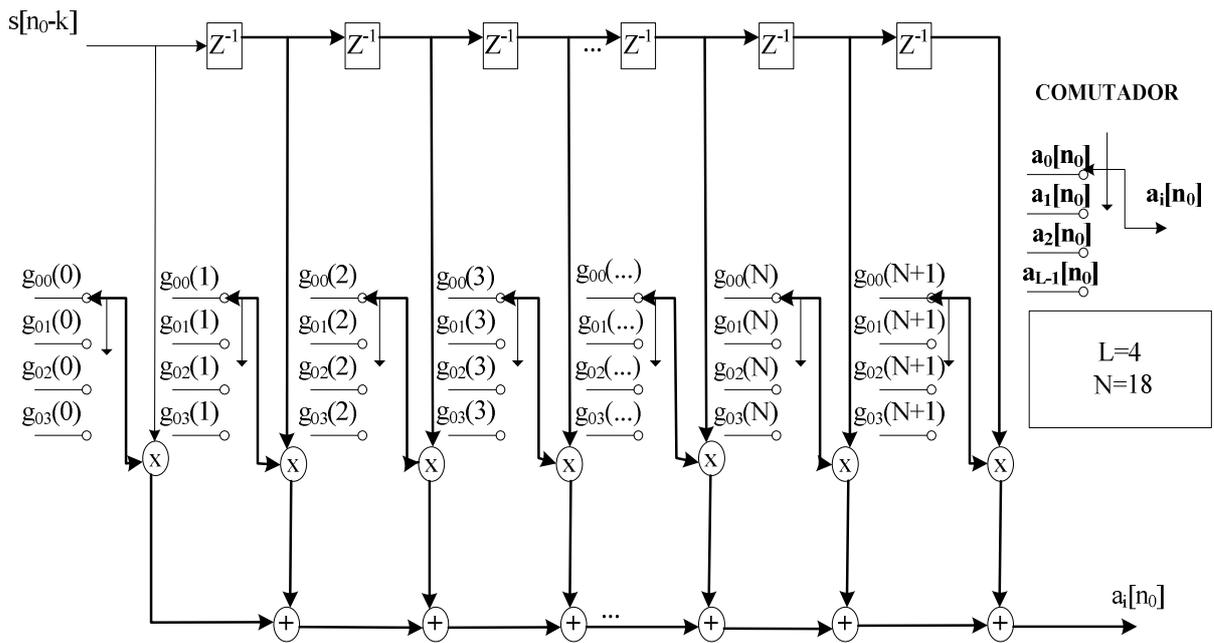


Figura 3.3 - Cálculo das seqüências $a_i[n]$.

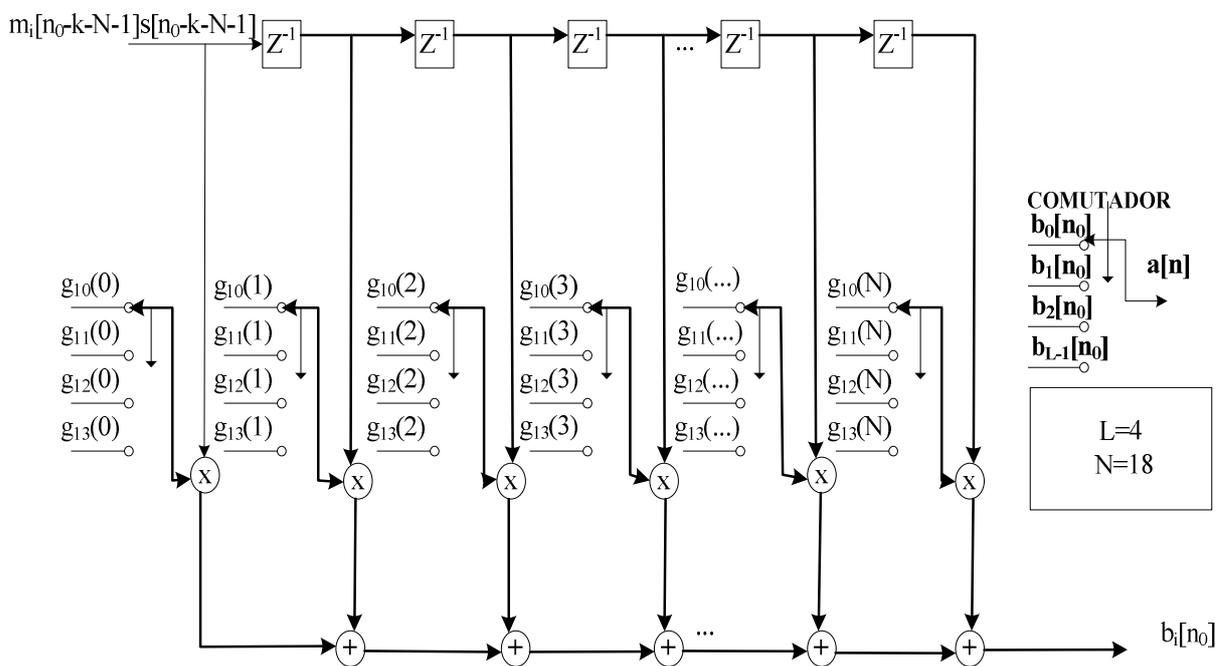


Figura 3.4 - Cálculo das seqüências $b_i[n]$.

As figuras 3.3 e 3.4 ilustram como foram desenvolvidas as equações (3.3) e (3.4) [18]. Esta estrutura polifásica do filtro RRC permite reduzir o esforço computacional do sistema, uma

vez que permite a realização de quatro convoluções em paralelo, aumentando desta forma a eficiência aritmética do algoritmo (leitura de dados sem repetição e coeficientes do filtro RRC empacotados em palavras de 64 bits).

Os coeficientes dos filtros $G_{0i}(z)$ e $G_{1i}(z)$ são guardados num bloco de memória com endereçamento linear enquanto que os símbolos de entrada são colocadas num vector circular, como mostra a figura 3.5, para o caso dos filtros $G_{0i}(z)$.

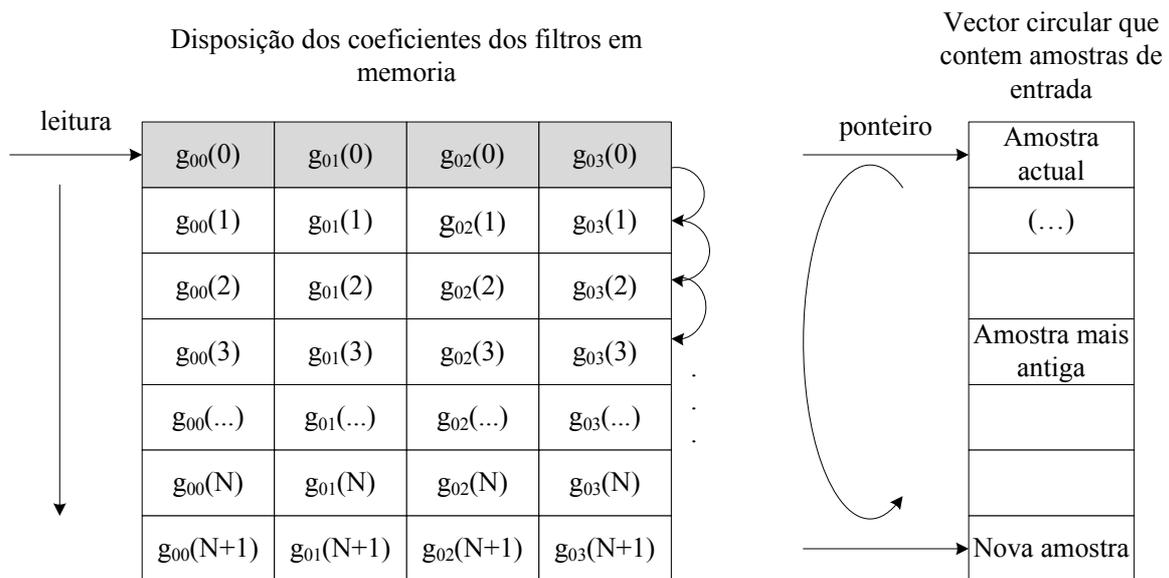


Figura 3.5 - Disposição dos dados na memória.

As leituras dos coeficientes dos filtros são efectuadas em palavras de 64 bits, isto é, em apenas um acesso à memória consegue-se ler dois coeficientes em simultâneo.

Com esta organização dos coeficientes do filtro e das amostras de entrada o desenvolvimento das equações (3.3) e (3.4) resume-se à execução em paralelo de cada um dos produtos e da sua adição no acumulador. O fluxograma apresentado na figura 3.6 indica a sequência de operações efectuadas para a realização da filtragem. O símbolo || indica que as instruções são realizadas em paralelo.

