

## **Célula de Medida de Viscosidade de Líquidos**

**Luís Filipe Fonseca Regada**

Dissertação para obtenção do Grau de Mestre em

**Engenharia Eletrotécnica e de Computadores**

**Júri:**

**Presidente:** Professor Doutor Marcelino Bicho dos Santos

**Orientador:** Professor Doutor José Luís Gonçalves Correia da Mata

**Co-Orientador:** Professora Doutora Maria João Marques Martins

**Orientador Externo:** Professor Doutor Pedro Nuno Mendonça dos Santos

**Vogal:** Professor Doutor Vitor Manuel da Silva Costa

**Outubro de 2012**

# **Célula de Medida de Viscosidade de Líquidos**

**Luís Filipe Fonseca Regada**

Dissertação para obtenção do Grau de Mestre em

**Engenharia Eletrotécnica e de Computadores**

**Júri:**

**Presidente:** Professor Doutor Marcelino Bicho dos Santos

**Orientador:** Professor Doutor José Luís Gonçalves Correia da Mata

**Co-Orientador:** Professora Doutora Maria João Marques Martins

**Orientador Externo:** Professor Doutor Pedro Nuno Mendonça dos Santos

**Vogal:** Professor Doutor Vitor Manuel da Silva Costa



# Agradecimentos

A realização deste trabalho representa o culminar de uma etapa que exigiu muito esforço e dedicação mas que também representa uma recompensa ao olhar para o trabalho realizado e gostar do que se vê. Quero agradecer ao Professor Doutor José Luís Gonçalves Correia da Mata pelo tema proposto e pelo acompanhamento e orientação ao longo da realização deste trabalho. Quero agradecer à Professora Doutora Maria João Marques Martins pelo apoio e motivação que sempre transmite. Quero agradecer ao Professor Doutor Pedro Nuno Mendonça dos Santos pelo incansável empenho e dedicação que demonstrou ao longo de todo o trabalho para que a concretização deste fosse uma realidade. Quero agradecer à Marli Gomes pelo apoio e disponibilidade prestados. Quero agradecer ao Diretor de Curso de Transmissões da Academia Militar Tenente-Coronel de Transmissões Engenheiro João Pedro Pereira Bastos Rocha pelo apoio, preocupação e motivação empregues ao longo de toda a minha formação.

Quero agradecer a todos os meus amigos e familiares que sempre me apoiaram ao longo das várias etapas e que me permitiram atingir esta fase da vida, em especial aos meus pais José Regada e Graciosa Fonseca Regada, aos meus irmãos Miguel Regada e Óscar Regada, à Fátima e ao Alexandre, ao Lúcio Rio, à Lúcia Nunes Rio e à pessoa que esteve sempre comigo, compreendendo e motivando mesmo nos momentos mais críticos, a minha namorada Patrícia Nunes Rio.

A todos vós o meu MUITO OBRIGADO!

# Resumo

O objetivo deste trabalho é o projeto, construção e teste de um sistema que permita a medida de viscosidade de líquidos utilizando o método do fio vibrante em regime livre.

A principal contribuição original deste trabalho é o modo de controlo do estímulo e aquisição de sinal no contexto da medição de viscosidade, utilizando uma interface gráfica (GUI) onde o utilizador pode controlar o estímulo e a taxa de aquisição do sensor. A Field Programmable Gate Array (FPGA) é utilizada para controlar e sincronizar todo o sistema e fazer a ligação entre a interface gráfica e o *hardware* desenvolvido para condicionar o sinal de estímulo do fio vibrante e de resposta do fio vibrante.

A amplitude eficaz máxima de corrente a estimular o fio vibrante  $I_{eficaz} = 0.825A$  permite utilizar fios vibrantes com um diâmetro maior e efetuar medições em líquidos mais viscosos.

Utilizando o protótipo desenvolvido para adquirir a resposta do fio vibrante fez-se o ajuste dos pontos experimentais à equação que descreve o comportamento teórico da resposta do fio vibrante, obtendo-se valores de frequência e amortecimento da resposta do fio vibrante com um desvio padrão de  $\sigma_f = 0.1Hz$  e  $\sigma_\Delta = 5 \times 10^{-5}$  respetivamente, permitindo assim calcular a viscosidade do líquido em estudo.

## **Palavras-chave:**

Condicionamento de Sinal, Fio Vibrante, FPGA, Interface Gráfica, Regime Livre, Viscosidade.

# Abstract

The goal of this project is the design, construction, and testing of a system that allows the measurement of viscosity of liquids using a vibrating wire cell working in free decay mode.

The main contribution of this work is the original mode of controlling the stimulus and the gathering of data in the context of viscosity measurement using a graphical interface (GUI) where the user can control the stimulus intensity and the acquisition rate. A Field Programmable Gate Array (FPGA) is used to control and synchronize the entire system connecting the GUI and the hardware.

The maximum current used to stimulate the vibrating wire ( $I_{RMS} = 0.825A$ ) allow the use of larger diameter vibrating wires leading to the possibility of measuring more viscous liquids.

Using the developed prototype to acquire the response of the vibrating wire and fitting the experimental points to the equation that describes the theoretical behaviour of vibrating wire response, we obtain values of frequency and logarithmic decrement with a standard deviation of  $\sigma_f = 0.1Hz$  and  $\sigma_{\Delta} = 5 \times 10^{-5}$  respectively, allowing compute the viscosity of the tested liquid.

## **Keywords:**

FPGA, Free Decay Mode, Graphical Interface, Signal Conditioning, Vibrating Wire, Viscosity.

# Índice

Agradecimentos.....	iii
Resumo.....	iv
Abstract.....	v
Índice.....	vi
Lista de Acrónimos.....	ix
Lista de Figuras.....	x
Lista de Fluxogramas.....	xiii
Lista de Tabelas.....	xiv
Capítulo 1.....	1
1    Introdução.....	1
1.1    Enquadramento.....	1
1.2    Motivação.....	2
1.3    Objetivos e Especificações.....	3
1.3.1    Objetivos.....	3
1.3.2    Método do fio vibrante.....	4
1.3.3    Cálculo da <i>fem</i> aos terminais do fio vibrante.....	5
1.3.4    Especificações.....	6
1.4    Sistema global.....	7
1.5    Estrutura da dissertação.....	9
Capítulo 2.....	10
2    Estado da Arte.....	10

2.1	Medidores de viscosidades.....	10
2.2	Outras formas de medição de viscosidade de líquidos utilizando o método do fio vibrante .	11
2.3	Métodos de interação com sensores em geral .....	14
Capítulo 3 .....		15
3	Interface gráfica com o utilizador .....	15
3.1	Motivação.....	15
3.2	Implementação.....	16
3.2.1	Estabelecer comunicação USB .....	16
3.2.2	Introdução dos parâmetros de condicionamento do impulso .....	18
3.2.3	Validação e envio dos parâmetros introduzidos .....	23
3.2.4	Receção e tratamento das amostras .....	24
Capítulo 4 .....		25
4	Projeto, simulação e implementação na FPGA .....	25
4.1	FPGA .....	25
4.2	FPGA utilizada.....	26
4.3	Linguagem de descrição de <i>hardware</i> (HDL) .....	27
4.4	Vantagens e desvantagens das FPGAs.....	28
4.5	Visão global.....	28
4.6	Comunicação entra a FPGA e o computador .....	30
4.6.1	Circuito <i>interface_serie</i> .....	30
4.6.2	Circuito <i>UART</i> .....	32
4.6.3	Circuito <i>decode_hexa_chars</i> .....	39
4.6.4	Circuito <i>encode_hexa_chars</i> .....	39
4.7	Circuito <i>interface_placa</i> .....	40
4.8	Circuito aquisição do sinal analógico.....	41
4.8.1	Conversão AD.....	42
4.8.2	Circuito <i>AD1_controller</i> .....	43
4.9	Circuito produção do sinal analógico.....	45
4.9.1	Conversão DA.....	46
4.9.2	Circuito <i>DA2_controller</i> .....	47
4.10	Circuito de sincronização.....	51

4.11	Circuito de interligações.....	54
4.12	Ficheiro UCF .....	56
Capítulo 5	.....	57
5	Condicionamento dos sinais analógicos .....	57
5.1	Visão global.....	57
5.2	Esquema elétrico da montagem.....	59
5.2.1	Deslocador de nível.....	61
5.2.2	Ganho de amplificação selecionável pelo utilizador .....	63
5.2.3	Amplificador de potência .....	64
5.3	Largura de banda .....	64
5.3.1	Largura de banda do sinal de resposta do fio vibrante .....	64
5.3.2	Largura de banda do circuito .....	65
5.4	Ruído .....	66
5.4.1	Ruido de entrada do primeiro andar do bloco de amplificação .....	66
Capítulo 6	.....	70
6	Testes e resultados .....	70
Capítulo 7	.....	78
7	Conclusões e trabalhos futuros.....	78
7.1	Conclusões.....	78
7.2	Contribuições originais.....	80
7.3	Propostas de trabalhos futuros .....	80
Referências	.....	81
ANEXOS	.....	85
Anexo A	.....	86
Anexo B	.....	87
Anexo C	.....	88
Anexo D	.....	91
Anexo E	.....	93

## Lista de Acrónimos

<b>ADC</b>	<i>Analog to Digital Converter</i>
<b>ASIC</b>	<i>Application Specific Integrated Circuits</i>
<b>CGS</b>	Sistema Centímetro-Grama-Segundo de Unidades
<b>CLB</b>	<i>Configuration Logic Blocks</i>
<b>cp</b>	Centipoise
<b>DAC</b>	<i>Digital to Analog Converter</i>
<b>DCM</b>	<i>Digital Clock Manager</i>
<b>DIDP</b>	<i>Diisodecyl phthalate</i>
<b>E/S</b>	Entrada/Saída
<b>fem</b>	Força Eletromotriz Induzida
<b>FPGA</b>	<i>Field Programmable Gate Array</i>
<b>HDL</b>	<i>Hardware Description Language</i>
<b>IOB</b>	<i>Input/Output Blocks</i>
<b>LUT</b>	<i>Look-Up Table</i>
<b>MSB</b>	<i>Most Significant Bit</i>
<b>p</b>	Poise
<b>RAM</b>	<i>Random Access Memory</i>
<b>RAS</b>	Registo de Aproximações Sucessivas
<b>SI</b>	Sistema Internacional de Unidades
<b>SRAM</b>	<i>Static Random-Access Memory</i>
<b>UART</b>	<i>Universal Asynchronous Receiver/Transmitter</i>
<b>UCF</b>	<i>User Constrain File</i>
<b>USB</b>	<i>Universal Serial Bus</i>

## Lista de Figuras

Figura 1.1 - Gradiente de velocidade de um fluido confinado entre uma placa fixa e uma móvel.....	1
Figura 1.2 - Sistema genérico de aquisição e tratamento de dados. ....	2
Figura 1.3 - Esquema de um sistema de controlo de um processo por computador. ....	3
Figura 1.4 - Célula de fio vibrante [2].....	4
Figura 1.5 - Interface entre a interface gráfica, a produção do impulso analógico e a aquisição da resposta do fio vibrante.....	8
Figura 1.6 - Sistema global.....	8
Figura 2.1 - Sistema de aquisição de dado utilizado em [3]. ....	12
Figura 2.2 - Sistema de medida a quatro fios utilizado em [2]. ....	12
Figura 2.3 - Célula do fio vibrante [10].....	13
Figura 2.4 - Circuito de medida [10].....	13
Figura 2.5 - (a) – Montagem do método de medição utilizando o fio vibrante. (b) – Detalhe que mostra o fio, e os ímanes [10].....	13
Figura 3.1 - Aplicação iniciada. ....	16
Figura 3.2 - Características por defeito da comunicação USB com a FPGA.....	16
Figura 3.3 - Comunicação estabelecida.....	17
Figura 3.4 - Inserir parâmetros de condicionamento do impulso a estimular o fio vibrante. ....	19
Figura 4.1 - Modelo conceptual de uma FPGA [27]. ....	25
Figura 4.2 - Estrutura básica do CLB [28].....	26
Figura 4.3 - Arquitetura da família Spartan-3 [29]. ....	26
Figura 4.4 - <i>Spartan-3 Board</i> [30].....	27
Figura 4.5 - Interligação da FPGA no sistema. ....	28
Figura 4.6 - Sistema global a implementar na FPGA. ....	29
Figura 4.7 - Cabo Série-USB.....	30

Figura 4.8 - Recepção de um byte. ....	33
Figura 4.9 - Evolução do registo à medida que se recebe um bit. ....	34
Figura 4.10 - Processamento dos 10 bits recebidos. ....	34
Figura 4.11 - Verificação de erros. ....	34
Figura 4.12 - Definição do bit de paridade <i>par</i> a enviar.....	38
Figura 4.13 - Registo <i>tfSReg</i> carregado, pronto a enviar bit a bit. ....	38
Figura 4.14 - Deslocamento dos bits no registo <i>tfSReg</i> e envio em série para o cabo.....	38
Figura 4.15 - Código VHDL que especifica o circuito <i>decode_hexa_chars</i> para a conversão de ASCII para hexadecimal.....	39
Figura 4.16 - Código VHDL que especifica o circuito <i>encode_hexa_chars</i> para a conversão de hexadecimal para ASCII.....	40
Figura 4.17 - PmodAD1 [33].....	41
Figura 4.18 - Diagrama de blocos do circuito PmodAD1 [33]. ....	41
Figura 4.19 - Diagrama de blocos do circuito AD7476A [34]. ....	42
Figura 4.20 - Diagrama temporal do ADC [34]. ....	42
Figura 4.21 - Função de transferência do ADC [34]. ....	43
Figura 4.22 - PmodDA2 [37].....	45
Figura 4.23 - Diagrama de blocos do circuito PmodDA2 [37]. ....	45
Figura 4.24 - Diagrama de blocos do circuito DAC121S101 [38].....	46
Figura 4.25 - Característica de transferência de entrada / saída do circuito DAC121S101 [38]. ....	46
Figura 4.26 - Diagrama temporal do circuito DAC121S101 [38]. ....	47
Figura 4.27 - UCF com localização dos sinais para DAC, ADC, saídas digitais, relógio e bits enviados e recebidos do cabo <i>Série_USB</i> . ....	56
Figura 5.1 - Sistema global.....	58
Figura 5.2 - Esquema elétrico. Parte 1. ....	59
Figura 5.3 - Esquema elétrico. Parte 2. ....	60
Figura 5.4 - Deslocador de nível.....	62
Figura 5.5 - Espectro do sinal de resposta do fio vibrante para diferentes amortecimentos. ....	65
Figura 5.6 - Andares de amplificação com ganho total de 1000. ....	67
Figura 5.7 - Auxiliar de cálculo da tensão total de ruído de entrada do amplificador do andar de entrada do bloco de amplificação.....	69

Figura 6.1 - Introdução das características do impulso a produzir.....	71
Figura 6.2 - Construção do gráfico com as amostras do impulso produzido.....	71
Figura 6.3 - Relés S3 e S4 sempre a conduzir, não sincronizados.....	72
Figura 6.4 - Relé S4 sempre a conduzir e S3 só a conduzir durante a duração do impulso.....	73
Figura 6.5 - Relés S4 e S3 sincronizados.....	73
Figura 6.6 - Simulação em <i>PSpice</i> do comportamento do deslocador de nível.....	74
Figura 6.7 - Resposta do fio vibrante antes da blindagem eletrostática.....	75
Figura 6.8 - 1ª Resposta do fio vibrante depois da blindagem eletrostática.....	75
Figura 6.9 - 2ª Resposta do fio vibrante depois da blindagem eletrostática.....	76
Figura 6.10 - “Precisão” dos bits de uma amostra.....	77
Figura A.1 - Duração do impulso introduzida pelo utilizador.....	86
Figura B.1 - Amplitude do impulso introduzida pelo utilizador.....	87
Figura C.1 - Diagrama temporal de aquisição da 1ª amostra.....	88
Figura C.2 - Diagrama temporal de aquisição da 2ª amostra.....	89
Figura D.1 - Montagem efetuada sem blindagem.....	91
Figura D.2 - Montagem efetuada com a placa “mais sensível” blindada.....	92
Figura E.1 - 3ª Resposta do fio vibrante depois da blindagem eletrostática.....	93
Figura E.2 - 4ª Resposta do fio vibrante depois da blindagem eletrostática.....	93

## Lista de Fluxogramas

Fluxograma 3.1 - Interface gráfica. Parte 1.....	18
Fluxograma 3.2 - Interface gráfica. Parte 2.....	20
Fluxograma 3.3 - Formatação da duração do impulso a excitar o fio vibrante.....	21
Fluxograma 3.4 - Formatação da amplitude do impulso a excitar o fio vibrante.....	22
Fluxograma 3.5 - Interface gráfica. Parte 3.....	22
Fluxograma 3.6 - Funcionamento do método que trata as amostras recebidas.....	24
Fluxograma 4.1 - Máquina de estados do circuito <i>interface_serie</i> . Parte 1.....	31
Fluxograma 4.2 - Máquina de estados do circuito <i>interface_serie</i> . Parte 2.....	32
Fluxograma 4.3 - Máquina de estados do circuito UART para a receção.....	33
Fluxograma 4.4 - Definição do relógio que comanda a receção <i>rClk</i> .....	35
Fluxograma 4.5 - Definição do relógio que comanda o envio <i>tClk</i> .....	36
Fluxograma 4.6 - Máquina de estados do circuito UART para o envio e sincronismo.....	37
Fluxograma 4.7 - Circuito <i>AD1_controller</i> .....	44
Fluxograma 4.8 - Máquina de estados para controlar a produção do impulso analógico.....	48
Fluxograma 4.9 - Máquina de estados para decidir quais as saídas do circuito <i>DA2_controller</i> .....	50
Fluxograma 4.10 - Máquina de estados do circuito de sincronização.....	54

## Lista de Tabelas

Tabela 3.1 - Esquema de amplificação.....	20
Tabela 5.1 - Auxiliar de cálculo da tensão total de ruído de entrada do amplificador do andar de entrada do bloco de amplificação.....	68
Tabela 6.1 - Quadro resumo dos parâmetros da equação para medidas efetuadas. ....	76
Tabela C.1 - Sequência exemplo de aquisição da 1. <sup>a</sup> amostra. ....	89
Tabela C.2 - Sequência exemplo de aquisição da 2. <sup>a</sup> amostra. ....	90

# Capítulo 1

## 1 Introdução

### 1.1 Enquadramento

A viscosidade dos líquidos é um dos parâmetros do controlo de qualidade. A indústria de processos químicos, nomeadamente a indústria petrolífera, de cremes, tintas, entre outras, necessitam de equipamentos que permitam conhecer o valor da viscosidade dos líquidos utilizados nos processos químicos.