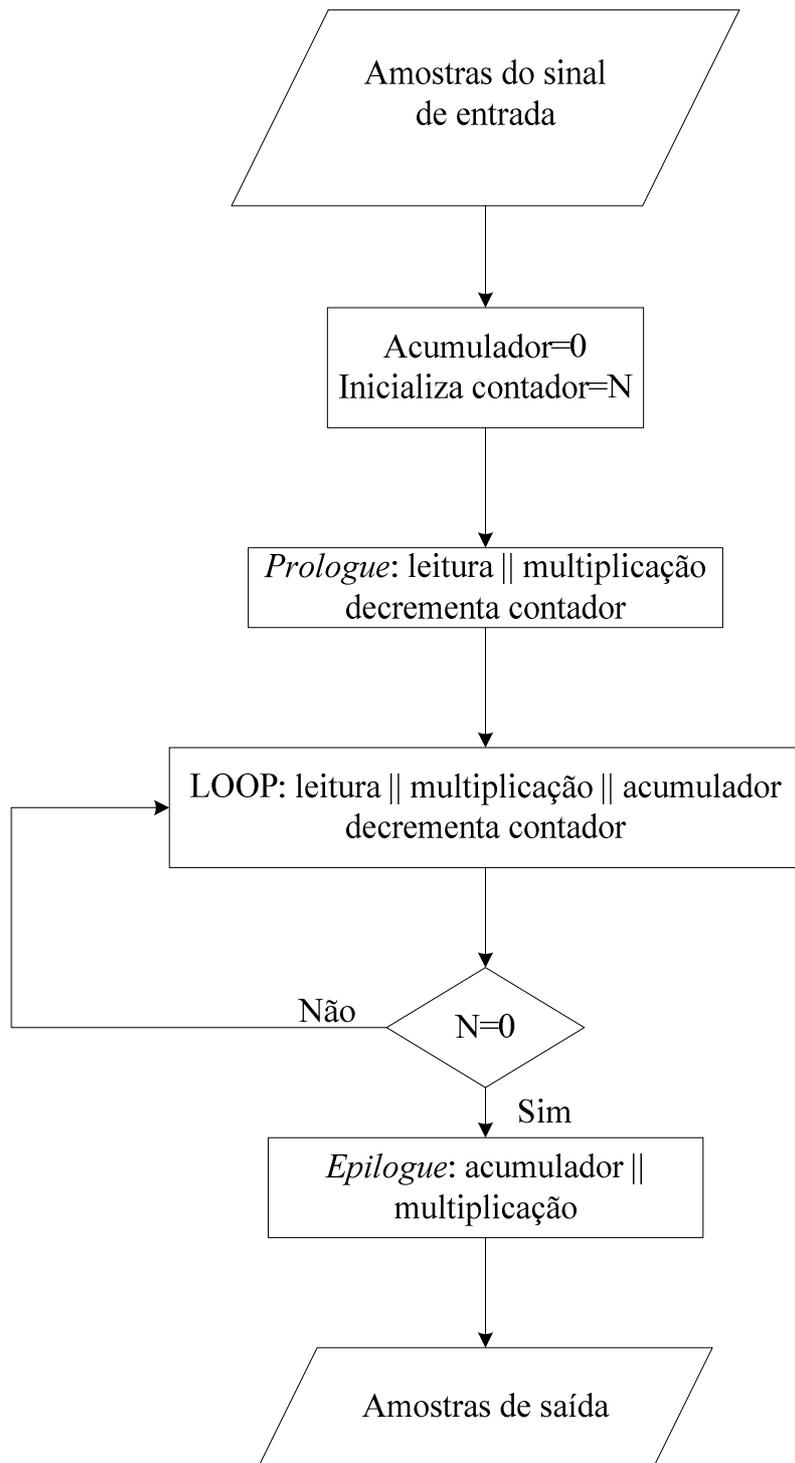


Figura 3.6 - Fluxograma da filtragem polifásica para as sequências $a_i[n]$ e $b_i[n]$.

A figura 3.7 mostra o diagrama da dependência gráfica da filtragem efectuada, onde são apresentadas todas as instruções e as unidades funcionais utilizadas.

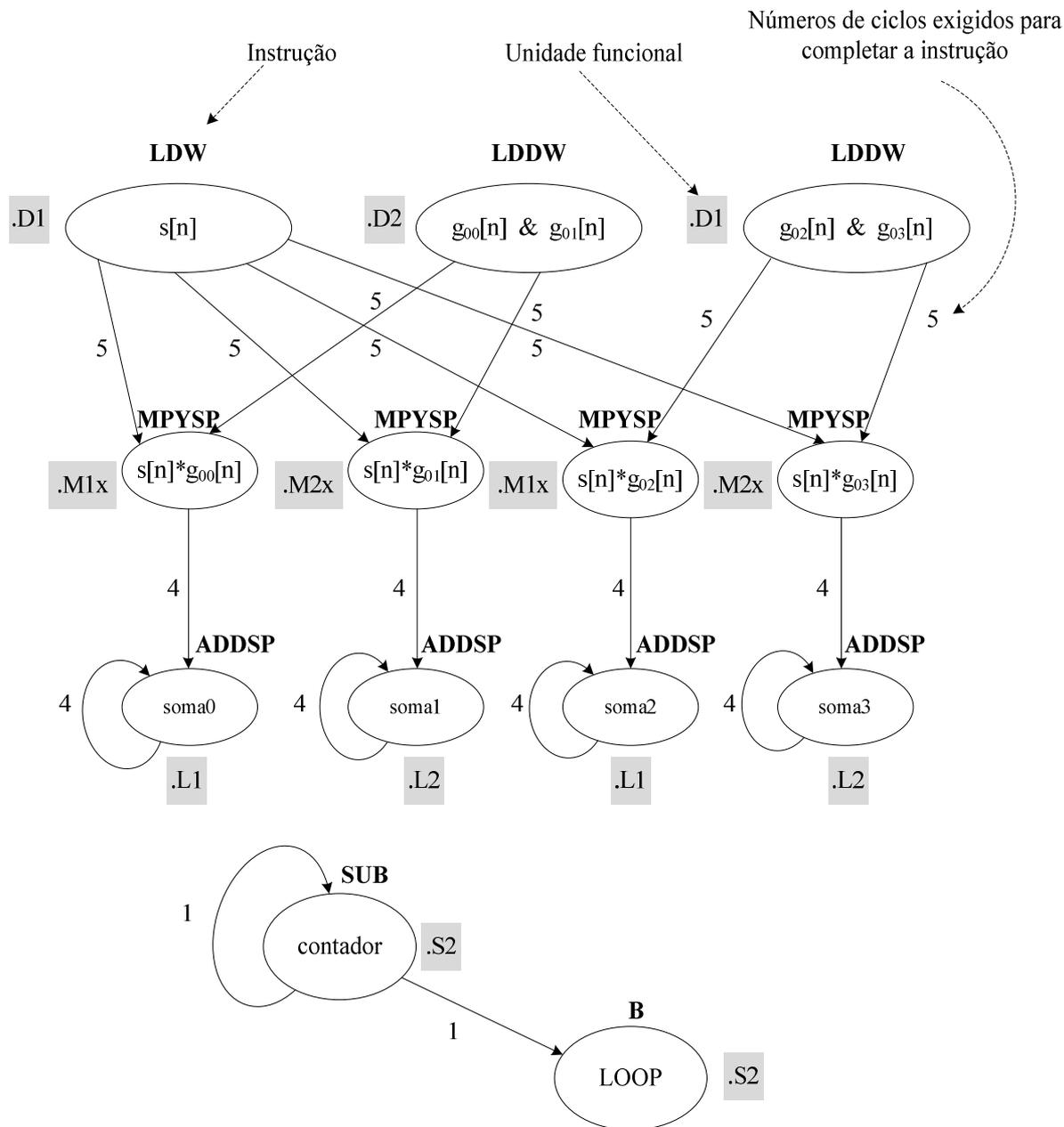


Figura 3.7 – Diagrama funcional de apoio à Figura 3.6.

A instrução LDW faz a leitura das amostras de entrada guardadas na memória em palavras de 32 bits. Um acesso à memória tem 5 ciclos de atraso, portanto, o valor estará disponível após 5 ciclos. A instrução LDDW faz leituras de palavras de 64 bits. Neste caso, em cada leitura feita são carregadas dois coeficientes do filtro em simultâneo. A instrução multiplicação, MPYSP, é executada 5 ciclos depois das instruções LDW e LDDW, isto é, quando os valores estão disponíveis. Esta instrução tem 4 ciclos de atraso. A instrução adição, ADDSP, é

executada 4 ciclos depois da instrução MPYSP e possui 4 ciclos de atraso. Por fim a instrução *branch*, B, é a instrução que possui mais ciclos de atraso, isto é, possui 6.

3.2.3 Cálculo dos coeficientes MM

Uma vez que já estão calculados os valores $a_i[n]$ e $b_i[n]$, a determinação dos coeficientes MM, $m[n]$, resume-se à resolução da equação (2.18). Esta consiste na multiplicação de cada símbolo à saída do modulador por um factor real pertencente ao intervalo]0,1], resultante da solução de uma equação quadrática cujos parâmetros são obtidos a partir da filtragem do sinal modulado pelo sistema FIR multi-ritmo. Nesta operação faz-se uso de algoritmos de divisão e de raiz quadrada de números reais.

Caso BPSK

Para a modulação BPSK a equação (2.18) é simplificada uma vez que a amplitude das amostras do sinal de entrada só contém valores reais (1 e -1). A função reduz-se a

$$f(A, a, b) = \begin{cases} 1 & , |a + b| \leq A \\ \frac{A - |b|}{|a|} & , |a + b| > A \end{cases} \quad (3.5)$$

O fluxograma apresentado na figura 3.8 indica a sequência de operações efectuadas para a realização da equação (3.5).

A divisão num processador C67x pode ser efectuada usando o recíproco do denominador (a função RCPS [10]) e de seguida multiplicando o resultado pelo numerador [8].

É possível aumentar a precisão da estimativa inicial de x , $v[0] = RCPS(x)$, utilizando o algoritmo iterativo de **Newton-Raphson** dado por,

$$v[n+1] = v[n] \times (2.0 - x \times v[n]). \quad (3.6)$$

A precisão aumenta em cada iteração, isto é, $v[n+1] \rightarrow \frac{1}{x}$. Para obter uma boa precisão fazemos duas iterações do algoritmo de **Newton-Raphson**. A divisão é realizada de acordo com o seguinte algoritmo [13]:

Algoritmo 1: Algoritmo de divisão de números reais

- Calcular o recíproco do 2º argumento: $v[0] = RCPSP(\arg 2)$;
 - Efectuar 1ª iteração: $v[1] = v[0] \times (2.0 - \arg 2 \times v[0])$;
 - Efectuar a 2ª iteração: $v[2] = v[1] \times (2.0 - \arg 2 \times v[1])$;
 - Multiplicar o 1º argumento pelo resultado da 2ª iteração: $resultado = \arg 1 \times v[2]$.
-

Na realização deste algoritmo usou-se como técnica de optimização o preenchimento dos *delays slots* (NOP's) com código útil. Assim conseguiu-se realizar quatro divisões em simultâneo.

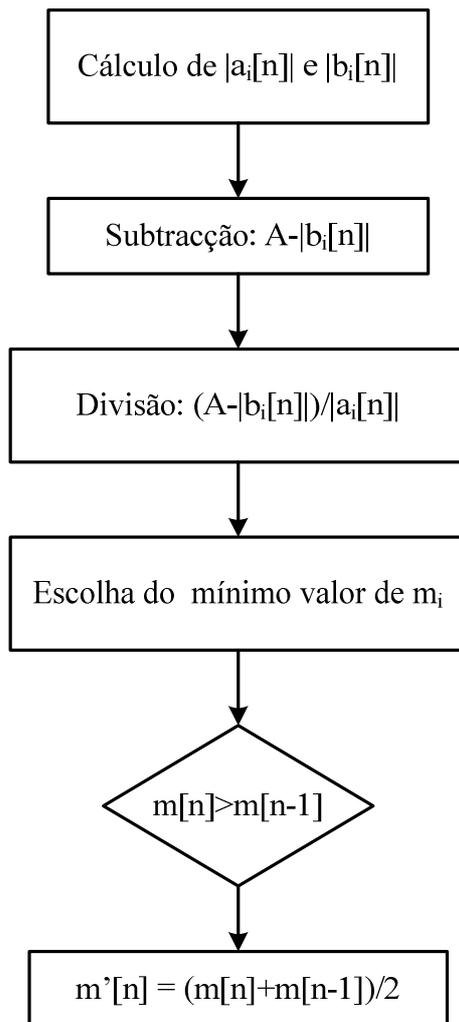


Figura 3.8 - Fluxograma do cálculo da função $f(A, a, b)$ para a modulação BPSK.

Caso QPSK

Para a modulação QPSK resolve-se a equação quadrática (2.18). Para além do algoritmo de divisão, foi desenvolvido o algoritmo da raiz quadrada de números reais. Neste caso as amostras do sinal de entrada são complexa, daí o processamento ser mais exigente. É utilizado um gerador que gera quatro símbolos diferentes $(1, -1, j, -j)$. Esses símbolos são colocados num vector circular. A figura 3.9 ilustra como os símbolos de entrada estão guardados na memória.

Vector circular que contem amostras de entrada (valores complexos)

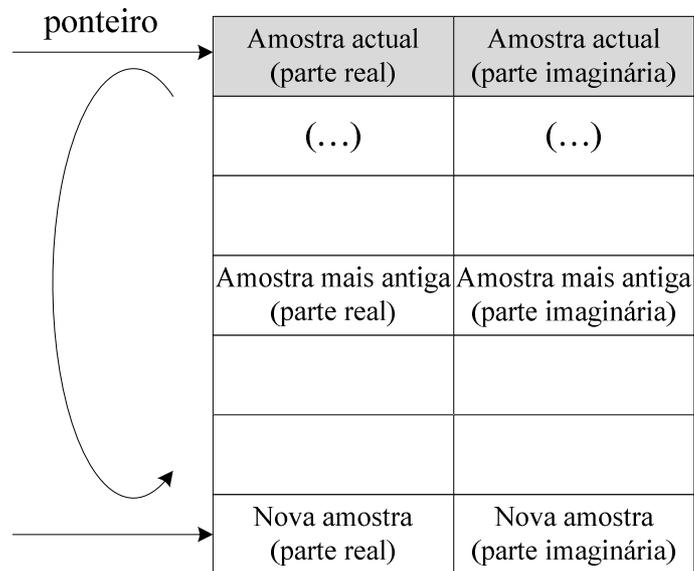


Figura 3.9 – Disposição dos símbolos de entrada para o caso QPSK.

Neste caso as leituras das amostras de entrada são efectuadas em palavras de 64 bits, isto é, em apenas um acesso à memória consegue-se ler duas amostras em simultâneo, diminuindo assim o número de leituras e, conseqüentemente, o tamanho do código.

O cálculo da raiz quadrada num processador C67x pode ser efectuado usando o recíproco da raiz quadrada do argumento, a função RSQRSP [10].

É possível aumentar a precisão da estimativa inicial do recíproco da raiz quadrada de x , $v[0] = RSQRSP(x)$, utilizando o algoritmo iterativo de **Newton-Raphson** dado por,

$$v[n+1] = v[n] \times (1.5 - (x/2) \times v[n] \times v[n]). \quad (3.7)$$

A precisão aumenta por cada iteração. Para obter uma boa precisão fazemos duas iterações do algoritmo **Newton-Raphson**. A raiz quadrada é realizada de acordo com o seguinte algoritmo [13]:

Algoritmo 2: Algoritmo da raiz quadrada de números reais

- Calcular o recíproco da raiz quadrada do argumento: $v[0] = RSQRSP(\text{arg})$;
 - Multiplicar o argumento por $1/2$: $V = 0.5 \times \text{arg}$;
 - Efectuar 1ª iteração: $v[1] = v[0] \times (1.5 - V \times v[0]^2)$;
 - Efectuar a 2ª iteração: $v[2] = v[1] \times (1.5 - V \times v[1]^2)$;
 - Multiplicar o 1º argumento pelo resultado da 2ª iteração: $\text{resultado} = \text{arg} \times v[2]$.
-

Também na realização deste algoritmo usou-se como técnica de optimização o preenchimento dos *delays slots*, realizando assim, quatro raízes quadradas em simultâneo. O fluxograma apresentado na figura 3.10 indica a sequência de operações efectuadas para o cálculo dos coeficientes MM para o caso da modulação QPSK.

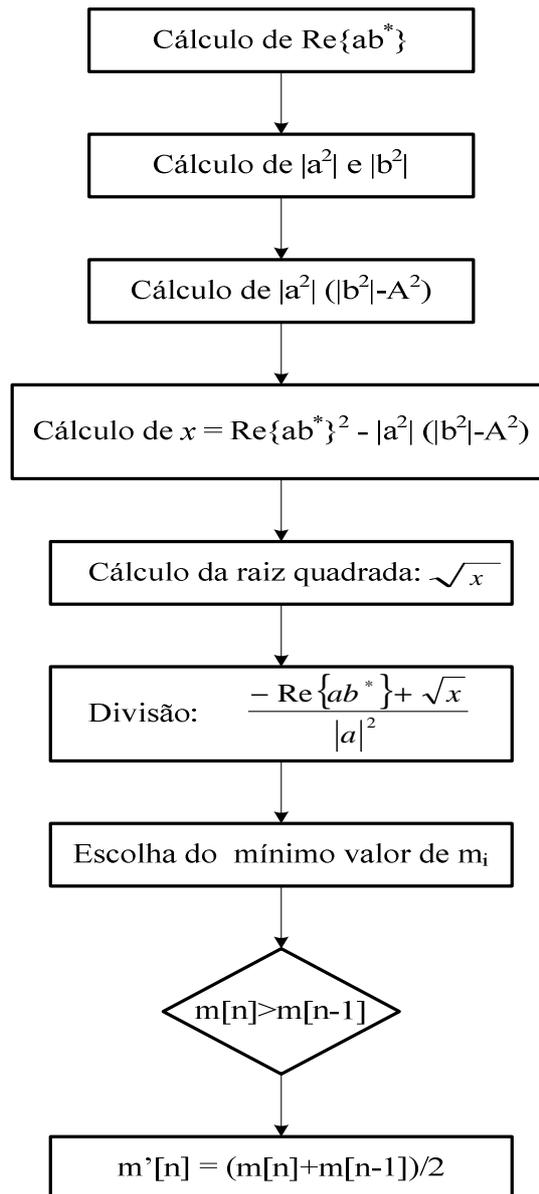


Figura 3.10 - Fluxograma do cálculo da função $f(A, a, b)$ para a modulação QPSK.

3.2.4 Filtragem polifásica dos filtros RRC

A última etapa do demonstrador do MPMM em tempo real consiste nas filtrações polifásicas dos filtros RRC. Como mostra a figura 3.1 foram realizadas duas filtrações, uma aplicada aos coeficientes MM e outra sem a aplicação do método MM.