A viscosidade pode ser definida como a medida da resistência de um fluido ao movimento. Ela descreve o atrito interno de um fluido que se observa quando uma camada do fluido se move relativamente a outra. Quanto mais viscoso for o fluido, maior a resistência ao movimento.

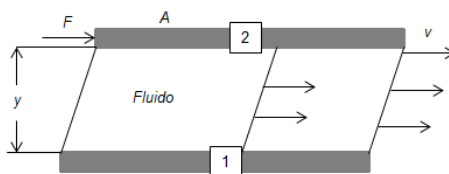


Figura 1.1 - Gradiente de velocidade de um fluido confinado entre uma placa fixa e uma móvel.

A definição do coeficiente de viscosidade de um fluido pode ser observada colocando duas placas de superfície plana e área  $A$  separadas por uma distância  $y$  e uma porção do fluido em estudo entre as mesmas, como na Figura 1.1. Tornando-se a placa 1 fixa e a placa 2 móvel com velocidade  $v$  quando aplicada uma força  $F$  como na Figura 1.1, observa-se que a velocidade do fluido é  $v$  na proximidade da placa 2 e vai diminuindo linearmente com a separação das placas. Constata-se assim que  $F$  é diretamente proporcional a  $v$  e a  $A$  mas é inversamente proporcional a  $y$ . À constante de proporcionalidade que relaciona estas grandezas dá-se o nome de coeficiente de viscosidade, também designado por viscosidade  $\eta$ :

$$F = \eta \frac{v.A}{y} \quad (1.1)$$

No Sistema Internacional de Unidades (SI) a unidade de medida da viscosidade é  $\frac{N.s}{m^2} = Pa.s$ . No sistema *centímetro-grama-segundo* de unidades (CGS) a unidade de medida da viscosidade é o *poise* (*p*), sendo o seu submúltiplo *centipoise* (*cp*) muito utilizado.

$$1Pa.s = 10p = 100cp \quad (1.2)$$

## 1.2 Motivação

A motivação para realização deste trabalho surge da necessidade de medição de viscosidade de líquidos não só com o único objetivo de conhecer o valor da viscosidade do líquido em causa, mas para definir líquidos padrões para calibração de medidores de viscosidade utilizados no mercado.

Outra motivação presente neste trabalho consiste na conceção e construção de um sistema de aquisição e tratamento de dados. Um sistema de aquisição e tratamento de dados é um sistema formado por um conjunto de dispositivos eletrónicos que permite captar dados do “mundo real” e transferi-los para um computador onde um programa adequado permite fazer o tratamento dos dados (Figura 1.2).

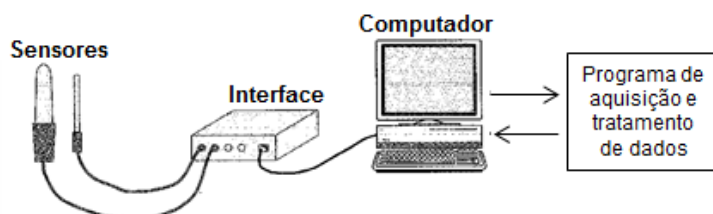


Figura 1.2 - Sistema genérico de aquisição e tratamento de dados.

Num sistema de aquisição e tratamento de dados, um computador liga-se a um ou vários dispositivos (sensores) capazes de captar estímulos físicos (neste caso, proporcionais à viscosidade de líquidos) e de os interpretar ou traduzir eletricamente utilizando conversores analógico/digital (ADC) e posteriormente enviá-los para o computador sob a forma de sinais digitais de modo a que possam ser trabalhados e apresentados sob a forma de gráficos ou outro modo desejado.

Além da aquisição dos dados, pode-se ainda atuar sobre o “mundo analógico”. Nesse caso, um computador tem um programa adequado que permite ao utilizador dizer que ação

pretende efetuar. Esse computador está ligado a um conversor digital/analógico (DAC) que vai interpretar essa ordem digital e a converter num estímulo físico analógico adequado (Figura 1.3).

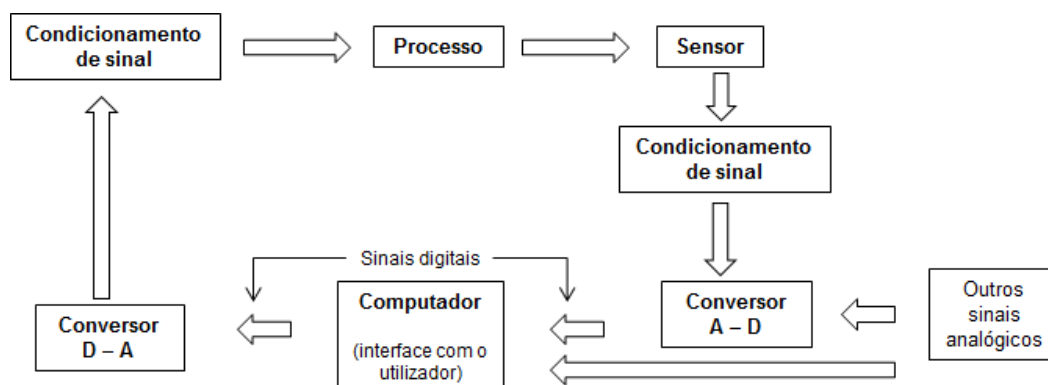


Figura 1.3 - Esquema de um sistema de controlo de um processo por computador.

Este trabalho surge assim como uma oportunidade de realização de um sistema que permite gerar um impulso analógico controlado pelo computador, adquirir a resposta desse impulso, efetuar condicionamento de sinal e ainda o tratamento dos dados. No condicionamento do sinal é necessário ter preocupações com ruído devido às especificações do trabalho. É ainda necessário efetuar uma interface gráfica que permita ao utilizador interagir com o sistema. Tudo isto é feito no contexto de medição de viscosidade de líquidos e com as limitações e especificações impostas pelo método utilizado, o método do fio vibrante em regime livre.

## 1.3 Objetivos e Especificações

### 1.3.1 Objetivos

O objetivo do trabalho é o projeto, construção e teste de um sistema portátil de estímulo e aquisição de dados para um sensor que permita obter a medição da viscosidade de líquidos utilizando o método do fio vibrante em regime livre. O sistema deverá permitir a caracterização da resposta da célula de fio vibrante em regime livre.

As especificações são impostas pelas limitações e restrições que o método utilizado impõe. Para se perceber essas limitações apresenta-se de seguida o método do fio vibrante.

### 1.3.2 Método do fio vibrante

A célula de fio vibrante foi desenvolvida e validada em [1] e consiste numa cápsula com um fio vibrante mergulhado numa amostra de líquido, num banho com temperatura controlada e um sistema de medição com quatro fios. [2]



Figura 1.4 - Célula de fio vibrante [2].

O transdutor do viscosímetro de fio vibrante é constituído por um fio de tungsténio fixo nas extremidades (observar Figura 1.4) e sujeito a uma força axial. O fio está situado no interior de uma cápsula e mergulhado no líquido em estudo. Externamente à cápsula estão uns ímanes permanentes que vão criar um campo magnético perpendicular ao fio.

Para estimular o fio vibrante faz-se passar uma corrente alternada através do fio, obtendo-se uma força que provoca um movimento de oscilação transversal no fio vibrante. Este movimento do fio sob ação de um campo magnético induz uma diferença de potencial nos terminais do fio [2]. Este efeito é conhecido como força eletromotriz induzida (*fem*). Pode-se utilizar o método do fio vibrante em regime livre ou em regime forçado. Ambos os métodos consistem em colocar o fio em movimento e detetar a tensão induzida aos seus terminais.

O regime forçado consiste em aplicar um sinal sinusoidal de amplitude constante ao fio fazendo variar a frequência e posteriormente medir a tensão induzida aos terminais do fio em função da frequência.

Em regime livre, o fio é estimulado com um impulso de corrente simétrico retangular que vai fazer com que este deixe a sua posição de repouso. A resposta do fio a este impulso de corrente é uma oscilação amortecida em função do amortecimento do líquido em estudo e do coeficiente de amortecimento do material do fio. As oscilações amortecidas são detetadas devido à tensão induzida aos terminais do fio. O sinal de resposta deve ter a seguinte forma [3]:

$$V(t) = Ae^{-\Delta\omega t} \cos(\omega t + \varphi) + V_1 \quad (1.3)$$

Este trabalho refere-se exclusivamente ao método do fio vibrante em regime livre.

A amplitude do sinal  $V(t)$  (tensão induzida aos terminais do fio) e o instante temporal  $t$  são obtidos pelo método experimental. A amplitude  $A$ , a taxa de decaimento  $\Delta$ , a frequência de oscilação  $\omega$ , a fase  $\phi$  e a componente contínua  $V_1$  são obtidos através de um ajuste não linear dos dados experimentais [1]. O objetivo deste trabalho é determinar os parâmetros que definem a equação (1.3).

Sabe-se que o movimento do fio depende do seu raio, da densidade, viscosidade e temperatura do líquido. No regime livre, se conhecer a frequência de oscilação, a taxa de decaimento e a densidade do líquido pode-se calcular o valor da viscosidade do líquido em estudo [1]. A relação entre a taxa de decaimento e a viscosidade do meio está modelada em [4].

### 1.3.3 Cálculo da *fem* aos terminais do fio vibrante

O valor instantâneo da *fem* depende da velocidade com que o condutor atravessa o campo, da densidade de fluxo magnético, do comprimento do condutor inserido no campo e do ângulo formado entre a direção do movimento do condutor e a direção do campo magnético.

$$e = \beta \cdot L \cdot V \cdot \sin(\alpha) \quad (1.4)$$

- $e$  = Valor instantâneo da *fem* no condutor, em volts (V)
- $\beta$  = Densidade de fluxo magnético, em Tesla (T)
- $L$  = Comprimento do condutor inserido no campo magnético, em metros (m)
- $V$  = Velocidade constante com que o condutor atravessa o campo magnético, em metros por segundo (m/s)
- $\sin(\alpha)$  = seno do ângulo formado entre a direção do movimento do condutor em relação com a direção do campo magnético.

Tendo em conta que o campo magnético é perpendicular ao fio ( $\alpha = 90^\circ$ ), obtém-se:

$$e_{max} = \beta \cdot L \cdot V_{max} \quad (1.5)$$

O fio vibrante utilizado tem diâmetro  $D = 100\mu\text{m}$  e comprimento  $L = 4\text{cm}$ . Está sujeito a  $\beta \approx 0.1\text{T}$ . O líquido utilizado nos testes tem uma frequência de ressonância de cerca de  $f \approx 1000\text{Hz}$ . Admitindo que o movimento de oscilação transversal do fio vibrante  $Y(t)$  é cerca de 1% do seu diâmetro, a amplitude máxima do movimento oscilatório é  $A \approx 1\mu\text{m}$ .

Sabendo:

$$V_{\max} = \left(\frac{dy}{dt}\right)_{\max} \quad e \quad Y(t) = A \sin(2\pi ft) \quad (1.6)$$

Então:

$$V_{\max} = A \times 2\pi f = 1 \times 10^{-6} \times 2\pi \times 1000 = 6.28 \times 10^{-3} \left(\frac{m}{s}\right) \quad (1.7)$$

Introduzindo o resultado (1.7) em (1.5) obtemos:

$$e_{\max} = 0.1 \times 0.04 \times 6.28 \times 10^{-3} = 25\mu V \quad (1.8)$$

Em [5] o fio utilizado tinha diâmetro  $D = 0.3\text{mm}$  e um comprimento  $L = 2\text{cm}$  e o campo magnético  $\beta = 0.3\text{T}$ . A frequência de ressonância era da ordem dos  $3100\text{Hz}$ .

Efetuada o mesmo raciocínio mas agora para os valores do artigo [5] obtém-se:

$$V_{\max} = A \times 2\pi f = 3 \times 10^{-6} \times 2\pi \times 3100 = 58.43 \times 10^{-3} \left(\frac{m}{s}\right) \quad (1.9)$$

O que leva ao seguinte resultado:

$$e = 0.3 \times 0.02 \times 58.43 \times 10^{-3} = 350\mu V \quad (1.10)$$

Observando-se os resultados obtidos em (1.8) e (1.10) resultantes da utilização do fio vibrante em estudo e do utilizado em [5] respetivamente, constata-se que os valores máximos da *fem* aos terminais do fio vibrante são da ordem de grandeza das dezenas a centenas de microvolts ( $\mu V$ ). Como a resposta do fio vibrante a adquirir é um transitório amortecido, a *fem* vai ser atenuada e poderá ser detetada até que esta seja da ordem de grandeza do ruído de entrada do sistema de amplificação, nomeadamente o primeiro andar de amplificação. Deste modo é necessário ter uma grande preocupação com o ruído eletromagnético e ter vários andares de amplificação seleccionáveis pelo utilizador permitindo amplificar o sinal cerca de cinco mil a cem mil de vezes para o poder adquirir.

#### 1.3.4 Especificações

Tendo em conta o método do fio vibrante em regime livre e as suas limitações, o sistema tem as seguintes especificações:

- O sistema tem que produzir um impulso de corrente simétrico que estimule o fio vibrante.
- A duração e amplitude desse impulso devem ser controladas pelo utilizador através de uma interface gráfica. Essa interface gráfica deverá também permitir ao utilizador escolher o ganho de amplificação da resposta do fio vibrante.
- O sistema tem que adquirir a resposta do fio vibrante que é um sinal com amplitude máxima de ordem das dezenas/centenas de microvolts com a forma de uma senoide amortecida.
- O sistema tem que amplificar o sinal de resposta do fio vibrante para o poder adquirir e minimizar o ruído.
- Pretende-se adquirir 1000 amostras com frequência de amostragem de 100kHz, ou seja, período de amostragem de 10 microssegundos ( $\mu s$ ) que corresponde a 10 milissegundos ( $ms$ ) de sinal. No entanto, deve-se proporcionar ao utilizador decidir o tempo de sinal que pretende amostrar, controlando a frequência de amostragem. Essa escolha deve ser feita através da interface gráfica.
- Após adquirir o sinal é necessário enviá-lo para um computador. Esse computador deverá ter uma interface gráfica onde deverá receber as amostras da resposta do fio vibrante e fazer o tratamento desses dados, permitindo fazer um gráfico ilustrativo do amortecimento do sinal e guardar esse gráfico, assim como os pontos do gráfico.
- Tendo em conta o espectro da resposta do fio vibrante a adquirir, é necessário garantir que a resposta em frequência dos amplificadores não provoca distorção ao sinal a adquirir.
- É necessário garantir que o ruído introduzido pela amplificação não compromete o sistema.

O resultado esperado é um protótipo funcional do sistema.

## 1.4 Sistema global

Tendo em conta que o objetivo do trabalho é o projeto, construção e teste de um sistema portátil de estímulo e aquisição de dados para um sensor que permita medir a viscosidade de líquidos utilizando o método do fio vibrante em regime livre, optou-se por dividir o problema em três blocos.

O primeiro bloco consiste em produzir uma interface gráfica que defina as características do impulso de corrente a estimular o fio vibrante, nomeadamente a duração e

amplitude. Deve ainda permitir que o utilizador escolha quanto tempo pretende amostrar a resposta do fio vibrante e quantas vezes pretende amplificar a resposta do fio vibrante. Esta interface gráfica deve ainda permitir receber os pontos da resposta do fio vibrante, gravá-los num ficheiro externo e construir um gráfico através desses pontos que possa ser gravado no local e formato desejado.

O segundo bloco consiste em fazer uma interface entre a interface gráfica com o utilizador e a produção do impulso de corrente e a aquisição da resposta do fio vibrante (Figura 1.5). Esta interface deve então controlar um DAC para produção do impulso a estimular o fio vibrante de acordo com as opções do utilizador e controlar um ADC para adquirir a resposta de acordo com as opções do utilizador. Esta interface deve ainda guardar todas as amostras da resposta do fio vibrante e só depois de adquirir todas as amostras é que as envia para o computador. Para tal, é necessário definir a comunicação entre esta interface e o computador. Além destas funcionalidades, esta interface deve permitir controlar um conjunto de relés para seleccionar o ganho da amplificação em função das opções do utilizador e outro conjunto de relés para sincronizar o estímulo do fio vibrante e a aquisição da resposta do mesmo garantindo que não há corrente a percorrer o fio vibrante quando não for desejado.

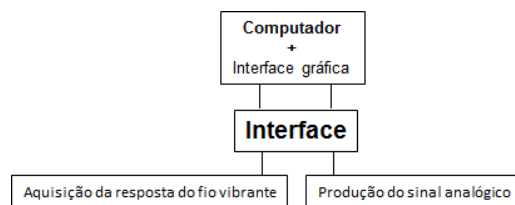


Figura 1.5 - Interface entre a interface gráfica, a produção do impulso analógico e a aquisição da resposta do fio vibrante.