Com vista a otimizar o código e a minimizar o número de operações aritméticas a realizar, evitando a duplicação de cálculos já efectuados anteriormente, foi feito uso dos valores resultantes da filtragem $G_{0i}(z)$ e $G_{1i}(z)$ (figura 2.4). Sabemos que a componente polifásica $E_i(z)$ é dada por,

$$E_i(z) = G_{0i}(z) + z^{-N-1}G_{1i}(z), \quad (3.8)$$

e como tal o resultado da filtragem pode ser obtido tendo por base o esquema representado na figura 3.11.

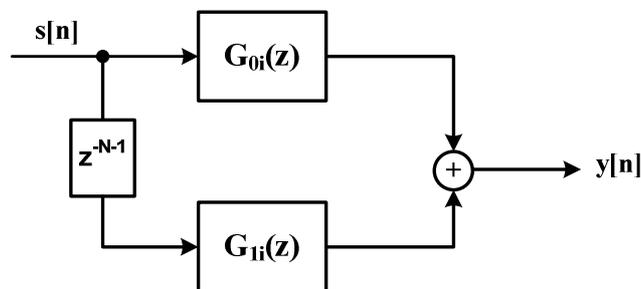


Figura 3.11 – Decomposição da componente polifásica $E_i(z)$.

Por observação da figura 2.4 pode-se concluir que com vista à obtenção do sinal de banda limitada sem MM, $y[n]_{(sem\ MM)}$, e do mesmo sinal aquando do uso de MM, $y[n]_{(com\ MM)}$, existe apenas a necessidade de cálculo do resultado da filtragem de $s[n]$ por $G_{1i}(z)$ e do sinal MM $m[n]s[n]$ por $G_{0i}(z)$. De acordo com a figura 3.11 o resultado da filtragem RRC polifásica pode ser obtida tendo por base o diagrama de blocos da figura 3.12.

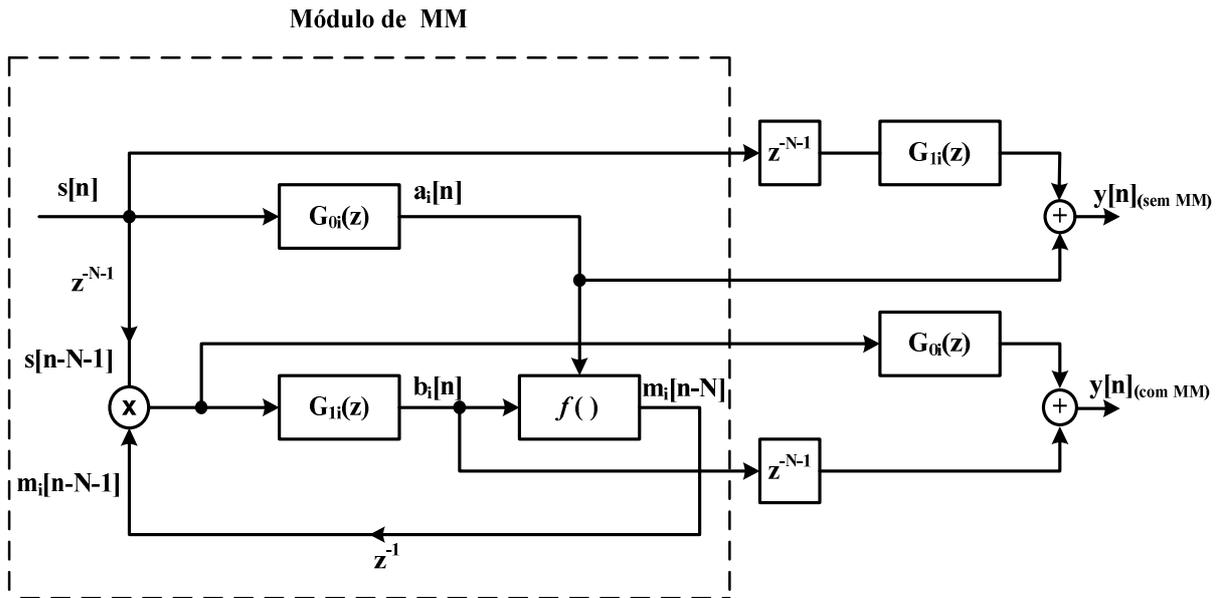


Figura 3.12 – Sistema desenvolvido para a filtragem polifásica do filtro RRC.

A sequência de operações efectuadas para a realização dessas duas filtragens adopta a mesma sequência descrita no fluxograma da figura 3.6. Os resultados finais das duas filtragens, $y[n]_{(sem\ MM)}$ e $y[n]_{(com\ MM)}$, são guardados em dois vectores circulares cujas amostras são enviadas para o DAC.

A entrada e saída de sinais analógicos é realizada por meio do CODEC AIC23 *stereo* de 18 bits [4]. A comunicação entre o processador de sinal e este periférico é realizada através do porto série McBSP1 (*Multichannel Buffered Serial Port*) por interrupções (o processador é interrompido sempre que o DAC está disponível para receber uma nova amostra).

Foi desenvolvida uma rotina de interrupção (em linguagem C) que consiste em fazer as leituras dos resultados das filtragens que estão nos vectores circulares e depois enviá-los para os canais do CODEC (canal direito e canal esquerdo). Finalmente, é ligado ao conector *line-out* do DSK um osciloscópio digital onde são visualizadas os sistemas desenvolvidos em tempo real.

CAPÍTULO 4

Resultados experimentais

Neste capítulo é analisado o desempenho dos sistemas MPMM_BPSK e MPMM_QPSK e de seguida são apresentados os resultados experimentais obtidos da realização dos mesmos.

4.1 Tempo de processamento

Para avaliar o desempenho do sistema é necessário realizar uma medição do tempo de processamento expresso em função do número de ciclos de relógio do processador gastos. O número de ciclos é medido usando um *timer* interno do processador [5].

A expressão seguinte permite calcular o tempo de processamento em função do conteúdo do *timer* e do relógio da CPU.

$$tempo = \frac{timer\ count \times 4}{clock\ source(Hz)} = \frac{número\ de\ ciclos}{225MHz} \quad [s].$$

A tabela 4.1 apresenta os números de ciclos bem como o tempo de processamento obtidos no desenvolvimento dos sistemas.

Sistema	Número de ciclos	Tempo de processamento	Frequência máxima de refreshamento do DAC
MPMM_BPSK	926	$4.1155 \times 10^{-6} \text{s}$	243kHz
MPMM_QPSK	1300	$5.7777 \times 10^{-6} \text{s}$	173kHz

Tabela 4.1 - Tempo de processamento.

Podemos constatar que o sistema MPMM_QPSK consome mais ciclos do que o sistema MPMM_BPSQ. Isto deve-se à resolução da equação quadrática (divisão e raiz quadrada de números reais) e ao processamento completo do ramo associado à componente complexa do sinal modulado. Outra particularidade importante, é que a frequência máxima de refreshamento do DAC é superior à frequência máxima de operação do DAC (96 kHz) o que ilustra o sucesso obtido na optimização dos algoritmos desenvolvidos.

4.2 Visualização da saída do sistema MPMM

De seguida são apresentados alguns exemplos experimentais. As figuras 4.1 e 4.2 ilustram os sistemas MPMM_BPSK e MPMM_QPSK, respectivamente. Estas foram obtidas usando um osciloscópio digital, durante o funcionamento do sistema em tempo real e para a frequência máxima de operação do DAC.

Analisando as figuras apresentadas é evidente a contribuição do filtro RRC para o aumento do PAPR, isto é, são visíveis os picos indesejáveis no sinal de saída. Pode-se verificar que o sistema controla a excursão do sinal transmitido, evitando os picos indesejáveis. Esse controlo ocorre de uma forma suave e eficiente conforme documentam as referidas figuras.

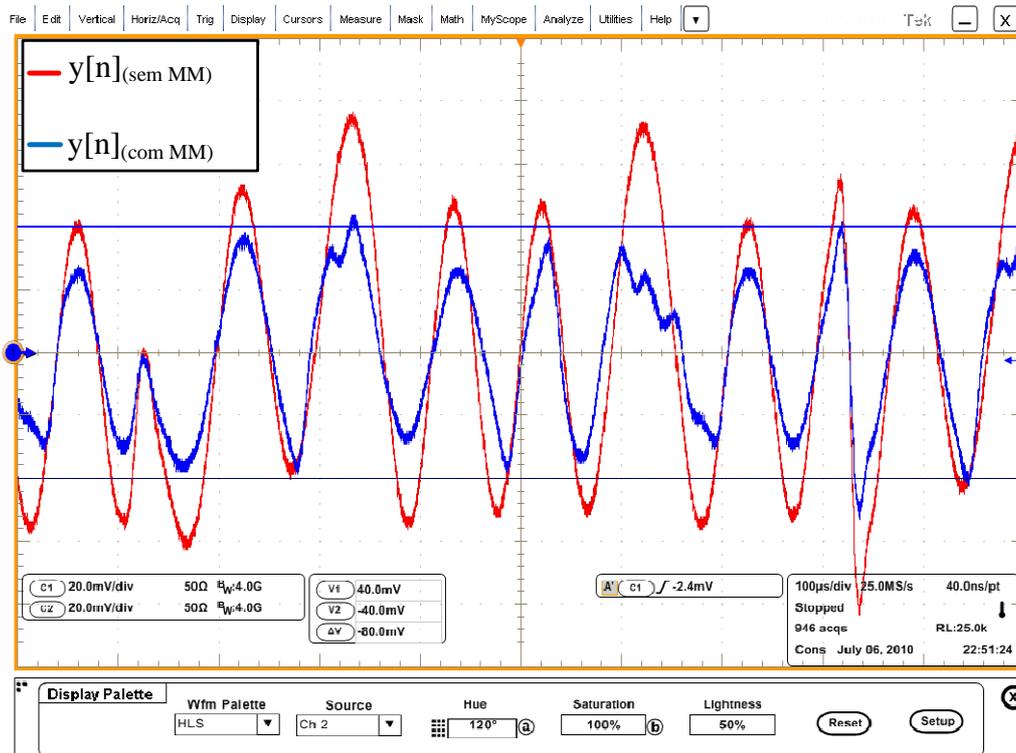
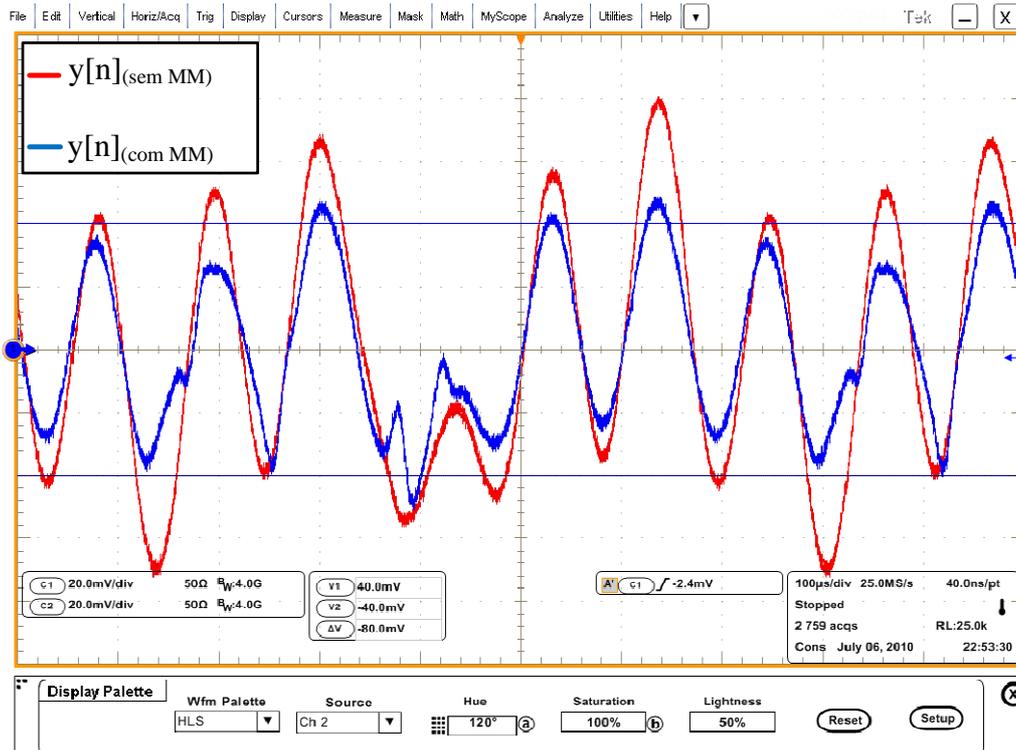


Figura 4.1 – Segmentos do sinal à saída do filtro RRC para o sistema MPMM_BPSK.

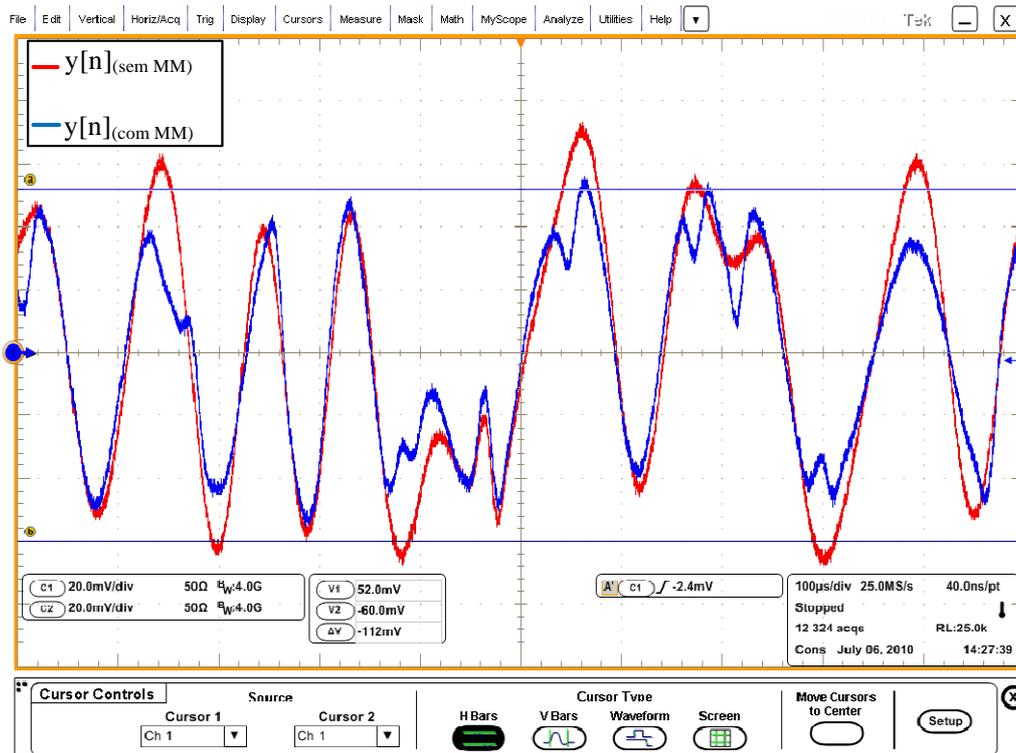
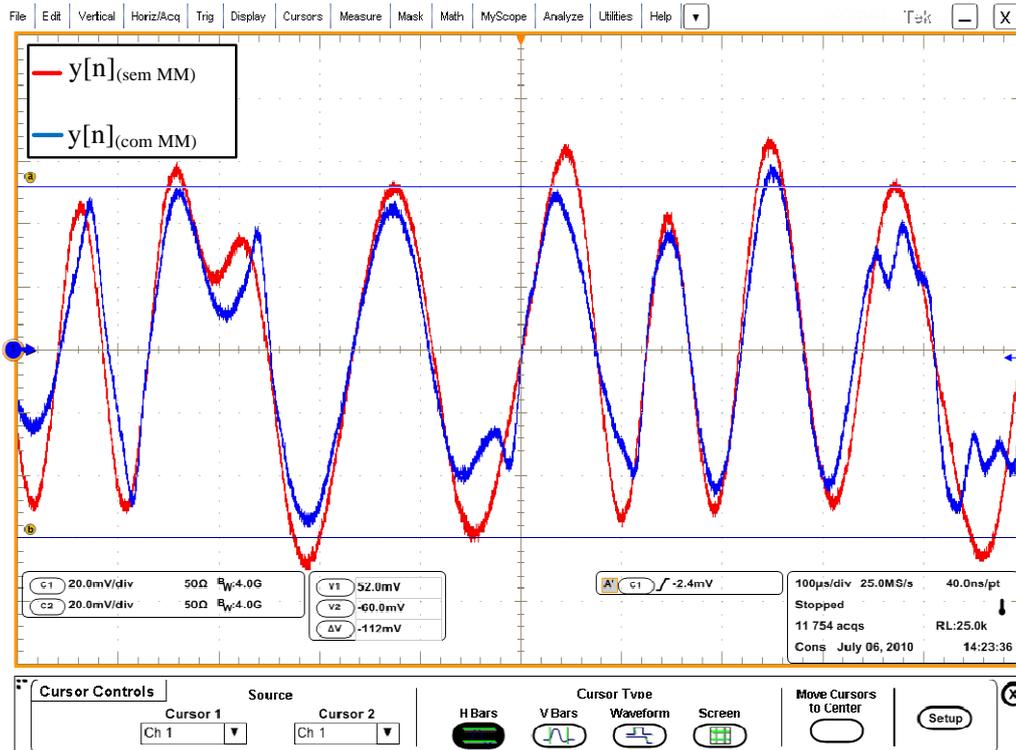


Figura 4.2 – Segmentos da componente real do sinal à saída do filtro RRC para o sistema MPMM_QPSK.

CAPÍTULO 5

Conclusões e considerações finais

Neste capítulo são apresentadas as principais conclusões retiradas durante a realização desta dissertação.

O desenvolvimento do modulador de magnitude no DSP exigiu um conhecimento profundo da arquitectura do DSP, bem como do método MPMM. As técnicas de processamento paralelo e de endereçamento do DSP permitiram desenvolver sistemas MPMM eficientes. A totalidade do código foi escrito em linguagem máquina (*Assembly*), usando linguagem C apenas para a inicialização das variáveis e para a chamada das rotinas.

Com base nos resultados experimentais obtidos em tempo real, ficou validado o método MPMM. Este reduz de forma eficiente o PAPR do sinal de banda limitada a transmitir. Essa redução é obtida evitando picos indesejáveis no sinal de saída. Verificou-se que o sistema é independente da modulação usada, permitindo assim, a generalização do método.

Explorou-se todas as potencialidades do DSP a nível do processamento paralelo de forma a maximizar o desempenho do sistema. No âmbito da decomposição polifásica do filtro RRC, foi utilizado um método que permite realizar várias convoluções em paralelo, aumentando desta forma, a eficiência aritmética do algoritmo.