O terceiro bloco consiste em produzir e condicionar o impulso de corrente a estimular o fio vibrante. Deve-se ainda amplificar a resposta do fio vibrante, efetuar a blindagem necessária e adquirir a resposta do fio (Figura 1.6).

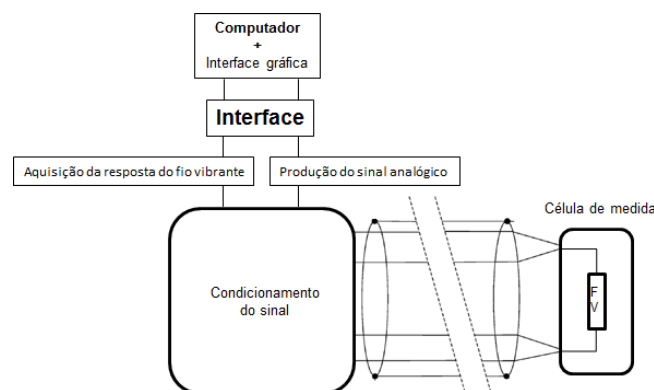


Figura 1.6 - Sistema global.

## 1.5 Estrutura da dissertação

A dissertação está organizada em sete capítulos.

O **Capítulo 1 – Introdução** faz o enquadramento, motivação, define os objetivos e as especificações do trabalho.

O **Capítulo 2 – Estado da Arte** apresenta a pesquisa efetuada relativa aos métodos de medição de viscosidade de líquidos e seguida apresenta outras formas de medição de viscosidade de líquidos utilizando o mesmo método que o utilizado nesta dissertação. São também apresentados outros tipos de atuação sobre sensores.

O **Capítulo 3 – Interface gráfica com o utilizador** onde é explicada a interface gráfica com o utilizador, correspondente ao primeiro bloco da secção 1.4.

O **Capítulo 4 – Projeto, simulação e implementação na FPGA** onde é apresentada a placa de desenvolvimento utilizada e explicada a interface entre a interface gráfica e a produção do impulso a estimular o fio vibrante e a aquisição da resposta do fio vibrante, correspondente ao segundo bloco secção 1.4.

O **Capítulo 5 – Condicionamento dos sinais analógicos** onde são apresentadas as opções realizadas no condicionamento do impulso a estimular o fio vibrante e no condicionamento da aquisição da resposta do fio vibrante, correspondente ao terceiro bloco secção 1.4. É neste capítulo que se calcula a largura de banda do sistema e da resposta do fio vibrante e onde se calcula o ruído de entrada nos andares de amplificação.

O **Capítulo 6 – Testes e resultados** é onde são apresentados os testes realizados ao sistema implementado e respetivos resultados.

O **Capítulo 7 – Conclusões e trabalhos futuros** onde são apresentadas as conclusões do trabalho efetuado e onde se efetua uma análise global aos resultados obtidos. São ainda apresentadas as contribuições originais e propostas para trabalho futuro.

# Capítulo 2

## 2 Estado da Arte

### 2.1 Medidores de viscosidades

O estudo da viscosidade de líquidos é necessário para o controlo de qualidade em várias indústrias e pode ser feito tendo em conta diversos objetivos como a indústria farmacêutica, cosmética, alimentar, química, análises clínicas, construção civil, petrolífera, definição de líquidos *standard* como a água, o tolueno ou *diisodecyl phthalate (DIDP)* para posteriormente ser utilizado para calibrar outros medidores de viscosidade [3] [6] [7].

Na indústria farmacêutica e cosmética é necessário saber o tempo de escoamento do líquido no frasco e o modo como o creme se espalha. Na química é útil para saber como é que a cola flui do recipiente, as tintas escorrem, etc. Se o objetivo do estudo da viscosidade de um líquido for, por exemplo, o escoamento do fluido para saber a resistência deste ao cisalhamento que vai dificultar o seu fluxo em tubulações, cavidades de moldes, entre outros, ou então a produção de tintas em que é necessário controlar a viscosidade da tinta para garantir determinadas propriedades desejadas é necessário um aparelho que meça a viscosidade do líquido em estudo e forneça um valor com uma determinada precisão. Mas se o objetivo do estudo da viscosidade de um líquido for a definição de um líquido *standard* para posteriormente ser utilizado para calibrar outros medidores de viscosidade, então a precisão da medida tem que ser muito maior do que no caso anterior [3] [7].

Existem diferentes tipos de viscosímetros baseados em diferentes métodos para se medir a viscosidade de líquidos. Tendo em conta a definição de viscosidade apresentada na secção 1.1 a medição da viscosidade é feita em condições dinâmicas ou cinemáticas. Ou se faz mover o líquido em estudo através de um obstáculo de geometria bem definida ou então faz-se mover um corpo de geometria bem definida mergulhado no líquido em estudo. Os viscosímetros para medição da viscosidade de líquidos podem ser divididos em duas

categorias, os primários (viscosímetros rotativos e de objetos vibrantes) e os secundários (viscosímetros capilares e de esfera em queda) [2] [8] [9].

Para qualquer método de medida utilizado é indispensável obter um modelo teórico que determine as características do movimento, quer do líquido em estudo quer dos sólidos imersos nele, que tenham em conta a geometria da situação e a relacionem com a viscosidade do líquido em estudo. O viscosímetro de fio vibrante é constituído por um fio metálico de secção cilíndrica fixo nas extremidades, imerso no líquido em estudo e efetua oscilações transversais (apresentado na seção 1.3.2). A célula de fio vibrante foi desenvolvida e validada em [1], assim como o modelo teórico.

## **2.2 Outras formas de medição de viscosidade de líquidos utilizando o método do fio vibrante**

Tal como já foi referido, o método do fio vibrante permite operar em regime livre e/ou em regime forçado.

Em [3] é utilizado o método do fio vibrante em regime livre e o sistema é controlado por um computador através de um barramento *IEC625* (Figura 2.1). Para gerar o impulso que estimula o fio vibrante foi utilizado um sintetizador "*Stanford Research Systems DS345*". Este gera um impulso quadrado que é ligado a um amplificador de potência *E* e ligado em série a um interruptor *K* e a uma resistência *R*. O amplificador *D* recolhe e amplifica a oscilação transitória resultante das oscilações livres do fio. A resposta da célula é posteriormente gravada por um digitalizador "*Advantech Ltd.*" (*PC\_Labcard*, modelo *PCL-318*, 12 bit e 105 amostras por segundo). Durante a aplicação do impulso de corrente foram utilizados dois relés *reed*. O interruptor *S* é mantido aberto para evitar a saturação do amplificador *D*. O interruptor *K* abre após o final do período do impulso de corrente de modo a garantir que a resistência *R* e a saída do amplificador *E* não ofereçam um caminho para a corrente e possam permitir leituras "falsas". Os amplificadores *D* e *E* estão ligados ao exterior desse módulo através de isoladores óticos que impedem o fluxo de correntes de ruído e permitem que o nó de terra não esteja a flutuar e seja ligado à gaiola de Faraday que envolve todo o circuito e está ligado à parede externa da célula de medição, proporcionando uma proteção contra o ruído externo.

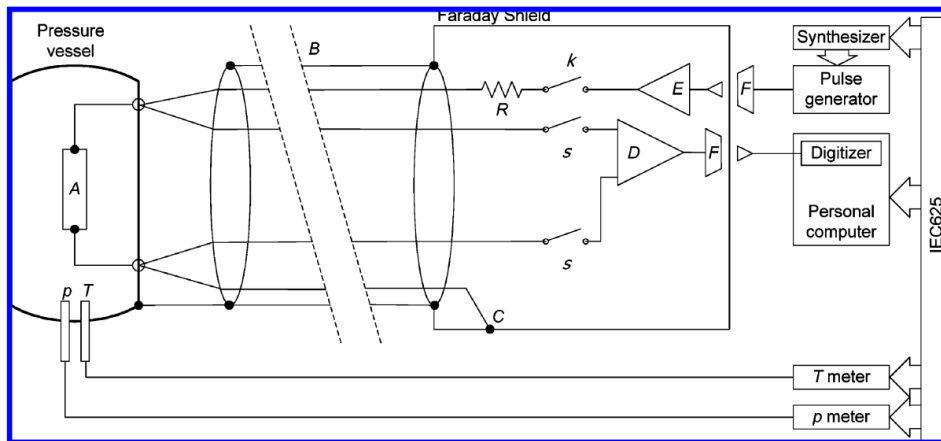


Figura 2.1 - Sistema de aquisição de dado utilizado em [3].

Em [2] é utilizado o método do fio vibrante em regime forçado, onde se desenvolve um sistema de medição a quatro fios que consiste num gerador de funções "Agilent 33250" que fornece o impulso que estimula o fio vibrante, colocado em série com uma impedância de referência  $Z_R=100\Omega$  que por sua vez está colocada em série com a célula de fio vibrante. A resposta do fio vibrante em tensão é adquirida por uma placa de aquisição "NI 6251" com 16 bits de precisão. Foi também desenvolvida uma aplicação em *LabVIEW* através de uma interface *GPIB* para controlar o gerador de funções e adquirir e processar os sinais adquiridos. O sistema utilizado foi o da Figura 2.2.

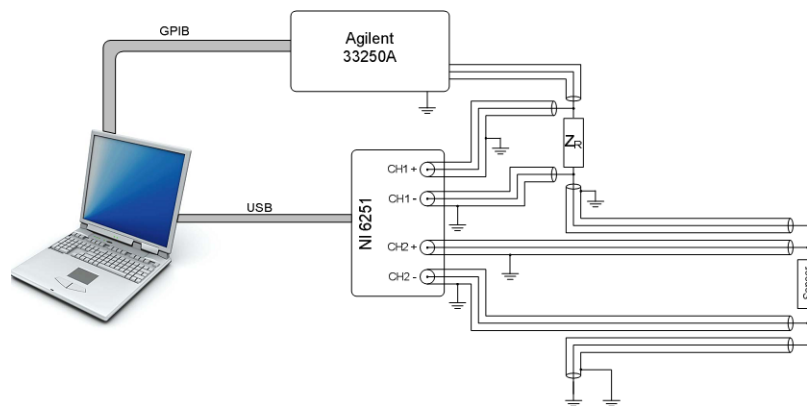


Figura 2.2 - Sistema de medida a quatro fios utilizado em [2].

Também em [10] é utilizado o método do fio vibrante em regime forçado para medição da viscosidade e uma medição a quatro fios para estimular e detetar a vibração do fio (Figura 2.3).

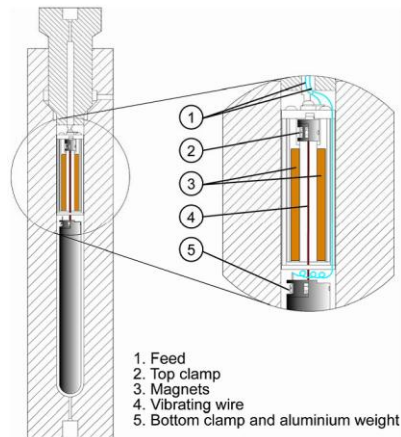


Figura 2.3 - Célula do fio vibrante [10].

Para estimular o fio vibrante foi utilizado um sintetizador “*Stanford Research Systems model SRS830 lock-in amplifier*” que pode operar com tensões entre  $0.004V_{RMS}$  e  $5.000V_{RMS}$  e a frequência pode ser ajustada com incrementos de 0.1Hz.

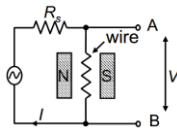


Figura 2.4 - Circuito de medida [10].

Para obter uma condução de corrente constante foi utilizada uma resistência em série com  $R_s = 1.0k\Omega$  (Figura 2.4). Escolheu-se este valor por ser muito maior do que a resistência do fio que é cerca de  $0.5\Omega$ . Os terminais A e B foram ligados às entradas diferenciais do amplificador *lock-in* (Figura 2.5).

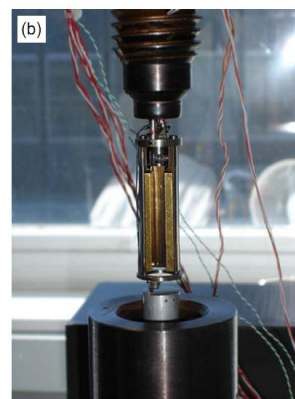
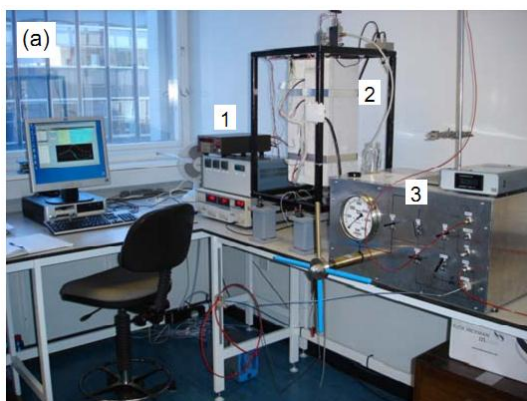


Figura 2.5 - (a) – Montagem do método de medição utilizando o fio vibrante: 1 – Instrumentos. 2 – Célula de medida isolada. 3 – Sistema de manuseamento de fluidos. (b) – Detalhe que mostra o fio e os ímãs [10].

## 2.3 Métodos de interação com sensores em geral

Existem diferentes tipos de sensores. Existem sensores de luminosidade (fotodíodos), de som (microfones), de temperatura (termómetro), de pressão (barómetro), de resistência elétrica (ohmímetro), de corrente elétrica (amperímetro), de tensão elétrica (voltímetro), de potência elétrica (wattímetro), de orientação (giroscópio), de movimento (velocímetro), de posição, de proximidade, etc.

Um sensor é um dispositivo que recebe e responde a um estímulo e que converte a resposta desse estímulo num sinal elétrico. Para comandar esse sensor e para adquirir e tratar o sinal pode-se recorrer a várias ferramentas.

Para interagir com sensores, nomeadamente com a célula de fio vibrante quer em regime livre quer em regime forçado, pode-se recorrer a um sintetizador “*Stanford Research Systems DS345*”, gravar os dados recorrendo a um digitalizador “*Advantech Ltd.*” e controlar o sistema com um computador ligado ao sistema através de um barramento *IEC625* como em [3].

Pode-se também utilizar, como em [2], o gerador de funções “*Agilent 33250*”, adquirir os dados com uma placa de aquisição “*NI 6251*” e controlar o sistema com uma aplicação em *LabVIEW* utilizando uma interface *GPIB* ou mesmo *USB*.

Outra abordagem para interagir com sensores (seja de fio vibrante ou de outro tipo) consiste em utilizar um microcontrolador (que já contém um processador, memória e periféricos de entrada e saída programáveis) ligados a ADC e/ou DAC. Neste campo existe uma grande variedade de trabalhos efetuados em técnicas para interfaces com sensores [11], em redes de sensores [12], identificação de fugas de gás [13], monitorização e controlo de temperatura [14] [15], deteção de alvos [16] e inclusive um controlador para um viscosímetro de fio vibrante [17].

Além destas abordagens, pode-se ainda utilizar *Field Programmable Gate Array* (FPGA) para interagir com sensores em geral. Existem diversas aplicações de FPGAs em redes de sensores [18] [19] [20] [21], monitorização de temperatura [22], sensores de humidade [23], sensores de posição [24] [25], sensores de deteção de vibrações em robôs industriais [26], entre outros.

# Capítulo 3

## 3 Interface gráfica com o utilizador

### 3.1 Motivação

A interface com o utilizador deve permitir ao utilizador controlar todo o sistema. Deve ser nesta interface que:

- O utilizador define as características do impulso de corrente a estimular o fio vibrante, nomeadamente a duração e a amplitude.
- O utilizador define a frequência de amostragem.
- O utilizador seleciona a amplificação da resposta do fio vibrante.
- A resposta do fio vibrante é reconstruída sob a forma de um gráfico.
- Guarda os pontos experimentais da resposta do fio vibrante e o gráfico.
- Se estabelece a comunicação pela porta *Universal Serial Bus* (USB) do computador com a FPGA.

Para se desenvolver a interface gráfica utilizou-se o programa “*Microsoft Visual Studio 2010*” e a linguagem de programação “*Visual C#*”, recorrendo a uma “*Windows Forms Application*”. Esta escolha deve-se ao facto de tanto o programa de desenvolvimento como a linguagem de programação mencionados serem desenvolvidos pela *Microsoft* para ambiente *Windows*. Por sua vez, desenvolver uma aplicação “*Windows Forms*” permite obter uma interface apelativa, intuitiva e gráfica, permitindo receber as ordens do utilizador através de caixas de texto e botões e enviá-las para a FPGA, comunicando através das portas USB, receber os dados da FPGA novamente através das portas USB, desenhar gráficos e gravá-los no local e formato desejado, assim como os pontos do gráfico, tudo isto em ambiente *Windows standard* e numa única aplicação.

## 3.2 Implementação

### 3.2.1 Estabelecimento da comunicação USB

A aplicação é iniciada com um duplo clique sobre o executável “Medidor\_de\_Viscosidade.exe”. Quando a aplicação é iniciada tem o seguinte aspeto:

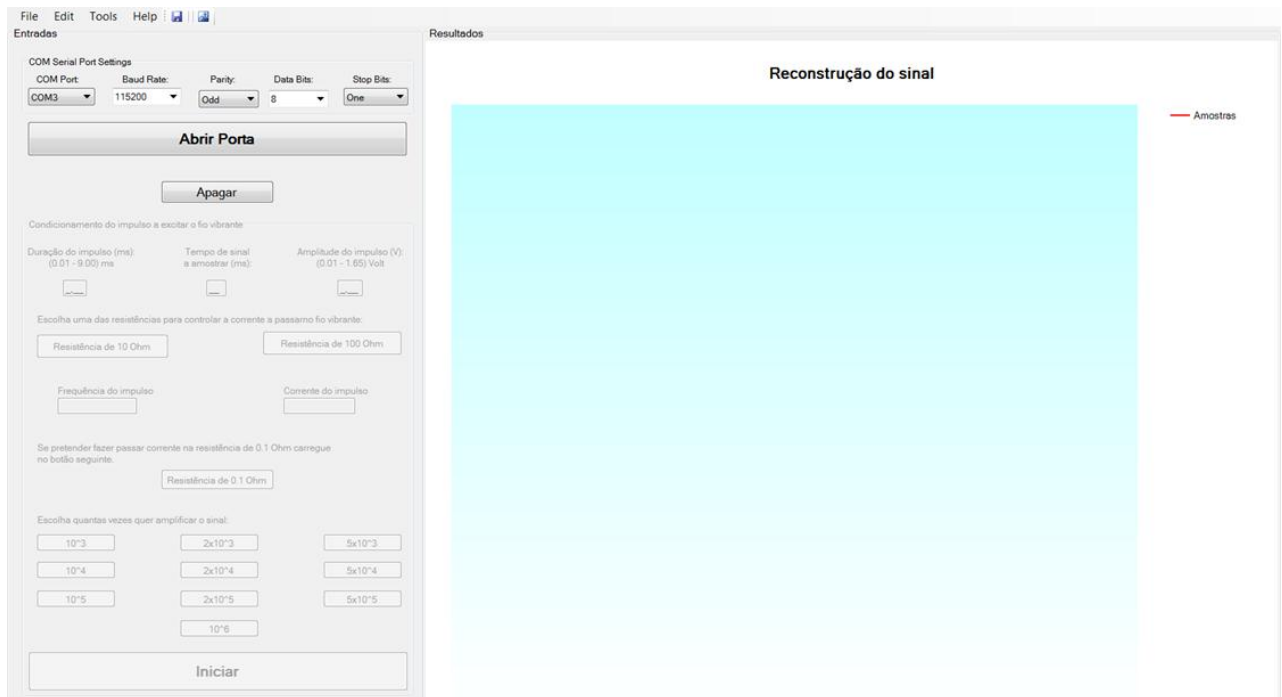


Figura 3.1 - Aplicação iniciada.

A aplicação desenvolvida começa por definir as características da comunicação USB, atualizar as portas USB disponíveis para se poder efetuar a comunicação com a FPGA e definir as características por defeito para a comunicação USB com a FPGA, sendo estas:

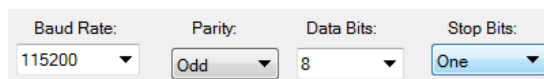


Figura 3.2 - Características por defeito da comunicação USB com a FPGA.

A razão de terem sido escolhidos estes valores deve-se ao facto de se ter implementado uma *Universal Asynchronous Receiver/Transmitter* (UART) na FPGA para estabelecer a comunicação com o computador com estas características (explicada na secção 4.6.2). Foi dada a possibilidade de o utilizador poder alterar as características da comunicação para o caso de se pretender utilizar outra FPGA ou outro equipamento de comunicação que

substitua a FPGA ou mesmo alterar a UART implementada sem ser necessário alterar a interface gráfica.

Se o utilizador pretender alterar as características da comunicação pode alterar e clicar no botão “Abrir Porta” para estabelecer a comunicação. Se o utilizador pretender utilizar a comunicação com as características por defeito basta clicar no botão “Abrir Porta”.

Nesta fase a aplicação limpa a área dos resultados, onde deverá ser apresentado o gráfico com a reconstrução da resposta do fio vibrante (à direita da janela) e inicializa o valor das variáveis que vão guardar as características do impulso analógico a produzir. Esta ação é realizada sempre que se clica no botão “Abrir Porta” ou quando se clica no botão “Apagar”, permitindo fazer novo ensaio experimental.

De seguida a aplicação verifica se já existe alguma ligação USB estabelecida na porta selecionada. Se existir, a aplicação fecha essa ligação. Se não existir, a aplicação estabelece a ligação com as características definidas e verifica se houve erros. Se houver erros informa o utilizador e termina a aplicação. Se não houver erros, estabelece a comunicação e atualiza a janela, alterando o botão “Abrir Porta” para “Fechar Porta”, desativando a zona de configuração da ligação USB e ativando a zona do condicionamento do impulso a estimular o fio vibrante. A janela fica agora com o seguinte aspeto:

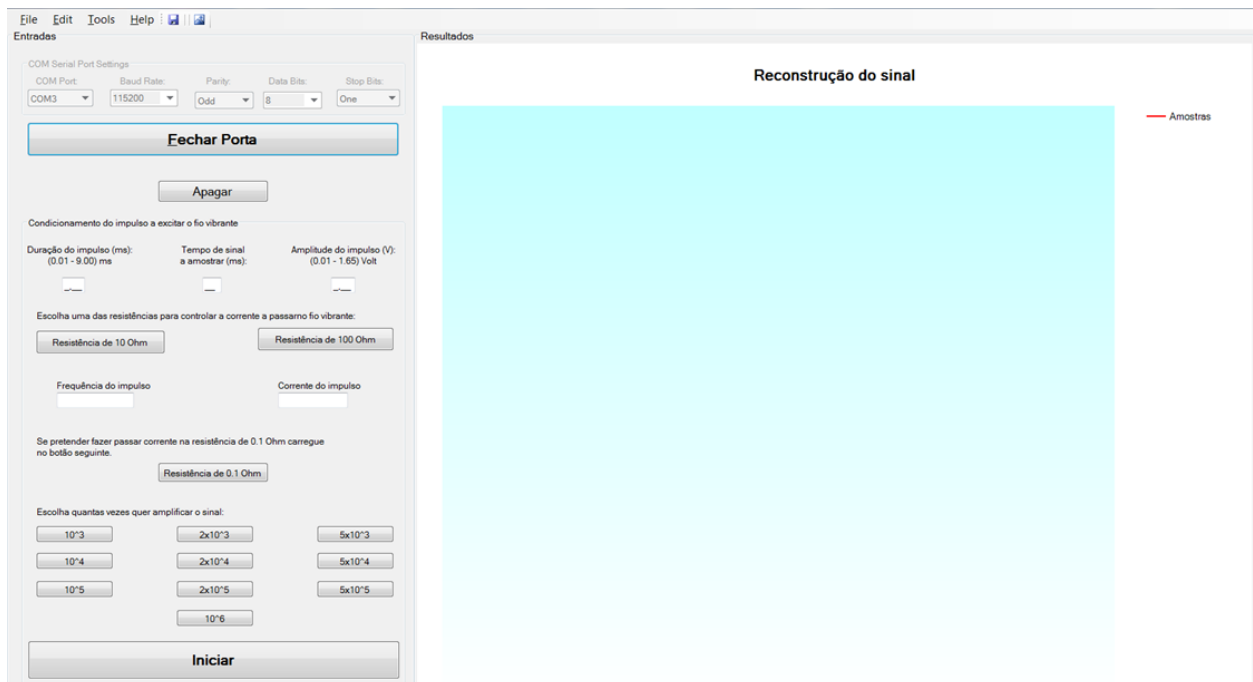
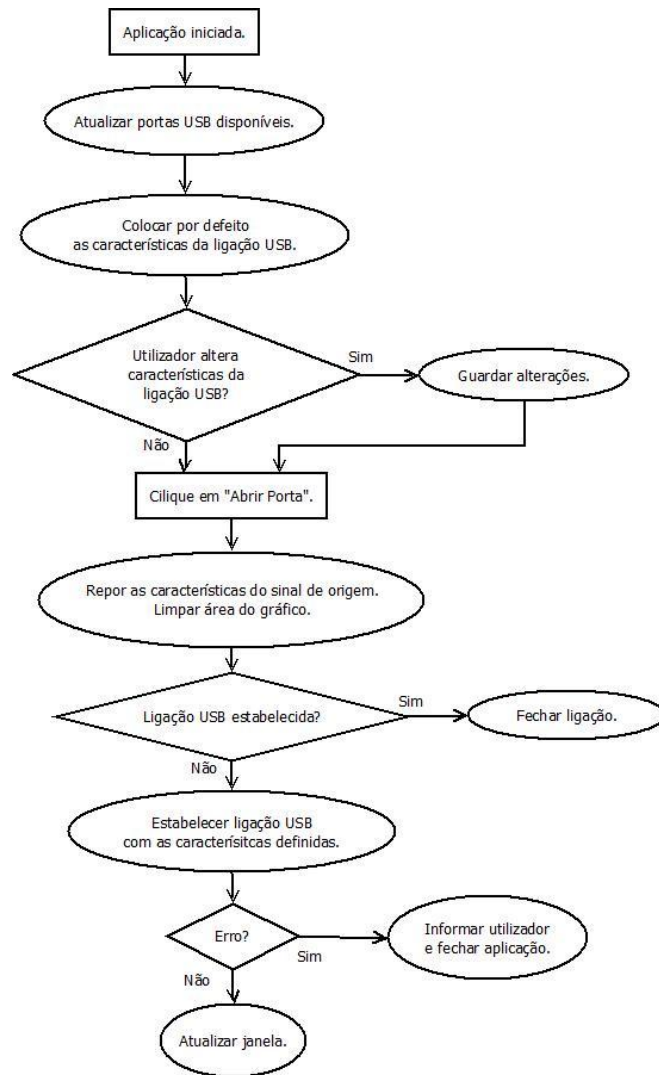


Figura 3.3 - Comunicação estabelecida.

O fluxograma seguinte resume o processo de estabelecimento da ligação USB:



Fluxograma 3.1 - Interface gráfica. Parte 1

### 3.2.2 Introdução dos parâmetros de condicionamento do impulso

Após a ligação USB estar estabelecida e a janela atualizada, o utilizador pode introduzir as características do impulso analógico desejado, podendo escolher a ordem com que introduz os valores. O utilizador tem que introduzir (Figura 3.4):

- O valor da duração do impulso compreendido entre 0.01ms e 9.00ms.
- A amplitude do impulso compreendida entre 0.01V e 1.65V.
- O tempo de sinal a amostrar.
- Optar se o impulso passa por uma resistência de 10Ω ou de 100Ω de modo a definir a corrente a estimular o fio vibrante.

- Optar se o impulso passa por uma resistência e 0.1Ω (para trabalhos futuros).
- Escolher o ganho de amplificação.

Condiçãoamento do impulso a excitar o fio vibrante

Duração do impulso (ms): (0.01 - 9.00) ms

Tempo de sinal a amostrar (ms):

Amplitude do impulso (V): (0.01 - 1.65) Volt

Escolha uma das resistências para controlar a corrente a passamo fio vibrante:

Resistência de 10 Ohm

Resistência de 100 Ohm

Frequência do impulso

Corrente do impulso

Se pretender fazer passar corrente na resistência de 0.1 Ohm carregue no botão seguinte.

Resistência de 0.1 Ohm

Escolha quantas vezes quer amplificar o sinal:

10<sup>3</sup> 2x10<sup>3</sup> 5x10<sup>3</sup>

10<sup>4</sup> 2x10<sup>4</sup> 5x10<sup>4</sup>

10<sup>5</sup> 2x10<sup>5</sup> 5x10<sup>5</sup>

10<sup>6</sup>

Iniciar

Figura 3.4 - Inserir parâmetros de condicionamento do impulso a estimular o fio vibrante.

Os valores da duração e da amplitude são introduzidos em “*masked texto boxes*” com o formato “\_.\_”. O tempo de sinal a amostrar é introduzido em “*masked texto boxes*” com o formato “\_”. Todas as restantes opções são introduzidas clicando nos respetivos botões. Para saber que se carregou em determinado botão, definiu-se uma variável global para cada botão e quando se carrega num botão, a aplicação corre um método que altera o valor dessa variável global.

Para se saber se se clicou no botão da “Resistência de 10 Ohm” ou na “Resistência de 100 Ohm” ou em nenhuma, inicializou-se uma variável com valor “00” e se se clicar no botão da “Resistência de 10 Ohm” atualiza-se o valor da variável para “01”, se se clicar no botão da “Resistência de 100 Ohm” atualiza-se o valor da variável para “10”. Deste modo, sabe-se que se o valor desta variável for “00” significa que o utilizador não clicou em nenhuma das opções.

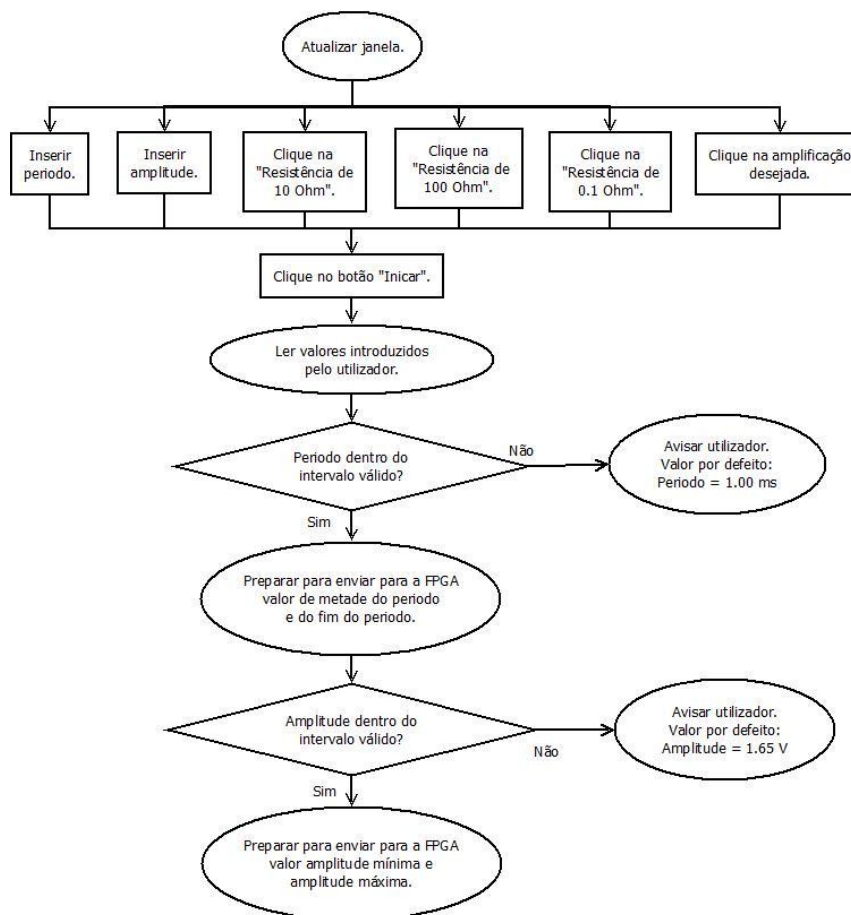
Para se saber se se clicou no botão da “Resistência de 0.1 Ohm”, define-se uma variável inicializada a “00” e se se clicar neste botão, o seu valor é atualizado para “11”.

Dispõe-se de três andares de amplificação e pretende-se amplificações de 1, 2, 5, 10, 20, 50, 100, 200, 500 e 1000 vezes (explicada na secção 5.2.2). Para tal, utilizou-se no primeiro e segundo andar de amplificação amplificações de 1 e 10 vezes e no terceiro andar de amplificação amplificações de 1, 2, 5 e 10 vezes. Deste modo é possível com oito valores diferentes de amplificação obter as dez opções possíveis. As dez opções são as seguintes:

<ul style="list-style-type: none"> <li>• <math>1 \times 1 \times 1 = 1</math></li> <li>• <math>1 \times 1 \times 2 = 2</math></li> <li>• <math>1 \times 1 \times 5 = 5</math></li> <li>• <math>1 \times 10 \times 1 = 10</math></li> <li>• <math>1 \times 10 \times 2 = 20</math></li> </ul>	<ul style="list-style-type: none"> <li>• <math>1 \times 10 \times 5 = 50</math></li> <li>• <math>10 \times 10 \times 1 = 100</math></li> <li>• <math>10 \times 10 \times 2 = 200</math></li> <li>• <math>10 \times 10 \times 5 = 500</math></li> <li>• <math>10 \times 10 \times 10 = 1000</math></li> </ul>
--	--

Tabela 3.1 - Esquema de amplificação.

Para codificar a opção ao clicar em cada botão, definiu-se que no primeiro e no segundo andar de amplificação, amplificar uma vez é codificado com “0” e amplificar dez vezes é codificado com “1”. No terceiro andar de amplificação, amplificar uma vez é codificado com “00”, amplificar duas vezes é codificado com “01”, amplificar cinco vezes é codificado com “10” e amplificar dez vezes é codificado com “11”. Por exemplo, se clicar no botão “ $5 \times 10^4$ ” a amplificação escolhida é  $1 \times 10 \times 5$ , ou seja, “0110”.

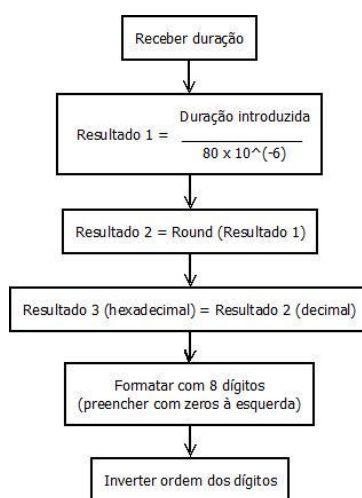


Fluxograma 3.2 - Interface gráfica. Parte 2.

Quando o utilizador clicar no botão “Iniciar” a aplicação atualiza as variáveis que guardam os valores introduzidos e verifica se a duração introduzida pertence ao intervalo válido. Se não pertencer, informa o utilizador e coloca por defeito uma duração de 1.00ms. Caso o valor pertença ao intervalo válido, faz-se a conversão do valor da duração introduzida para número de ciclos de relógio a que a FPGA comanda o DAC produzir sinais analógicos.