Para ambos os algoritmos desenvolvidos a frequência máxima de refreshamento do DAC é muito superior à frequência máxima de operação do DAC (96 kHz) o que ilustra o êxito obtido na utilização das técnicas de otimização.

Anexo A

DSP TMS320C6713

É muito importante o conhecimento da máquina quer a nível físico quer a nível de interacção com o *software* específico, com vista ao máximo aproveitamento de todos os recursos posto á nossa disposição pelo DSP.

Os DSP's possuem algumas propriedades que o destacam dos restantes microprocessadores de uso geral. Assim:

- Apresentam um consumo muito menor quando comparado, por exemplo, com um processador *Pentium*.
- Permitem o processamento simultâneo de mais do que uma instrução. No caso do TMS320C6713, este consegue processar até oito instruções em paralelo num único ciclo de relógio, ao contrário do *Pentium* que apenas executa uma instrução por ciclo. Permite ainda vários acessos à memória num único ciclo de instrução, ou seja, verifica-se a existência de uma elevada largura de banda entre a CPU e a memória.
- Estão otimizados para a repetição de operações (*looping*) comuns nos algoritmos de processamento de sinal, ou seja, multiplicar e adicionar (*Multiply and Accumulate - MAC*).
- Contemplam modos de endereçamento especializados, como o endereçamento circular.
- Possuem periféricos próprios que permitem uma *interface* de entrada/saída eficiente com outros dispositivos, como por exemplo, microfone e altifalante.

As características gerais do modelo TMS320C6713 usado são:

- Frequência de relógio: 225 MHz;
- 8 unidades funcionais independentes separadas 4 a 4 em dois lados semelhantes: - L1, L2, S1, S2, M1, M2, D1 e D2. Seis ALUs e dois multiplicadores;
- Possibilidade de execução de 8 instruções, de 32-bit, por ciclo;
- 32 registos internos de 32 bits;
- 32-bit de endereçamento, que permite endereçar até 512 Mbyte de memória externa;
- Tempo de ciclo de 4,44 ns;
- 1800 MIPS (*Million Integer Operations per Second*)
- 1350 MFLOPS (*Million Floating-Point per Second*) ;
- 2 níveis de *cache* em memória interna;
- 16 canais de EDMA (acesso directo à memória configuráveis pela DSP/BIOS);
- 2 *timers* de 32 bits;
- EMIF – Interface com a memória externa

A.1 Arquitectura do processador TMS320C6713

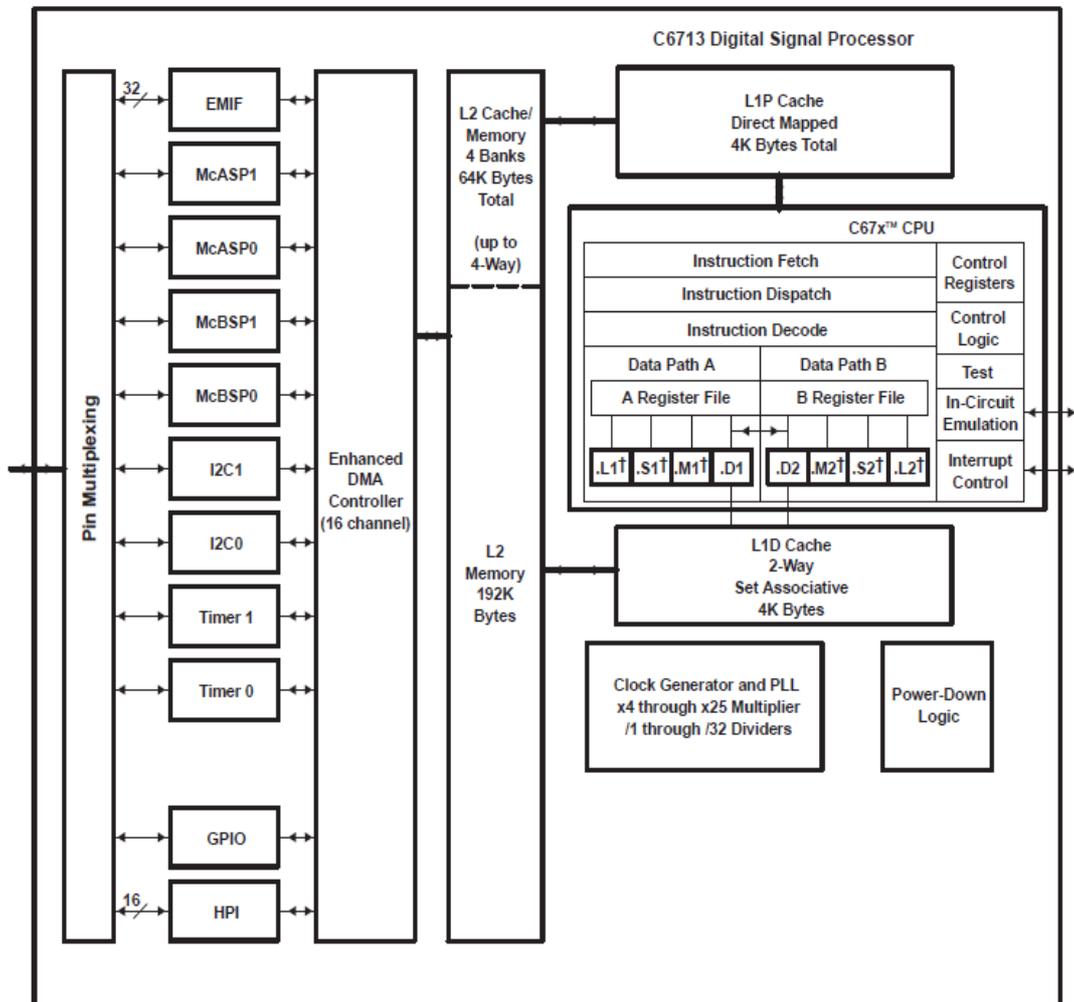


Figura A.1 - Arquitectura do processador TMS320C6713 [14].

Na figura A.1 apresenta-se um diagrama da arquitectura do processador TMS320C6713 onde:

- EMIF - *External memory Interface*;
- HPI - *Host Port Interface*;
- McASP_x - *Multichannel Audio Serial Port*;
- McBSP_x - *Multichannel buffered Serial Port*:
 - São portas podem enviar dados (em série) automaticamente para a memória com a ajuda do controlador DMA;
- I2C_x - *two Inter-Integrated Circuit*;

- Timerx - 2 contadores de 32 bits;
- GPIO - *General-Purpose Input/Output*;
- HPI - *Host-Port Interface*;
- EDMA controller -*Enhanced Direct-Memory-Access* (16 canais independentes);
- L2 Memory/Cache - *Level 2 Memory Cache* (64 kbyte).
- L2 memory - *Level 2 Memory* (192 kbyte);
- L1P - *Level 1 Program Cache* (4 kbyte);
- CPU - *Central Processing Unit*;
- L1D - *Level 1 Data Cache* (4 kbyte);
- PLL – gerador de relógio PLL (*Phase-Locked Loop*) ;
- *Power-down logic*.

A.1.1 Unidade de Processamento Central (CPU)

A Unidade de Processamento Central é constituída por diversos elementos conforme é mostrada na figura A.2.

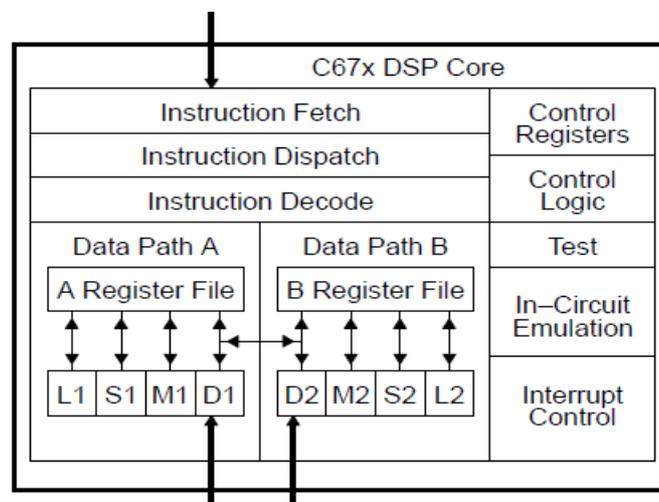


Figura A.2 - CPU do processador TMS320C6713 [14].

CPU Data Paths – Unidades Funcionais

O CPU é composto por dois blocos denominados de *data path* A e *data path* B [4, 5]. Cada bloco é composto por 4 unidades funcionais, .L, .M, .S, .D; 16 registos de 32 bits de uso comum (denominados por A0-A15 para o *data path* A e por B0-B15 para o *data path* B); múltiplas ligações de transferência de dados entre cada bloco e a memória; comunicação de dados entre registos e unidades funcionais constituintes do bloco e ainda de comunicação de dados entre blocos. Nas figuras A.3 e A.4 está representada a estrutura descrita.