A comunicação com a FPGA é efetuada enviando e recebendo os dados em hexadecimal e com a ordem invertida. Em hexadecimal porque foi definido que os valores que a FPGA recebe e envia são em hexadecimal. Com a ordem invertida porque os dados são enviados para uma memória *Random Access Memory* (RAM) implementada na FPGA do endereço mais baixo para o mais alto.

A FPGA comanda o DAC para este produzir sinais analógicos de 80ns em 80ns (ver secção 4.9.2). Para isso, divide-se a duração introduzida pelo período de produção de sinais analógicos. Arredonda-se este valor, converte-se para hexadecimal, formata-se com tamanho fixo de 8 dígitos (preenchendo com zeros à esquerda, para não alterar o valor) e inverte-se a ordem dos dígitos. Assim, estes oito dígitos hexadecimais estão prontos para enviar para a FPGA (Fluxograma 3.3).



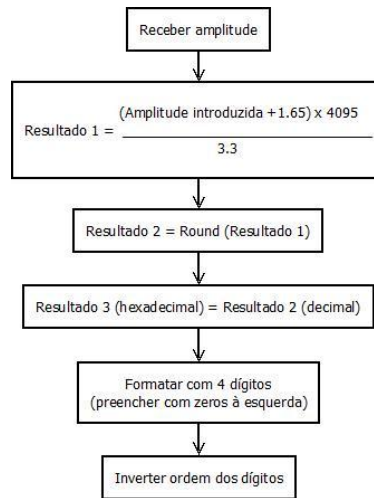
Fluxograma 3.3 - Formatação da duração do impulso a excitar o fio vibrante.

De seguida efetua-se o mesmo raciocínio do Fluxograma 3.3 mas com metade da duração introduzida (para indicar onde é metade da duração do impulso analógico a produzir). No Anexo A apresenta-se um exemplo destes cálculos.

Posteriormente a aplicação verifica se a amplitude introduzida pertence ao intervalo válido. Se não pertencer, informa o utilizador e coloca uma amplitude eficaz por defeito de 1.65V. Caso pertença ao intervalo válido, a aplicação determina qual é a amplitude máxima e mínima do impulso produzido (uma vez que se pretende produzir uma onda quadrada centrada em 1.65V, pois a FPGA produz tensões entre 0V e 3.3V, logo a referência é 1.65V). Para saber qual é a amplitude máxima soma-se 1.65V ao valor introduzido. Para saber a amplitude mínima, subtrai-se a amplitude introduzida a 1.65V e segue-se o mesmo raciocínio.

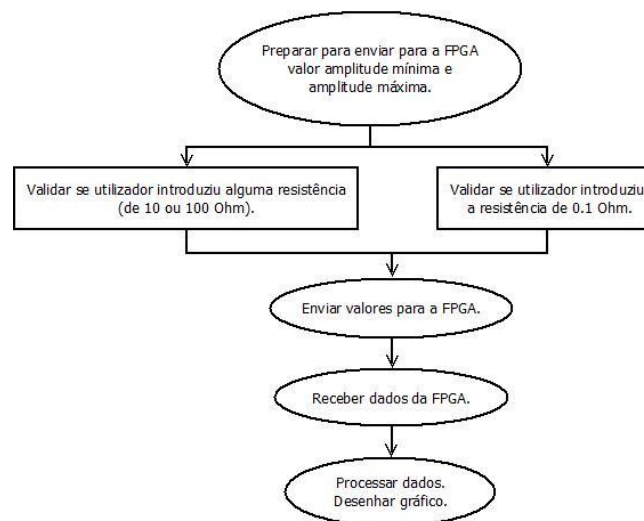
Como o DAC utilizado (PmodDA2) tem 12 bits de precisão ( $2^{12} = 4096$  níveis de conversão, com o mínimo "000000000000" = 0V e o máximo "111111111111" = 3.3V) para se fazer a conversão do valor introduzido para a escala do PmodDA2, faz-se:  $\frac{\text{Amplitude} \times 4095}{3.3}$ .

Arredonda-se este valor, converte-se para hexadecimal, formata-se com tamanho fixo de 4 dígitos (preenchendo com zeros à esquerda) e inverte-se a ordem dos dígitos (Fluxograma 3.4). No Anexo B apresenta-se um exemplo destes cálculos.



Fluxograma 3.4 - Formatação da amplitude do impulso a excitar o fio vibrante.

Optou-se por formatar a amplitude com tamanho fixo de 4 dígitos hexadecimais, porque o valor máximo que é possível é  $2^{12} = "111111111111" = 4095_{(10)} = FFF_{(16)}$  que ocupa 3 dígitos no máximo. No entanto, utiliza-se 4 dígitos e não 3 porque a RAM implementada na FPGA tem 32 bits por endereço, pelo que 4 hexadecimais ocupam 16 bits de um endereço, ou seja, metade de um endereço. Tendo em conta que se está a enviar o valor máximo da amplitude do impulso analógico a produzir e ainda falta enviar o valor mínimo do mesmo, como ele também está codificado em 4 hexadecimais que ocupam 16 bits, reserva-se um endereço de memória para a amplitude, sendo que a parte baixa do endereço guarda a amplitude máxima e a parte alta guarda a amplitude mínima.



Fluxograma 3.5 - Interface gráfica. Parte 3.

### 3.2.3 Validação e envio dos parâmetros introduzidos

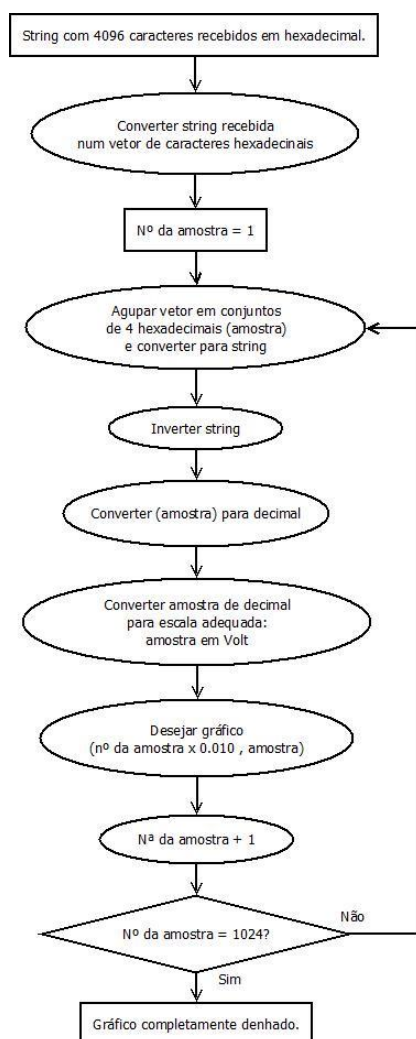
Para se validar se o utilizador introduziu alguma das resistências de 10 ou 100 Ohm, verifica-se o valor da variável (Fluxograma 3.5). Se este for “00” é porque não introduziu. Neste caso informa-se o utilizador e coloca-se por defeito a “Resistência de 10 Ohm”. Para se validar se o utilizador introduziu ou não a resistência de 0.1Ohm, verifica-se o valor da variável. Se este for “00” é porque não introduziu. Verifica-se ainda a amplificação introduzida. Depois de todas as validações serem efetuadas envia-se para a FPGA, os valores formatados pela seguinte ordem:

1. Número de ciclos de relógio do DAC para indicar metade da duração do impulso analógico.
2. Número de ciclos de relógio do DAC para indicar a duração total do impulso.
3. Amplitude mínima do impulso.
4. Amplitude máxima do impulso.
5. Resistência escolhida (10 ou 100 Ohm).
6. Se se escolheu ou não a resistência de 0.1 Ohm.
7. Valor da amplificação desejada.
8. Tempo de sinal a amostrar.
9. Caractere “L”, que funciona como um caractere especial para indicar que todos os dados já foram enviados. Quando a FPGA receber este caractere, sabe que já recebeu todos os dados e inicia o seu processamento.

Após a FPGA efetuar o processamento (secção 4.5), esta envia os resultados desse processamento para o computador novamente pela ligação USB já estabelecida. Na interface gráfica há uma *thread* que é lançada quando houver dados para receber. Nessa *thread* fez-se um atraso para dar tempo que todos os dados sejam recebidos. Após receber todos os dados, que são 4096 caracteres hexadecimais, essa *thread* invoca um método para tratar os dados recebidos.

As amostras são recebidas numa *string* com 4096 caracteres hexadecimais. São 4096 caracteres hexadecimais porque foram obtidas 1024 amostras e cada amostra está codificada com 4 hexadecimais, ou seja, 16 bits cada. Apesar do PmodAD1 codificar cada amostra com 12 bits, adicionou-se “0000” à esquerda para que cada amostra ficasse com 16 bits, sem alterar o valor da mesma. Como a RAM implementada na FPGA tem 32 bits por endereço e 512 endereços, cada endereço guarda 2 amostras, guardando assim um total de 1024 amostras.

### 3.2.4 Receção e tratamento das amostras



Fluxograma 3.6 - Funcionamento do método que trata as amostras recebidas.

O método que processa as amostras vai então converter a *string* recebida num vetor de caracteres hexadecimais e depois agrupa esse vetor em conjuntos de 4 caracteres que volta a converter em *string*. Como a FPGA envia os dados do endereço mais baixo para o mais alto e do bit menos significativo para o mais significativo, é necessário inverter a ordem da *string* que guarda os 4 hexadecimais (uma amostra). De seguida converte essa amostra de 4 hexadecimais para decimal e converte a amostra para a escala adequada em Volt, fazendo:

$$Amostra\_Volt = \frac{Amostra\_decimal \times 3.3}{4095} \quad (3.1)$$

À medida que se vão obtendo amostras vai-se desejando o gráfico, sendo um gráfico em que no eixo das abcissas se coloca o *tempo(ms)* e no eixo das ordenadas se coloca o valor da amostra em *Volt(V)*.

# Capítulo 4

## 4 Projeto, simulação e implementação na FPGA

### 4.1 FPGA

A FPGA é um circuito integrado que pode ser reconfigurado por *software* para implementar sistemas digitais como decodificadores, processadores, controladores, etc. A estrutura básica de uma FPGA consiste numa matriz de três tipos de elementos programáveis. Estes elementos são os blocos lógicos, as células de entrada/saída e a zona de interligações.

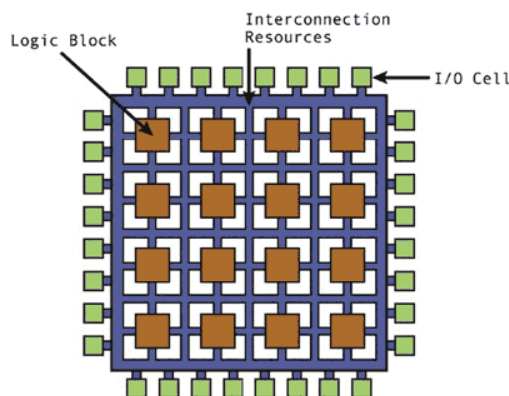


Figura 4.1 - Modelo conceptual de uma FPGA [27].

Os **blocos lógicos**, conhecidos por *Configuration Logic Blocks (CLBs)* são tipicamente compostos por um *flip-flop D* e uma *Look-Up Table (LUT)*, que consiste numa RAM de 16 bits.

As **células de entrada/saída**, conhecidas por *Input/Output Blocks (IOBs)*, são *buffers* bidirecionais com saída em alta impedância que permitem configurar um pino para funcionar como entrada ou como saída.

A **zona de interligações** faz a ligação entre os CLBs e os IOBs e consiste num conjunto de ligações verticais e horizontais.

Para implementar uma função lógica, armazena-se na LUT a tabela de verdade da mesma. Para utilizar essa função lógica utiliza-se a LUT em modo de leitura.

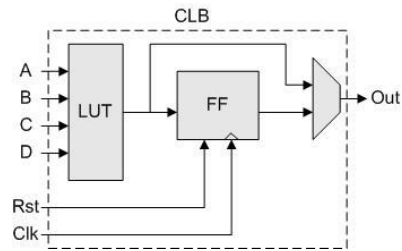


Figura 4.2 - Estrutura básica do CLB [28].

Além destas estruturas básicas, a FPGA disponibiliza outras estruturas de *hardware* para auxiliar a implementação de funções lógicas definidas pelo projetista. Estes recursos estão geralmente dispostos em colunas ao longo da FPGA. Como se pode observar na Figura 4.3 os recursos podem ser blocos de RAM de 18 Kbits (*Block RAM*), multiplicadores dedicados (*Multipliers*) e *Digital Clock Manager* (DCM) que pode ser utilizado para multiplicar ou dividir a frequência do relógio de entrada no circuito.

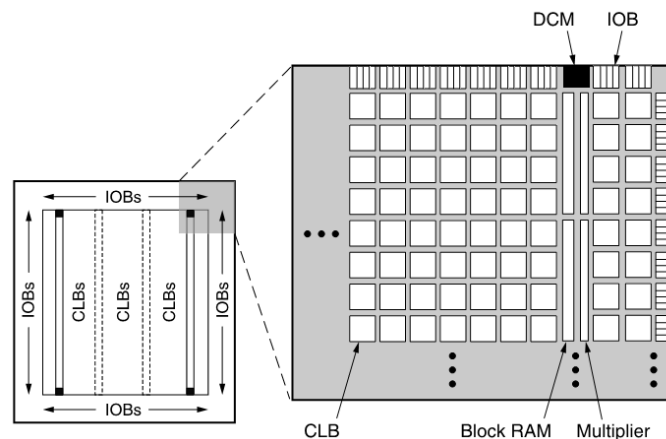


Figura 4.3 - Arquitetura da família Spartan-3 [29].

## 4.2 FPGA utilizada

No mercado existem uma série de famílias de FPGAs. Uma dessas famílias é a *Spartan-3*. A *Spartan-3* faculta uma série de FPGAs com características e potencialidades distintas. A FPGA adotada para realização do trabalho foi a *XC3S1000* [29] pelo que a arquitetura que se falará de agora em diante será relativa à FPGA utilizada.

Para realizar a interface entre a interface gráfica, a produção do impulso analógico e a aquisição da resposta do fio vibrante utilizou-se o *kit* de desenvolvimento *Spartan-3 Starter Kit Board* [30]. Escolheu-se utilizar este *kit* de desenvolvimento devido ao seu preço de mercado (na ordem dos 150€) e capacidades suficientes para realização do trabalho pretendido e ainda possibilidade de expansão ou melhoria futura do trabalho realizado. Algumas das características relevantes desta placa de desenvolvimento são: oscilador de 50MHz, 1 Mbyte (256Kb x 32) de *Static Random-Access Memory (SRAM)* com 10ns de tempo de acesso e três “*expansion connectors*” de 40 pinos para Entrada/Saída (E/S), entre outras.



Figura 4.4 - *Spartan-3 Board* [30].

### 4.3 Linguagem de descrição de *hardware* (HDL)

As linguagens de descrição de *hardware* permitem descrever projetos independentes da tecnologia e das ferramentas utilizadas, um nível mais alto de abstração, redução do tempo de desenvolvimento de um projeto e facilidade de atualização dos projetos. Deste modo, as HDLs juntamente com as FPGAs vieram assim revolucionar o desenvolvimento de sistemas digitais.

Atualmente as HDLs mais utilizadas são o VHDL e o Verilog. A linguagem utilizada na realização deste trabalho foi VHDL (de VHSIC – *Very High Speed Integrated Circuit*).

Existem no mercado vários programas que suportam este tipo de linguagens. A ferramenta utilizada na realização deste trabalho foi a “*Xilinx ISE Design Suite 12.4*”. Esta ferramenta permite descrever o circuito desejado utilizando VHDL, fazer a implementação, simulação e configurar o dispositivo alvo que é a FPGA utilizada.

## 4.4 Vantagens e desvantagens das FPGAs

Para aplicações em que um *chip* personalizado é essencial, considera-se utilizar FPGAs para a criação de protótipos de componentes que serão *Application Specific Integrated Circuits* (ASIC). No entanto não há problema de se distribuir FPGAs em produtos finais. Esta decisão requer a análise de custo, uma vez que o desenvolvimento de um ASIC é um processo bastante caro e inflexível, mas que gera componentes de custo reduzido quando produzidos em larga escala e a FPGA possui baixo custo de prototipagem e alta flexibilidade, no entanto, o preço dos produtos finais que utilizam FPGA são pouco competitivos se comparados a ASICs com alta escala de produção.

A FPGA tem a vantagem de poder ser configurada através de um cabo (JTAG ou USB) e desempenhar determinadas funções, como um chip que acabou de ser ligado. A configuração da FPGA é volátil, ou seja, é perdida sempre que a FPGA é desligada da alimentação. Recorrer a FPGAs permite baixar o custo do projeto, testar e fazer modificações quase sem custos, de modo idêntico ao *software*. Embora as FPGAs ofereçam muitas vantagens, existem algumas desvantagens, das quais se destacam o facto de as FPGAs poderem ser mais lentas do que o ASIC equivalente e mais caras [31].

## 4.5 Visão global

Pretende-se que a FPGA seja capaz de receber parâmetros provenientes de um cabo USB ligado ao computador onde corre a interface gráfica e que os guarde numa memória. Posteriormente pretende-se que a FPGA controle um DAC que produza um impulso analógico com as características definidas pelo utilizador e que correspondem aos dados gravados na memória. A FPGA deve também controlar uns relés que pertencem ao bloco do condicionamento de sinal e esse controlo deve ser feito em função das opções do utilizador, que também estão gravados na memória. De seguida, a FPGA deve controlar um ADC para adquirir amostras da resposta do fio vibrante e deve guardar essas amostras noutra memória. Por fim, a FPGA deve enviar as amostras que estão na memória de saída de volta para a interface gráfica (no computador) através do cabo USB. Todo este processo deve ser síncrono.

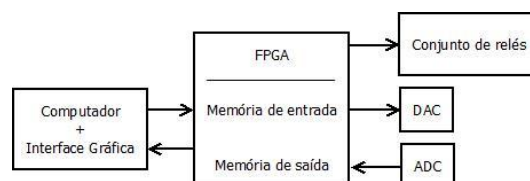


Figura 4.5 - Interligação da FPGA no sistema.

Na FPGA foram implementados circuitos com as seguintes funções (ver Figura 4.6):

**Interface\_serie:** circuito responsável pela comunicação com o computador.

**UARTcomponent:** circuito responsável por receber os dados da porta série bit a bit e agrupá-los em bytes a enviar para a *Decode\_hexa\_chars*. Circuito responsável por receber as amostras agrupadas por bytes do *Encode\_hexa\_chars* e enviá-los para a porta série bit a bit.

**Decode\_hexa\_chars:** circuito responsável por converter os caracteres ASCII (8 bits) provenientes da UART para a respetiva representação hexadecimal (4 bits). Quando receber o caractere “L” gera um sinal que indica que já recebeu tudo.

**Memoria\_entrada\_na\_FPGA:** RAM de 18432 bits (18 kbits) de duplo porto. Porto *A* com 4096 endereços e 4 bits por endereço. Porto *B* com 512 endereços e 36 bits por endereço. No porto *A* são guardados os valores hexadecimais provenientes do *Decode\_hexa\_chars*. Do porto *B* são enviados os dados para o *Circuito3*.

**Circuito3:** recebe os valores da *Memória\_entrada\_na\_FPGA*, processa-os e envia-os para o *DA2\_controller* e para as saídas digitais. Recebe as amostras do *AD1\_controller* e envia-as para a *Memória\_saida\_da\_FPGA*.

**DA2\_controller:** controla a *PmodDA2* para este produzir o impulso de tensão.

**AD1\_controller:** controla o *PmodAD1* para este amostrar a resposta do fio vibrante.

**Memoria\_saida\_da\_FPGA:** RAM de 18432 bits de duplo porto. Porto *A* com 4096 endereços e 4 bits por endereço. Porto *B* com 512 endereços e 36 bits por endereço. No porto *B* são guardadas as amostras digitalizadas. Do porto *A* são enviadas as amostras para o *Encode\_hexa\_chars*.

**Encode\_hexa\_chars:** circuito responsável por converter as amostras da sua representação em hexadecimal para caracteres ASCII.

**Interface\_placa:** circuito responsável por permitir fazer *debug*.

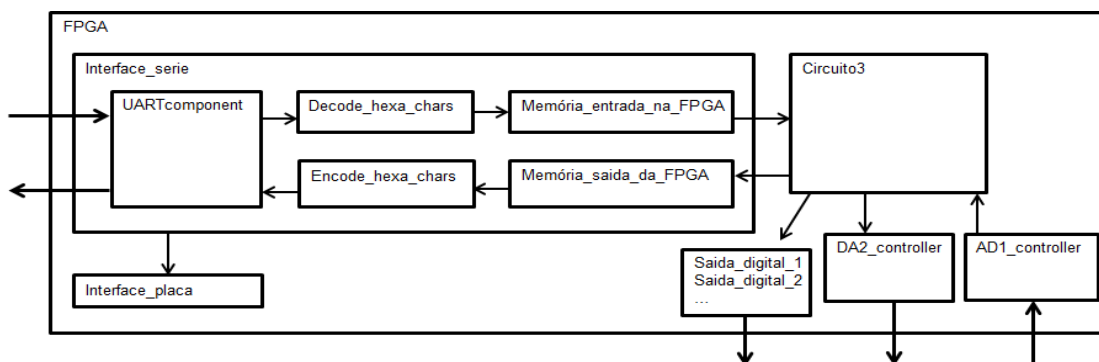


Figura 4.6 - Sistema global a implementar na FPGA.

## 4.6 Comunicação entra a FPGA e o computador

Tendo em conta que a FPGA utilizada não possui ligação USB disponível, efetuou-se uma ligação utilizando um cabo Série-USB, como o da Figura 4.7 para estabelecer a comunicação entre o computador e a FPGA.



Figura 4.7 - Cabo Série-USB.

Para utilizar este cabo é necessário instalar um *driver* que permite que o computador reconheça o dispositivo USB como uma porta COM tradicional [32]. A interface gráfica implementada cria uma porta COM virtual que permite aceder ao dispositivo USB como se este fosse uma porta COM. Deste modo a interface gráfica envia os dados em hexadecimal e estes chegam à FPGA em ASCII. Por outro lado, quando se pretende enviar dados da FPGA para a interface gráfica, é necessário converter os dados de hexadecimal para ASCII.

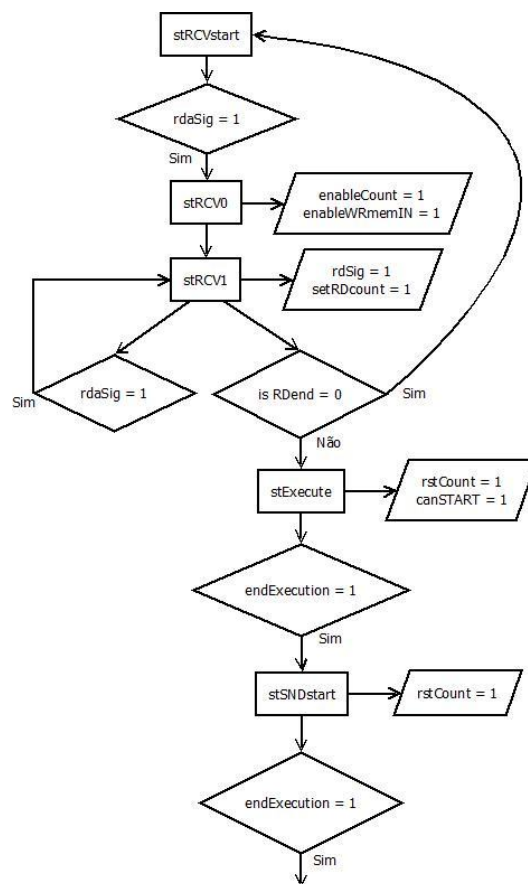
Assim, a comunicação com a FPGA é efetuada enviando e recebendo os valores em hexadecimal e com a ordem invertida. Em hexadecimal porque foi implementado um circuito na FPGA de modo a que os valores que a FPGA recebe e envia sejam em hexadecimal. Com a ordem invertida porque os dados são enviados/recebidos para/de uma memória RAM implementada na FPGA, do endereço mais baixo para o mais alto e do bit menos significativo para o bit mais significativo.

### 4.6.1 Circuito *interface\_serie*

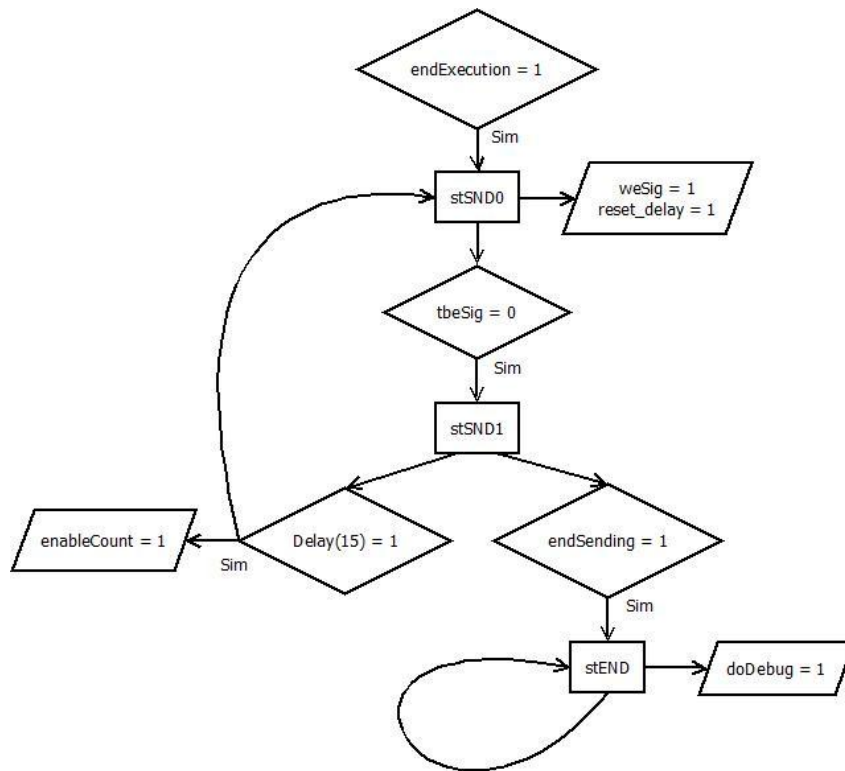
O circuito responsável pela receção/envio dos dados do/para computador para/da FPGA é o circuito *interface\_serie*. Neste circuito foi implementada uma máquina de estados que controla a receção dos dados, dá a ordem para iniciar o processamento (após todos os dados terem sido recebidos) e controla o envio dos dados.

Como se pode observar nos Fluxograma 4.1 e 4.2, os estados que coordenam a receção da informação são os estados “*stRCVstart*”, “*stRCV0*” e “*stRCV1*”. O estado que coordena o início do processamento nos outros componentes é o “*stExecute*”. Os estados que coordenam o envio das amostras são os estados “*stSND0*”, “*stSND1*” e “*stEND*”.

A máquina de estados é iniciada no estado *stRCVstart*, onde permanece até que seja recebido um byte (*rdaSig = 1*), após o qual a máquina de estados evolui para o estado *stRCV0*. Neste estado dá-se a ordem para escrever na memória de entrada na FPGA (*enableWRmemIN=1*) para guardar os valores provenientes do computador. Depois, a máquina de estados evolui para o estado *stRCV1*, onde se verifica se há mais dados a receber. Se ainda não terminou de receber tudo volta ao *stRCVstart*. Quando terminar de receber todos os bytes, ou seja, quando receber o “L” (*isRDend = 1*) a máquina de estados evolui para o estado *stExecute* onde é dada a ordem para os restantes componentes iniciarem o processamento (*CanStart = 1*). Quando neste estado houver a receção do sinal “endExecution = 1” a indicar que os componentes já terminaram o seu processamento, a máquina de estados evolui para o estado *stSND0* onde inicia um contador (*delay*) que impõe o ritmo de envio dos dados (*reset\_delay = 1*) e onde se informa o *buffer* de envio que está pronto a enviar informação. Quando o *buffer* estiver disponível, a máquina de estados evolui para o estado *stSND1* onde se verifica se já terminou de enviar tudo (*endSending = 1*) que acontece quando tiver enviado 1024 amostras de 4 bytes cada (4096 bytes) ou, caso não tenha enviado tudo, verifica se já pode enviar o próximo byte (*delay(15) = 1*). Se puder enviar o próximo byte, a máquina de estados volta ao estado *stSND0* e atualiza o contador do número de bytes enviados. Se já tiver enviado tudo a máquina de estados evolui para o estado *stEND* onde permanece e indica que se pode fazer *debug* (*doDebug = 1*).



Fluxograma 4.1 - Máquina de estados do circuito *interface\_serie*. Parte 1.

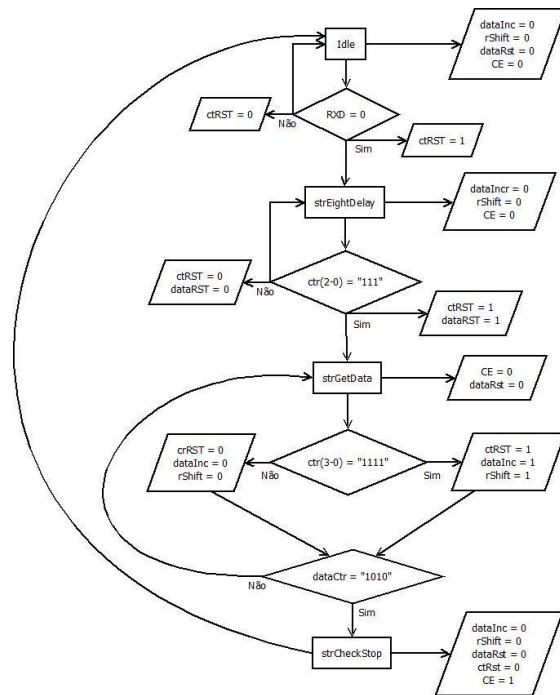


Fluxograma 4.2 - Máquina de estados do circuito *interface\_serie*. Parte 2.

#### 4.6.2 Circuito UART

O componente UART é um dos subcomponentes do componente *interface\_serie* e inclui um recetor e um transmissor. O recetor desloca os dados bit a bit e reagrupa-os do modo desejado. O transmissor é um *shift register* que carrega dados em paralelo e os desloca para a saída bit a bit. O UART implementado tem três máquinas de estados, sendo a *rstate* responsável pela recepção, a *tstate* responsável pelo envio e a *TBEstate* responsável por sincronização.

A máquina de estados de recepção (apresentada no Fluxograma 4.3) é iniciada no estado *Idle*. A linha série está a '1' quando está no estado *Idle*. A recepção começa com o *Start bit*, que é um '0' ( $RXD = 0$ ). De seguida são recebidos 10 bits no estado *strGetData*, dos quais 8 bits são dados, 1 bit de paridade e 1 "stop bit". Foi implementada paridade ímpar (*odd*) que coloca o bit de paridade a '0' quando o número de dados a '1' é ímpar. O *stop bit* é sempre '1'. Quando se receber o *stop bit* significa que já se recebeu 1 byte. Nesta fase processa-se esse byte e a máquina de estados evolui para o estado *Idle* para receber outro byte.



Fluxograma 4.3 - Máquina de estados do circuito UART para a receção.

A taxa de amostragem utilizada é 16 vezes o valor do *baud rate*, isto significa que um bit é amostrado 16 vezes. O esquema de amostragem funciona da seguinte maneira:

1. Esperar até que o sinal de entrada seja um '0' (*start bit*) e aí iniciar o contador de amostragem.
2. Quando o contador de amostragem tiver o valor 7, o sinal de entrada atinge o ponto médio do bit de início. Reinicia-se o contador de amostragem e o contador do número de bits recebidos.
3. Quando o contador de amostragem atingir o valor 15, recebe-se o primeiro bit de dados (*d0*). Guarda-se esse bit num registo, reinicia o contador de amostragem e incrementa o contador do número de bits recebidos.
4. Repetir o ponto 3 mais sete vezes para receber os 8 bits de dados.
5. Repetir o ponto 3 mais uma vez para receber o bit de paridade.
6. Repetir o ponto 3 mais uma vez para receber o *stop bit*.



Figura 4.8 - Receção de um byte.

O registo onde são guardados os dez bits recebidos é um registo que funciona de acordo com a Figura 4.9:



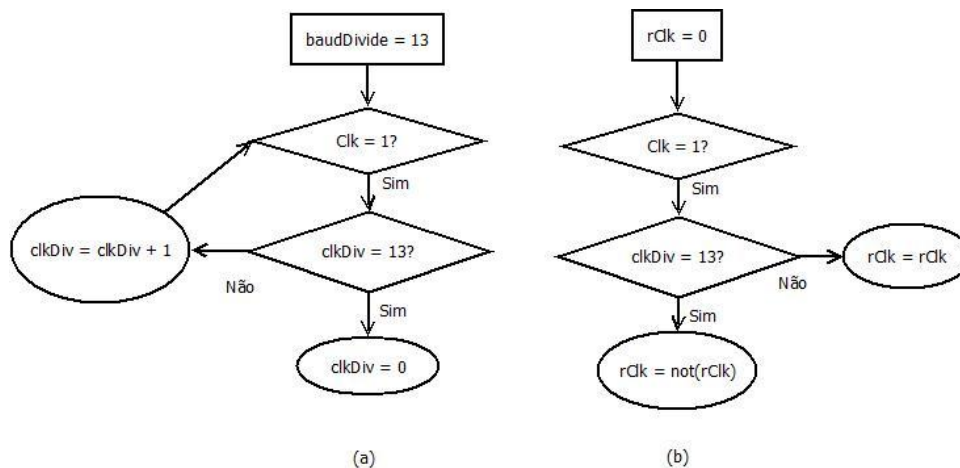
segundo, isto é, 16 vezes o valor do *baud rate*. Como é o relógio da FPGA que comanda a amostragem e tem uma frequência de 50 MHz, é necessário um contador que conte até:

$$\frac{50 \times 10^{-6}}{115200 \times 16} = 27.1 \approx 28 \quad (4.1)$$

Esse contador tem que controlar o relógio *rClk* com período equivalente a 28 ciclos de relógio da FPGA. Pode-se então dizer que o período do relógio de recepção é  $rClk = 28 \times 20 \times 10^{-6} = 0.56 \mu s$ . Isto significa que se demora  $0.56 \mu s$  a receber 1 bit, logo para receber os 10 bits (8 bits de dados, 1 bit de paridade e 1 bit de stop) demora-se  $5.6 \mu s$ . Como são recebidos 17 bytes, a recepção demora  $17 \times 5.6 \times 10^{-6} = 95.2 \mu s$ .

O valor do relógio *rClk* é atualizado ('0' ou '1') de cada vez que  $clkDiv = 14$ , ou seja, metade do período. Como o contador inicia a contagem em zero, isto acontece ao ciclo 13. Foi utilizada a variável  $baudDivide = 13$  para o caso de se pretender alterar o valor do *baud rate*, apenas necessitar de repetir o cálculo (4.1) para outro valor de  $baudDivide$  e atualizar o valor de  $baudDivide$ .