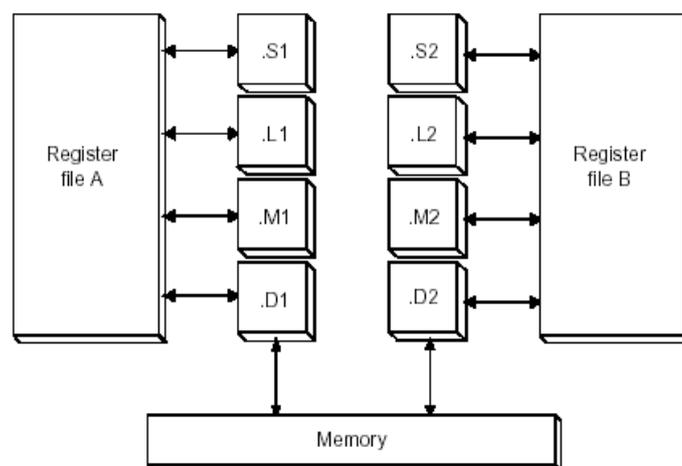


Figura A.3: Unidades funcionais do TMS320C6713.

Os registos são usados para guardar dados, ponteiros para dados, ou como registos condicionais (apenas os registos A2, A1, B0, B1 e B2 podem ser usados como registos condicionais), etc. É também possível efectuar agrupamentos de registos, permitindo desta forma dispor de operandos de 40 e 64 bits.

A cada unidade funcional está associado um grupo específico de operações possíveis de realizar. Assim, a unidade .L é usada para operações lógicas e aritméticas; a unidade .S é usada para operações aritméticas, manipulação de bits e operações de salto; a unidade .M é usada para operações de multiplicação; e a unidade .D é usada para operações aritméticas e de leitura e escrita.

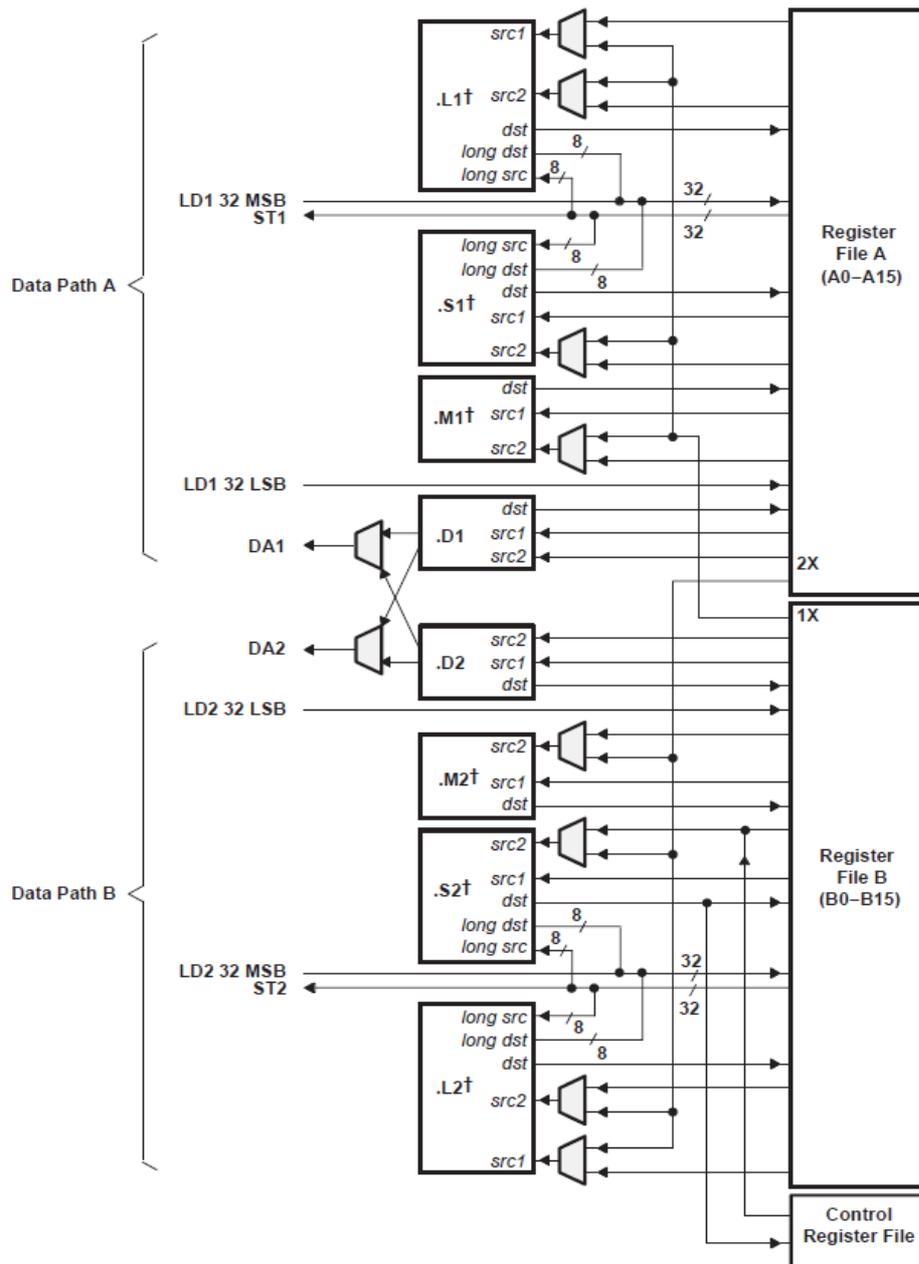


Figura A.4 - *Data Paths* do processador TMS320C6713 [14].

É de referir a existência de *cross paths* que permitem a ligação de um lado do CPU (A ou B) ao outro (B ou A), permitindo a utilização de registos dos dois *data paths*.

Unidade de controlo do programa (Program control unit)

A unidade de controlo do programa é constituída por três elementos:

- *Program fetch unit* (pedido);
- *Instruction dispatch unit* (despacho);
- *Instruction decode unit* (descodificação).

Estas três unidades operam sequencialmente e são necessárias para que a CPU consiga executar as instruções. É de notar que cada unidade é capaz de lidar com oito instruções por ciclo.

Para gerar um *fetch packet* (FP), que corresponde a um grupo de oito instruções, são requeridas quatro fases:

- PG (*phase generate*): o CPU gera o endereço do pedido;
- PS (*phase send*): o CPU envia o endereço para a memória;
- PW (*phase wait*): o CPU espera que os dados fiquem prontos;
- PR (*phase read*): o CPU lê o *opcode* da memória.

Uma vez que o processador tem oito unidades funcionais e cada unidade funcional pode apenas executar instruções específicas, as instruções têm de ser despachadas para as unidades apropriadas. Isto é conseguido pela unidade de despacho (*Instruction dispatch unit*). Quando a instrução *opcode* atinge a unidade de descodificação (*instruction decode unit*), o *opcode* é descodificado.

A.2 DSK C6713

O *Digital Signal Processor Starter Kit* (DSK) TMS320C6713 é uma plataforma de desenvolvimento de baixo custo, onde está inserido o processador TMS320C6713. Contém um interface com o porto série de um computador, o que permite desenvolver e testar aplicações eficientemente. A figura A.5 apresenta uma ilustração do DSK TMS320C6713.

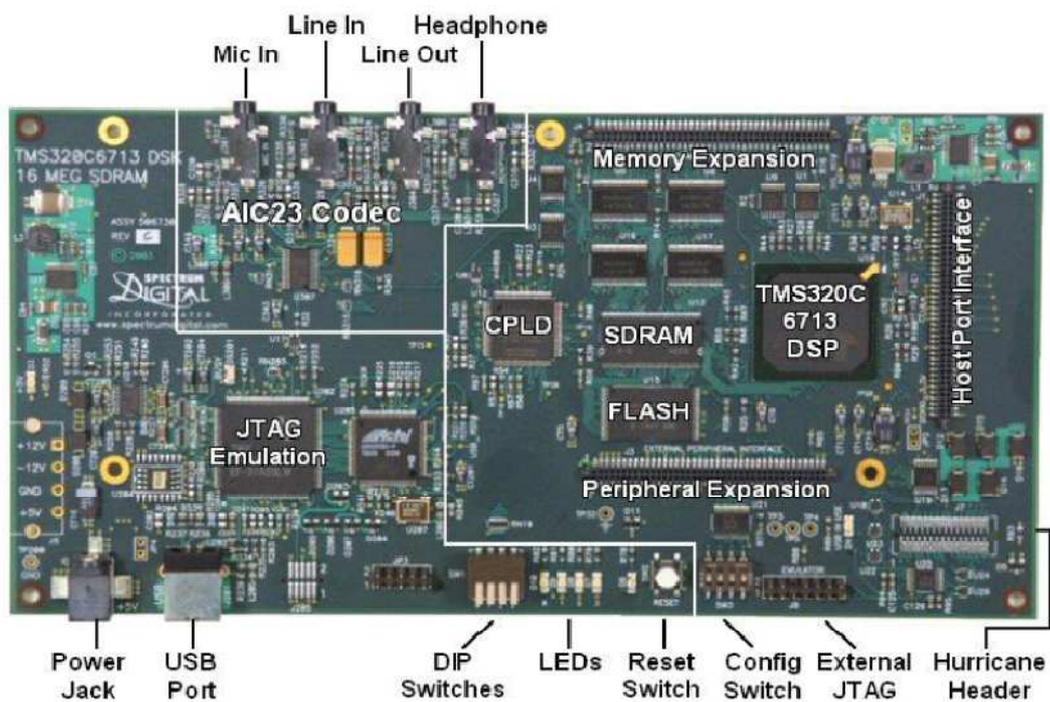


Figura A.5 - Arquitetura do DSK TMS320C6713 [20].

As suas características principais são:

- Processador DSP TMS320C6713 da TI a funcionar a 225MHz;
- AIC23 (ADC e DAC), dois canais analógicos estéreo, com frequência de amostragem de 8 a 96 kHz;
- 512k Bytes de memória *flash*;
- SDRAM: *Synchronous Dynamic RAM* (8 MB / 16 MB)
- Portos de entrada e saída:
 - Entrada: 4 DIP *switch*;

- Saída: 4 LED;
- Interface JTAG (*Joint Test Action Group*): Comunicação por USB com o *host*;
- CPLD : *Complex Programmable Logic Device*;
- Quatro conectores de audio de 3.5mm (microfone, *line-in*, *line-out*, altifalante);
- Tensão de operação 3,3V;
- Formato numérico de vírgula flutuante compatível com normas do IEEE.

A.2.1 Code Composer Studio (CCS)

O DSK TMS320C6713 inclui também o *software Code Composer Studio (CCS)* para programar o DSP.

O CCS é um ambiente de desenvolvimento integrado (IDE) que disponibiliza um *software* simples e fácil de utilizar. Integra um compilador que suporta código em linguagem C ou em linguagem máquina (*assembly linear* ou *assembly*), o que permite alguma versatilidade no projecto. A linguagem C (alto nível) requer menor esforço de programação quando comparada com a linguagem máquina. Por outro lado, usando linguagem máquina devidamente otimizada, o tempo de processamento é muito menor, aumentando assim a eficiência do algoritmo. A figura A.6 ilustra a eficiência *versus* esforço de programação para os três tipos de linguagens de programação suportados pelo CCS para o TMS320C6713: C, *assembly linear*, e *hand optimized assembly* [4, 5, 6].

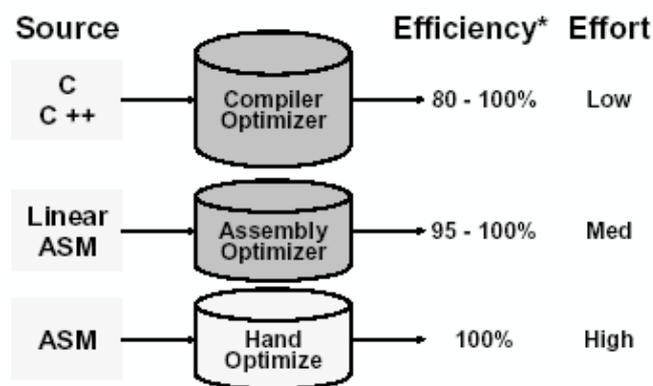


Figura A.6 - Eficiência vs. esforço para as várias linguagens de programação [16].

Anexo B

Representação dos números reais

O TMS32C6713 processa dados do tipo vírgula flutuante, FP (*floating-point*) que podem ser representados por precisão simples e precisão dupla (figura B.1).

No formato de precisão simples, um número é expresso por,

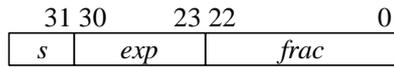
$$-1^S \times 2^{\text{exp}-127} \times 1.\text{frac} , \quad (\text{B.1})$$

onde S representa o bit sinal (bit 31), exp os bits da parte inteira (bits 23 a 30), e frac os bits da parte fraccionária (bits 0 a 22). Consequentemente, números tão grandes quanto 3.4×10^{38} ou tão pequenos como $1,175 \times 10^{-38}$ podem ser representados.

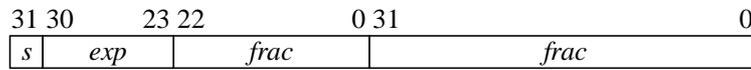
No formato de precisão dupla, estão disponível um maior número de bits para as partes inteira e fraccionaria através do uso de duas *words*. Assim:

$$-1^S \times 2^{\text{exp}-1023} \times 1.\text{frac} . \quad (\text{B.2})$$

Desta forma, números tão grandes como 1.7×10^{308} ou tão pequenos como 2.2×10^{-308} podem ser representados.



(a)



(b)

Figura B.1 - Representação de um número em vírgula flutuante:

(a) precisão simples; (b) precisão dupla.

As duas representações para os números em vírgula flutuante encontram-se definidas na norma ANSI/IEEE Std. 745-1985 [17].

Anexo C

Endereçamento circular

Os modos de endereçamento no processador C6713 podem ser linear ou circular. Uma forma de se obter maior desempenho no processamento é usar registros circulares, utilizando BK0 ou BK1 (tabela C.1). BK0 e BK1 são dois campos de 5 *bits* presentes no registro de modo de endereçamento, AMR (*Addressing Mode Register*). Todos os registros podem fazer endereçamento linear. Contudo, apenas oito podem fazer endereçamento circular: A4-A7 (usados pela unidade D1) e B4-B7 (usados pela unidade D2) [4, 9]. A figura C.1 mostra os modos de endereçamento associados aos registros.

O endereçamento circular assume maior importância quando existe a necessidade de processamento de sinais em tempo real, em que as amostras na saída devem ser produzidas ao mesmo tempo em que as amostras da entrada estão a ser adquiridas. O uso de registros circulares exige a actualização dos índices de endereçamento sempre que uma nova amostra é obtida. Essa actualização é feita através do uso de ponteiros.

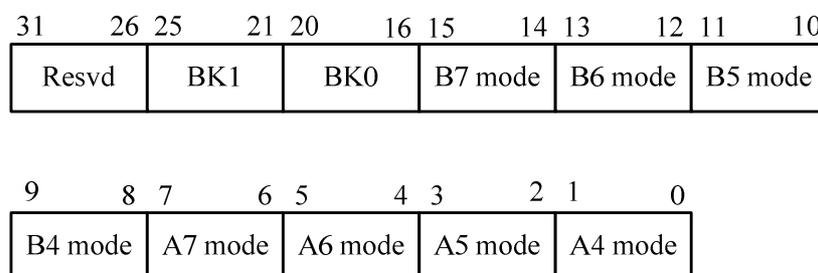


Figura C.1 - Registro de modo de endereçamento, AMR.

Modo	Descrição
00	Para endereçamento linear (por defeito)
01	Para endereçamento circular usando BK0
10	Para endereçamento circular usando BK1
11	Reservado

Tabela C.1 – AMR e descrição.

Bibliografia

- [1] M. Gomes, *et al.*, "Power Efficient Back-off Reduction through Polyphase Filtering Magnitude Modulation", *IEEE Commun. Letters*, vol.13, pp. 606-609, August, 2009.
- [2] M. Gomes, *et al.*, "Polyphase Magnitude Modulation for Peak Power Control," in *European Signal Processing Conference (EUSIPCO)*, Glasgow, 2009, pp. 1151-1155.
- [3] M. Gomes, *et al.*, "Efficient M-QAM Transmission using Compacted Magnitude Modulation Tables," in *IEEE Global Tele-comm. Conf. (GLOBECOM)*, New Orleans, LA, 2008.
- [4] Chassaing Rulph, *Digital Signal Processing and Applications with the C6713 and C6416 DSK*, John Wiley & Sons, 2004.
- [5] Dahnoun, N. *Digital Signal Processing Implementation using the TMS320C6000TM DSP Platform*, Prentice Hall, 2000.
- [6] Dahnoun, N. "C6000 Teaching Materials", *Texas Instruments Incorporated*, 2002.
- [7] Kehtarnavaz, N., Ketamat, M., *DSP System Design: using the TMS320C6000*, Prentice Hall, Upper River, USA, 2001.
- [8] Kehtarnavaz, N., *Real Time Digital Signal Processing based on the TMS320C6000*, Elsevier, 2005.
- [9] Tretter, S., A., *Communication System Design Using DSP Algorithms with Laboratory Experiments for the TMS320C6713TM DSK*, Springer, 2008.

- [10] “*TMS320C67x/C67x+ DSP CPU and Instruction Set Reference Guide*”, SPRU733A, Texas Instruments Incorporated, November 2006.
- [11] “*TMS320C6000 Programmer’s Guide*”, SPRU198I, Texas Instruments Incorporated, March 2006.
- [12] “*TMS320C6000 Code Composer Studio Tutorial*”, SPRU301C, Texas Instruments Incorporated, February 2000.
- [13] “*TMS320C67xx Code Divide and Square Root Floating Point Functions*”, SPRA516, Texas Instruments Incorporated, February 1999.
- [14] “*TMS320C6713 Floating point digital Signal Processor*”, SPRS186L, Texas Instruments Incorporated, November 2005.
- [15] Press William H., Teukolsky Saul A., Vetterling William T., Flannery Brian P., *Numerical Recipes in C*. Cambridge University Press, second edition.
- [16] Teodoro, S., Pereira, S., “*Códigos LDPC em DSP* ”, Relatório da Disciplina de Projecto, Universidade de Coimbra, Coimbra, Portugal 2004.
- [17] *IEEE Standard for Binary Floating-Point Arithmetic, Standards Committee of the IEEE Computer Society*, ANSI/IEEE Std. 745-2008, Institute of Electrical and Electronics Engineers Incorporated, 29 de Agosto de 2008.
- [18] Harris, F. J., *Multirate Signal Processing for Communication Systems*, Prentice Hall PTR, Upper Saddle River, NJ, 2004.
- [19] Ambroze A., Tomlinson M., and Wade G., “*Magnitude modulation for small satellite earth terminals using qpsk and OQPSK*,” in *Proc. IEEE ICC’03*, vol. 3, Anchorage, Alaska, 2003, pp. 2099 – 2103.
- [20] <http://www.deetc.isel.ipl.pt/sistemastele/STRC/>