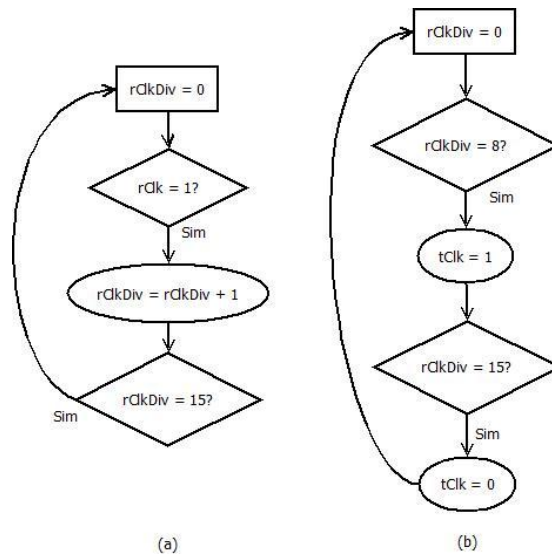
O Fluxograma 4.4 espelha este raciocínio:



Fluxograma 4.4 - Definição do relógio que comanda a recepção *rClk*. (a) – Atualização do *clkDiv*. (b) – Definição do *rClk*.

O relógio que controla o envio *tClk* é atualizado ('0' ou '1') de cada vez que  $rClkDiv=15$ , ou seja, *tClk* tem 16 vezes o período do *rClk*. Pode-se então dizer que o período do relógio de envio é  $tClk = 0.56 \mu s \times 16 = 8.96 \mu s$ .

O Fluxograma 4.5 espelha este raciocínio:

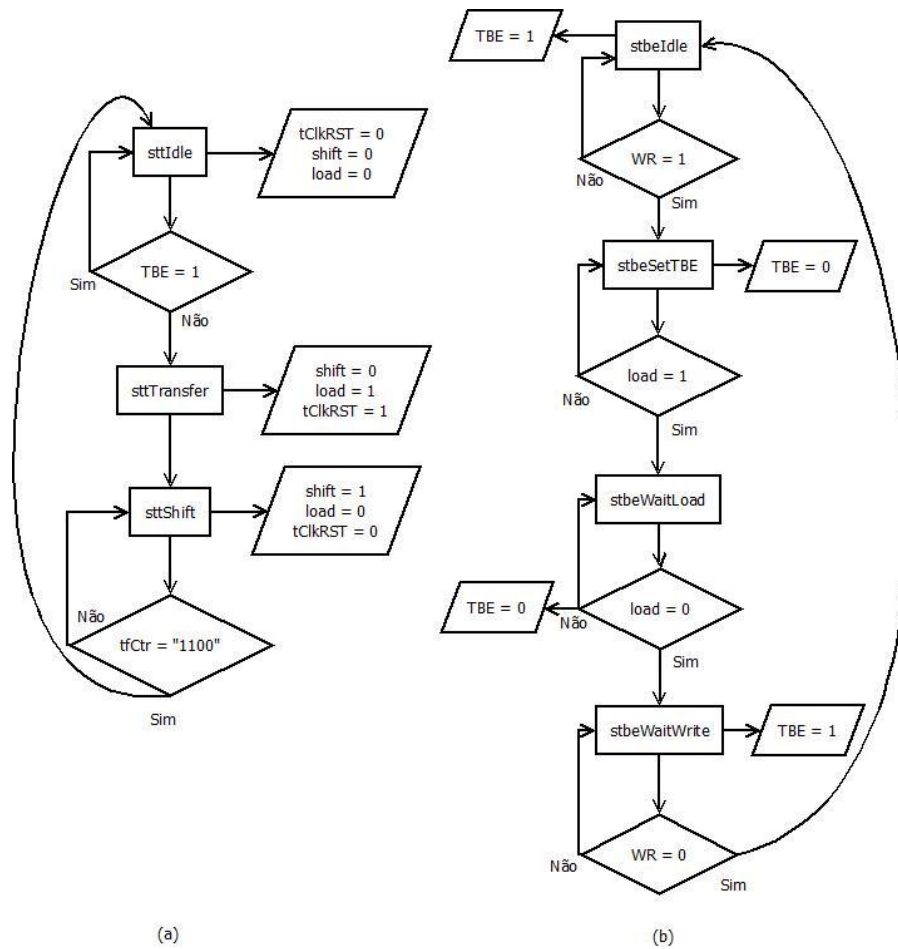


Fluxograma 4.5 - Definição do relógio que comanda o envio *tClk*. (a) – Atualização do *rClkDiv*. (b) – Definição do *tClk*.

A máquina de estados responsável pelo envio *tstate* e a máquina de estados responsável pela sincronização *TBEstate* estão ligadas para funcionarem de modo síncrono, tendo o seguinte comportamento:

- Tanto a máquina de estados *tstate* como a *TBEstate* são iniciadas no estado *Idle*. *TBEstate* coloca um sinal *TBE = '1'* para indicar que o cabo está livre.
- Quando *TBEstate* receber o sinal para iniciar o envio de dados (*WR = '1'*) evolui para o estado *stbeSetTBE* e *tstate* evolui para o estado *sttTransfere*. *WR* é produzido no estado *stSND0* do circuito *interface-serie*.
- *TBEstate* atualiza-se o sinal *TBE = '0'* para reservar o cabo para enviar dados e permitir que no circuito *interface-serie* se evolua para o estado *stSND1*.
- *tstate* dá a ordem para carregar o registo com os dados a enviar (*load = '1'*).
- Quando se iniciar o carregamento dos dados *TBEstate* evolui para o estado *stbeWaitLoad* onde se verifica se o registo já está carregado e *tstate* evolui para o estado *sttShift* onde se indica que o registo já está carregado e pode-se iniciar o deslocamento dos dados bit a bit (*shift = '1'*), até se carregarem 11 bits.
- Quando se carregarem os 11 bits, *tstate* volta para o estado inicial para recomeçar o processo e enviar a restante informação.
- Paralelamente, *TBEstate* verifica que os dados já foram carregados e evolui para o estado *stbeWaitWrite* onde atualiza o sinal *TBE = '1'* para indicar que o cabo está livre e verifica se ainda há ordem para enviar dados (*WR = '1'*).
- Após enviar um conjunto de informação (11 bits) o circuito *interface-serie* atualiza a ordem de envio de dados para não permitir enviar mais dados (*WR='0'*) e no estado *stSND1* verifica-se se foi enviado o último byte. Se não

tiver sido o último byte espera-se um determinado tempo identificado por  $delay(15)$  que corresponde a  $2^{15} \times 20 \times 10^{-9} = 655.36 \mu s$  para a máquina de estados de *interface-serie* se evolua para o estado *stSND0* e permitir enviar outro conjunto de informação ( $WR = '1'$ ).



Fluxograma 4.6 - Máquina de estados do circuito UART para o envio e sincronismo. (a) – Envio. (b) – TBE.

Pretende-se enviar 1024 amostras de 16 bit cada, ou seja, 16384 bits que estão gravadas em hexadecimal na memória de saída da FPGA. Mas essa informação tem que ser codificada em ASCII no circuito *encode\_hexa\_chars*. Resumindo, tem que se enviar  $1024(amostras) \times 16(bit\ por\ amostra) \times 2 = 32768 = 4096(bytes)$ .

Para controlar a quantidade de bytes enviados e indicar qual o último byte a ser enviado, efetuou-se um contador que vai contando o número de bytes enviados e quando este for igual a  $2^{12} = "111111111111" = 4096$  gera o sinal a indicar que já enviou todos os dados ( $endSending = '1'$ ).

Tendo em conta que o espaçamento temporal entre bytes enviados é 655.36µs e se pretende enviar 4096 bytes, o tempo que se demora a enviar as amostras de volta para o computador é  $4096 \times 655.36 \times 10^{-6} = 2.68s$ .

Para se definir o bit de paridade *par* faz-se XOR entre os 8 pares de bits de dados a enviar e depois o NOT do seguinte modo:

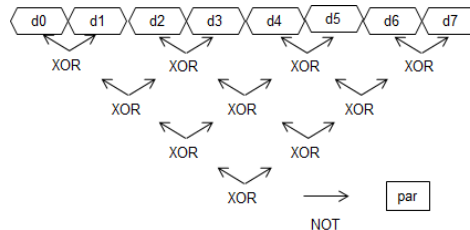


Figura 4.12 - Definição do bit de paridade *par* a enviar.

Quando no estado *tstate* é dada a ordem para carregar o registo *tfSReg* com os dados a enviar (*load = '1'*) primeiro é carregado o *stop bit* '1', seguido do bit de paridade (*par*), depois os 8 bits de dados e por fim o *start bit* '0' (Figura 4.13).

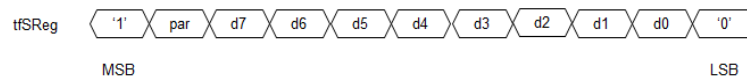


Figura 4.13 - Registo *tfSReg* carregado, pronto a enviar bit a bit.

Quando no estado *sttShift* se indica que o registo já está carregado e pode-se iniciar o deslocamento dos dados bit a bit (*shift = '1'*), este deslocamento consiste em enviar os bits em série para o cabo, começando pelo bit menos significativo, o *start bit* e deslocar os bits para a direita, preenchendo a parte mais significativa do registo com '1' (Figura 4.14).

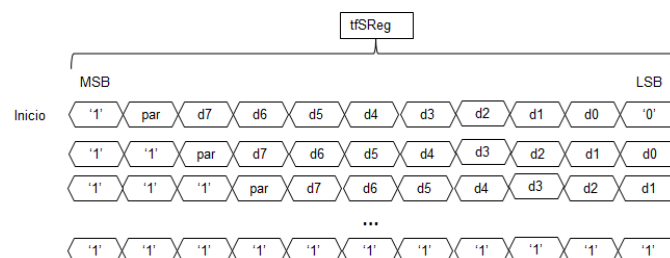


Figura 4.14 - Deslocamento dos bits no registo *tfSReg* e envio em série para o cabo.

### 4.6.3 Circuito *decode\_hexa\_chars*

O circuito *decode\_hexa\_chars* faz a conversão de um byte em código ASCII (que corresponde a um dígito da interface gráfica) para a sua representação em hexadecimal. A representação em ASCII é proveniente do circuito *UART* e a representação hexadecimal é encaminhada para a memória de entrada na FPGA. A conversão de ASCII (*din*) para hexadecimal (*dout*) é a apresentada a seguir em código VHDL:

```
entity decode_hexa_chars is
  port (din : in std_logic_vector (7 downto 0);
        dout : out std_logic_vector (3 downto 0);
        isEnd : out std_logic
        -- isHexa : out std_logic
        );
end decode_hexa_chars;

architecture Behavioral of decode_hexa_chars is
begin
  with din select
    dout <=
      X"0" when X"30",
      X"1" when X"31",
      X"2" when X"32",
      X"3" when X"33",
      X"4" when X"34",
      X"5" when X"35",
      X"6" when X"36",
      X"7" when X"37",
      X"8" when X"38",
      X"9" when X"39",
      X"A" when X"41" | X"61",
      X"B" when X"42" | X"62",
      X"C" when X"43" | X"63",
      X"D" when X"44" | X"64",
      X"E" when X"45" | X"65",
      X"F" when X"46" | X"66",
      X"0" when others;

  with din select
    isEnd <= '1' when X"4C" | X"6C", --caracteres: L ou l
              '0' when others;
end Behavioral;
```

Figura 4.15 - Código VHDL que especifica o circuito *decode\_hexa\_chars* para a conversão de ASCII para hexadecimal.

É neste circuito que se identifica a recepção do caractere *L* e se gera o sinal *isEnd* que identifica que já chegaram todos os dados e se pode iniciar o processamento.

### 4.6.4 Circuito *encode\_hexa\_chars*

O circuito *encode\_hexa\_chars* faz a conversão da representação em hexadecimal de um dígito para a sua representação em código ASCII. A representação em hexadecimal é proveniente da memória de saída da FPGA e a sua representação em código ASCII é encaminhada para o circuito *UART*. A conversão de hexadecimal (*din*) para ASCII (*dout*) é a apresentada a seguir em código VHDL:

```

entity encode_hexa_chars is
  port (din : in std_logic_vector (3 downto 0);
        dout : out std_logic_vector (7 downto 0)
        );
end encode_hexa_chars;

architecture Behavioral of encode_hexa_chars is
begin
  with din select
    dout <=
      X"30" when X"0",
      X"31" when X"1",
      X"32" when X"2",
      X"33" when X"3",
      X"34" when X"4",
      X"35" when X"5",
      X"36" when X"6",
      X"37" when X"7",
      X"38" when X"8",
      X"39" when X"9",
      X"41" when X"A",
      X"42" when X"B",
      X"43" when X"C",
      X"44" when X"D",
      X"45" when X"E",
      X"46" when X"F",
      X"58" when others;
end Behavioral;

```

Figura 4.16 - Código VHDL que especifica o circuito *encode\_hexa\_chars* para a conversão de hexadecimal para ASCII.

#### 4.7 Circuito *interface\_placa*

O circuito *interface\_placa* é um circuito que foi desenvolvido com o objetivo de possibilitar efetuar *debug*. Colocou-se a hipótese de se retirar este circuito após terminar a implementação na FPGA, mas ponderando a sua complexidade e acréscimo de degradação de performance da FPGA com a possibilidade de alterações no futuro, optou-se por se manter este circuito.

O circuito *interface\_placa* permite verificar nos *displays* de 7 segmentos a informação guardada na memória de entrada na FPGA e na memória de saída da FPGA, usando os *switches* da placa para introduzir o endereço das posições de memória a visualizar. Por omissão são visualizados os 2 bytes menos significativos da memória de entrada na FPGA para o endereço selecionado. O botão 2 permite visualizar os 2 bytes mais significativos da memória de entrada na FPGA. O botão 3 permite visualizar os 2 bytes menos significativos da memória de saída da FPGA. Os botões 2 e 3 em simultâneo permitem visualizar os 2 bytes mais significativos da memória de saída da FPGA.

Além disso, este circuito permite verificar se foi produzido o sinal que indica que todos os bytes foram enviados e pode iniciar o *debug* (*doDebug* = '1'). Também permite verificar se foi produzido o sinal que indica que todo o processamento já foi feito e dá a ordem para iniciar o envio das amostras (*execDONE*= '1'). Por fim, permite verificar se foi produzido o sinal que dá

a ordem para iniciar o processamento, após a recepção de todos os dados ( $canSTART = '1'$ ). Estes sinais são indicados nos *led 7 a 5*, respetivamente [30].

## 4.8 Circuito aquisição do sinal analógico

Para digitalizar a resposta analógica do fio vibrante foi utilizado um ADC “*PmodAD1*” [33] (Figura 4.17).



Figura 4.17 - PmodAD1 [33]

O circuito PmodAD1 converte sinais analógicos numa gama dos 0 aos 3.3 Volt e numa escala de 0 a 4095 (12 bits), produzindo até 1 milhão de amostras por segundo. Tem dois canais de conversão simultânea, cada canal com um conversor de 12 bits (AD7476A [34]) e um filtro. Pode-se utilizar os dois canais em simultâneo ou apenas um canal. Para fazer a conversão, o circuito PmodAD1 utiliza o protocolo SPI/MICROWIRE™ [35] para enviar os dados para a FPGA. O barramento série SPI pode alcançar os 20 MHz. O PmodAD1 liga-se à FPGA aos pinos de Entrada/Saída (E/S) através do *J1 Connector*. A entrada analógica liga-se ao *J2 Connector* (Figura 4.18).

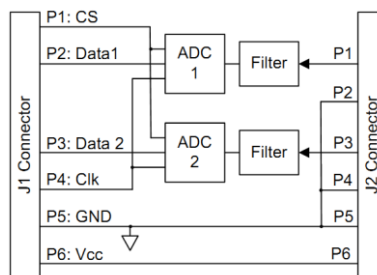


Figura 4.18 - Diagrama de blocos do circuito PmodAD1 [33].

O circuito AD7476A tem como entradas as alimentações ( $V_{DD}$  e  $GND$ ), o sinal analógico a digitalizar ( $V_{IN}$ ), o relógio do componente que determina o ritmo de conversão ( $SCLK$ ) e o sinal que seleciona qual dos canais de conversão é utilizado ( $Chip Select - CS$ ). A saída deste componente é um bit ( $SDATA$ ) que é fornecido a cada ciclo de relógio  $SCLK$

(Figura 4.19). O resultado da conversão é um conjunto de 16 bits em série, sendo que os primeiros 4 bits são zeros e os restantes 12 bits são uma amostra.

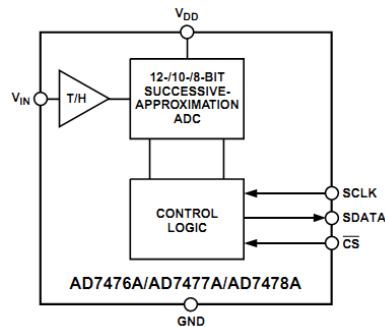


Figura 4.19 - Diagrama de blocos do circuito AD7476A [34].

#### 4.8.1 Conversão AD

O circuito AD7476A é um conversor analógico-digital por aproximações sucessivas. Esta técnica consiste numa comparação da tensão de entrada com uma tensão gerada pelo conversor DA resultante de um código binário do Registro de Aproximações Sucessivas (RAS). Quando é dado início a uma conversão ( $CS = 0$  e  $SCLK = 1$ ) o *Most Significant Bit* (MSB) do RAS é colocado a '1' (1/2 do valor do fim de escala). A tensão gerada pelo DA é comparada com a tensão de entrada. Se a tensão de entrada for superior à gerada pelo DA então o MSB é mantido a '1'. O teste ao próximo bit é feito quando  $SCLK = 1$ , colocando o bit seguinte a '1' (1/4 do valor do fim de escala). Se a tensão de entrada for inferior à gerada pelo DA então o MSB é colocado a '0' e o bit seguinte é testado. Este processo é repetido até terem sido encontrados todos os bits (Figura 4.20).

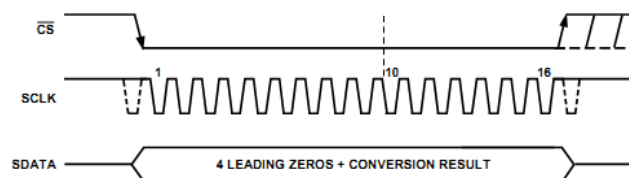


Figura 4.20 - Diagrama temporal do ADC [34].

Este conversor executa uma conversão de aproximação sucessiva usando um conversor DA para fazer redistribuição de cargas. A conversão envolve três fases distintas: amostragem (*sample*), retenção (*hold*) e redistribuição de cargas (*charge redistribution*) [34]. A função de transferência do ADC é apresentada na Figura 4.21:

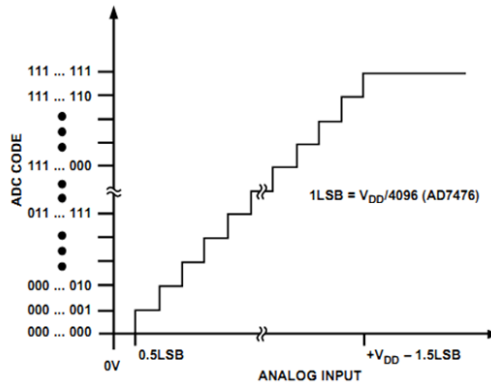


Figura 4.21 - Função de transferência do ADC [34].

#### 4.8.2 Circuito *AD1\_controller*

O circuito *AD1\_controller* é o circuito que controla o circuito PmodAD1 através do protocolo SPI [36].

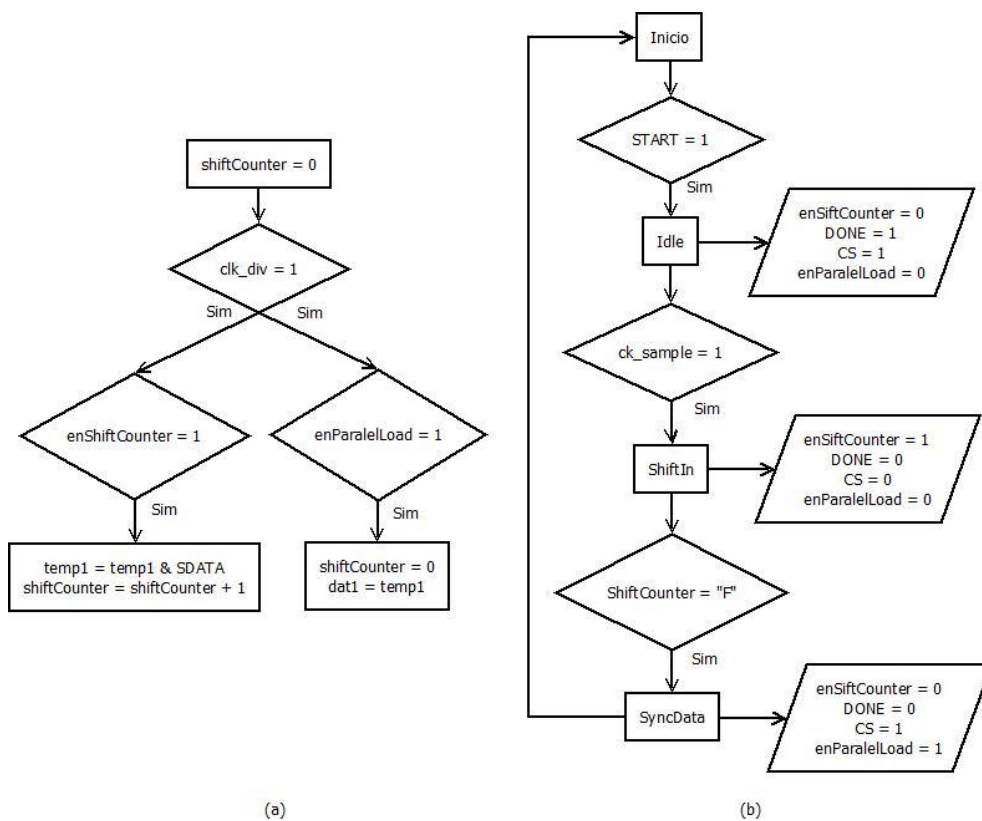
As entradas do circuito *AD1\_controller* são o relógio do circuito (*CLK*) com 0.48µs de período, um sinal de *reset* para reiniciar o componente (*RST*), o bit com o valor da aproximação da amostra (*SDATA1*), a ordem de iniciar a aquisição de uma amostra (*ck\_sample*) que é um relógio com 10µs de período e a ordem para iniciar a amostragem (*START*).

As saídas do circuito *AD1\_controller* são o relógio que vai controlar o circuito PmodAD1 (*SCLK*) que é igual ao relógio do circuito *AD1\_controller* (*CLK*) mas com um desfaseamento de meio período (ou seja, negado), um sinal que seleciona qual o canal de conversão é utilizado e dá ordem para reter a amostra no PmodAD1 até completar a conversão e tenha a amostra de 12 bits disponível (*CS*), o registo com a amostra de 12 bits (*DATA1*) e o sinal que indica que a amostra está disponível (*DONE*).

- A máquina de estados do circuito *AD1\_controller* é iniciada no estado *Inicio* onde verifica se já foi dada a ordem de iniciar a amostragem (*START = 1*).
- Quando for dada a ordem para se iniciar a amostragem, a máquina de estados evolui para o estado *Idle*, que é um estado onde o circuito aguarda que o seja dada ordem para iniciar a aquisição de uma amostra. Esta ordem vem de um relógio externo ao circuito, com período de 10µs, que é o período de amostragem desejado.
- Quando for dada a ordem para iniciar a aquisição de uma amostra a máquina de estados evolui para o estado *ShiftIn* onde se dá ordem ao PmodAD1 para

iniciar a conversão ( $CS = 0$ ) e se vai preenchendo um registo de 16 bits com o bit proveniente do circuito PmodAD1 com o valor da aproximação da amostra. Este processo demora 16 ciclos de relógio ( $Clk\_div$ ) a preencher o registo *temp1* com uma amostra. O relógio  $Clk\_div$  tem 0.4 $\mu$ s de período.

- Após os 16 ciclos de relógio, a máquina de estados evolui para o estado *SyncData* onde é dada a ordem ao PmodAD1 para terminar a amostragem ( $CS=1$ ) e é dada a ordem para se transferir os 12 bits menos significativos do registo *temp1* para o registo *dat1* para se enviar a amostra para a saída do circuito e permitir receber outra amostra.
- Após terminar estas operações, a máquina de estados evolui de novo para o estado *Inicio*, onde repete o processo para adquirir novas amostras.



Fluxograma 4.7 - Circuito *AD1\_controller*. (a) – Processo que controla o preenchimento do registo com as amostras digitalizadas e a saída da amostra de 12 bits. (b) – Máquina de estados do circuito *AD1\_controller*.

Para se obter uma amostra são necessários:

- 16 ciclos de deslocamento do bit da aproximação para obter uma amostra.
- 1 ciclo para fazer o carregamento paralelo da amostra para a saída do circuito.
- 1 ciclo para passar do estado *Inicio* para o estado *Idle* para continuar a amostragem.

- Pelo menos 1 ciclo para começar a aquisição de outra amostra.

Ou seja, são necessários pelo menos 19 ciclos de relógio  $Clk\_div$ . Estes 19 ciclos têm que demorar no máximo o período de amostragem que é  $10\mu s$ . Isto implica que o período do  $Clk\_div$  não pode ultrapassar os  $\frac{10\mu s}{19} = 0.526\mu s$ . Para satisfazer esta restrição, implementou-se divisor do relógio da FPGA em que o período pretendido era  $0.48\mu s$ . Calculou-se que são necessários  $\frac{0.48\mu s}{20ns} = 24$  ciclos de relógio da FPGA para fazer estes  $0.48\mu s$ . No Anexo C apresenta-se um exemplo da aquisição de duas amostras.

## 4.9 Circuito produção do sinal analógico

Para produzir o sinal analógico recorre-se a um conversor DA “PmodDA2” [37] (Figura 4.22).



Figura 4.22 - PmodDA2 [37].

O circuito PmodDA2 produz sinais analógicos através da sua representação digital numa gama dos 0 aos 3.3 Volt e numa escala de 0 a 4095 (12 bits). Tem dois canais de conversão simultânea, cada canal com um conversor de 12 bits (DAC121S101 [38]). Pode-se utilizar os dois canais em simultâneo ou apenas um canal. Para fazer a conversão, o circuito PmodDA2 utiliza o protocolo SPI/MICROWIRE™ [35]. O PmodDA2 liga-se à FPGA aos pinos de Entrada/Saída (E/S) através do *J1 Connector*. A saída analógica liga-se ao *J2 Connector*.

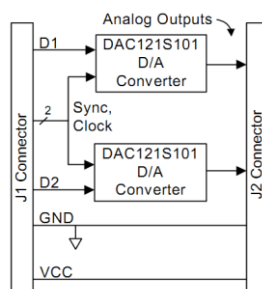


Figura 4.23 - Diagrama de blocos do circuito PmodDA2 [37].

### 4.9.1 Conversão DA

O circuito DAC121S101 tem como entradas as alimentações  $GND$  e  $V_A$  (sendo que  $V_A$  também é referência para as entradas), o sinal digital num registo de 16 bits correspondente ao sinal analógico desejado ( $D_{IN}$ ), o relógio do componente que determina o ritmo de entrada dos bits em série em  $V_A$  ( $SCLK$ ) e o sinal que sincroniza a entrada dos dados ( $SYNC$ ). A saída deste componente é o valor analógico correspondente ao registo de entrada no componente ( $V_{OUT}$ ).

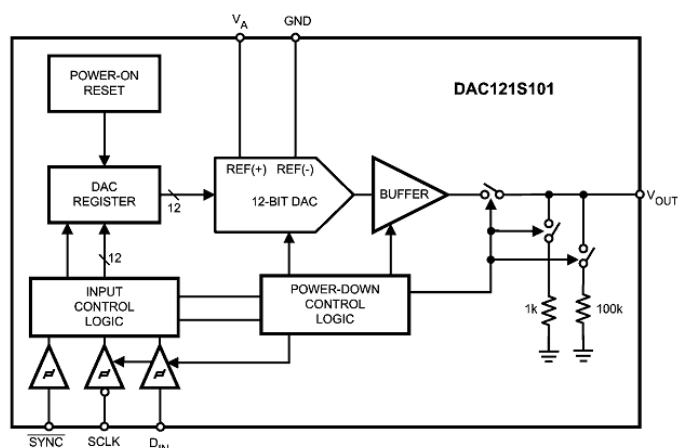


Figura 4.24 - Diagrama de blocos do circuito DAC121S101 [38].

O circuito DAC121S101 tem a seguinte característica de transferência:

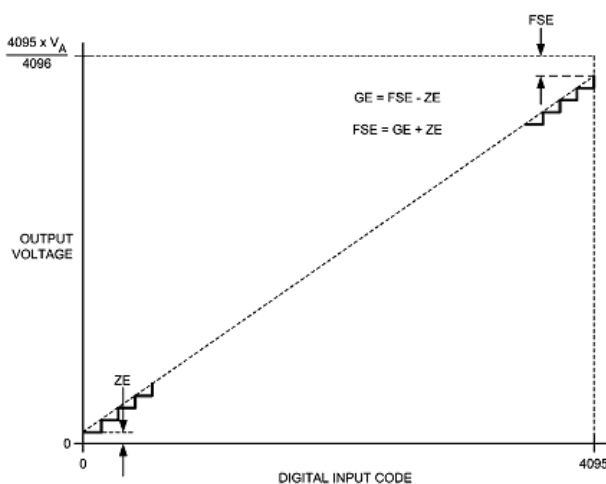


Figura 4.25 - Característica de transferência de entrada / saída do circuito DAC121S101 [38].

O diagrama temporal de conversão do AD7476A é o seguinte:

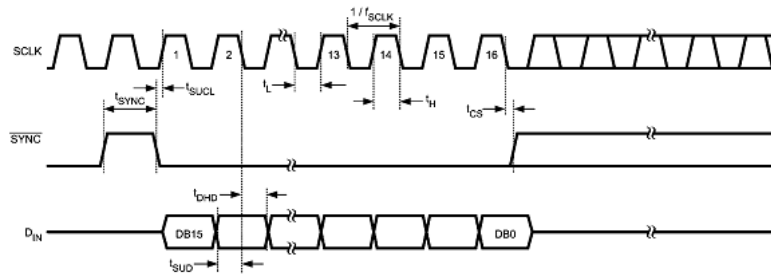


Figura 4.26 - Diagrama temporal do circuito DAC121S101 [38].

### 4.9.2 Circuito DA2\_controller

O circuito *DA2\_controller* é o circuito que controla o circuito PmodDA2 através do protocolo SPI [36].

As entradas do circuito *DA2\_controller* são o relógio do circuito (*CLK*) que é o relógio da FPGA (50 MHz), um sinal de *reset* para reiniciar o componente (*RST*), a ordem para iniciar a produção do impulso analógico (*START*), quatro registos com valores proveniente da interface gráfica, sendo um com o valor de metade da duração do impulso analógico a produzir (*Meio\_Periodo*), outro com o valor do final da duração (*Fim\_Periodo*), outro com o valor da amplitude máxima do impulso analógico a produzir (*Amplitude\_maxima*) e outro com o valor da amplitude mínima.

As saídas do circuito *DA2\_controller* são o relógio que vai controlar o circuito PmodDA2 (*CLK\_OUT*), o sinal que vai sincronizar início da conversão DA (*nSYNC*), o sinal que indica que a conversão foi feita (*DONE*) e um sinal que vai sincronizar dois relés no condicionamento dos sinais analógicos (*switch\_3\_4*) para abrir o relé anterior à célula do fio vibrante e fechar o relé posterior à célula do fio vibrante.

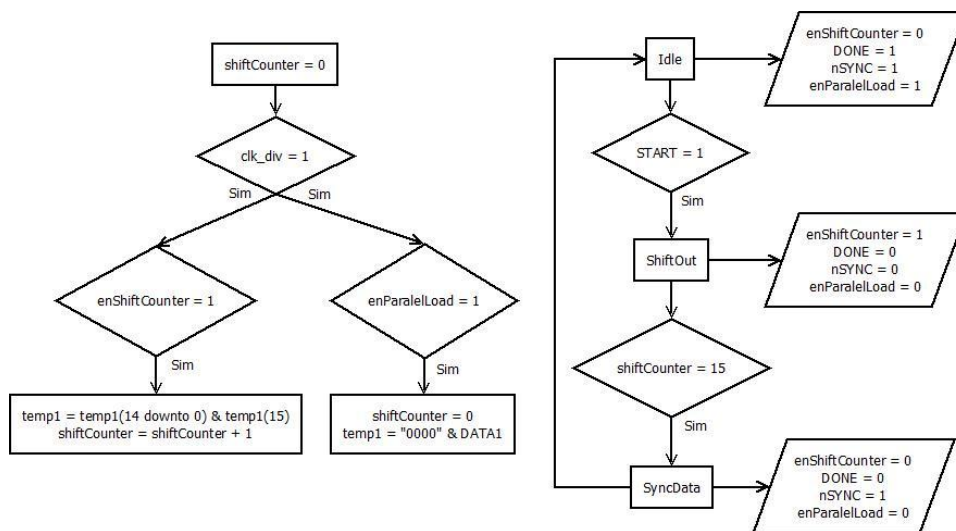
Neste circuito foram implementadas duas máquinas de estado, sendo uma para controlar a produção do impulso analógico e outra para decidir quais as saídas do circuito *DA2\_controller*.

O relógio que vai controlar o circuito PmodDA2 na produção do impulso analógico (*CLK\_OUT*) é igual ao *clk\_div* que é uma divisão da frequência do relógio da FPGA (20ns) por quatro, obtendo-se assim um período de 80ns.

- A máquina de estado que controla a produção impulso analógico é iniciada no estado *Idle* onde permanece até que se recebam todos os dados da interface

gráfica e se gravem na RAM de entrada da FPGA, após o qual é gerado um sinal  $START = 1$ .

- Quando receber o sinal  $START = 1$  a indicar que pode iniciar a conversão DA, a máquina de estados evolui para o estado *ShiftOut* onde permanece 15 ciclos de relógio  $clk\_div$  com período de 80ns. Nestes 15 ciclos a máquina de estados desloca bit a bit do MSB para o LSB o valor da amplitude do impulso analógico a gerar (guardado no registo *DATA1*) para o registo temporário *temp1*.
- Quando o contador *shiftCounter* contar 16 ciclos de relógio (do 0 ao 15) a máquina de estados evolui para o estado *SyncData* onde o sinal *nSync* é produzido para dar ordem para levar os dados presentes no registo temporário *temp1* para a saída que por sua vez vão entrar no chip DAC121S101 no circuito PmodDA2 para produzir o impulso analógico com a amplitude contida em *DATA1*.
- De seguida a máquina de estados evolui para o estado *Idle* onde é gerado o sinal *DONE* a indicar que um sinal analógico foi gerado com sucesso e se reinicia o contador. Se ainda se verificar que  $START = 1$ , ou seja, se ainda não se terminar a produção do impulso analógico, o processo repete-se.



Fluxograma 4.8 - Máquina de estados para controlar a produção do impulso analógico.

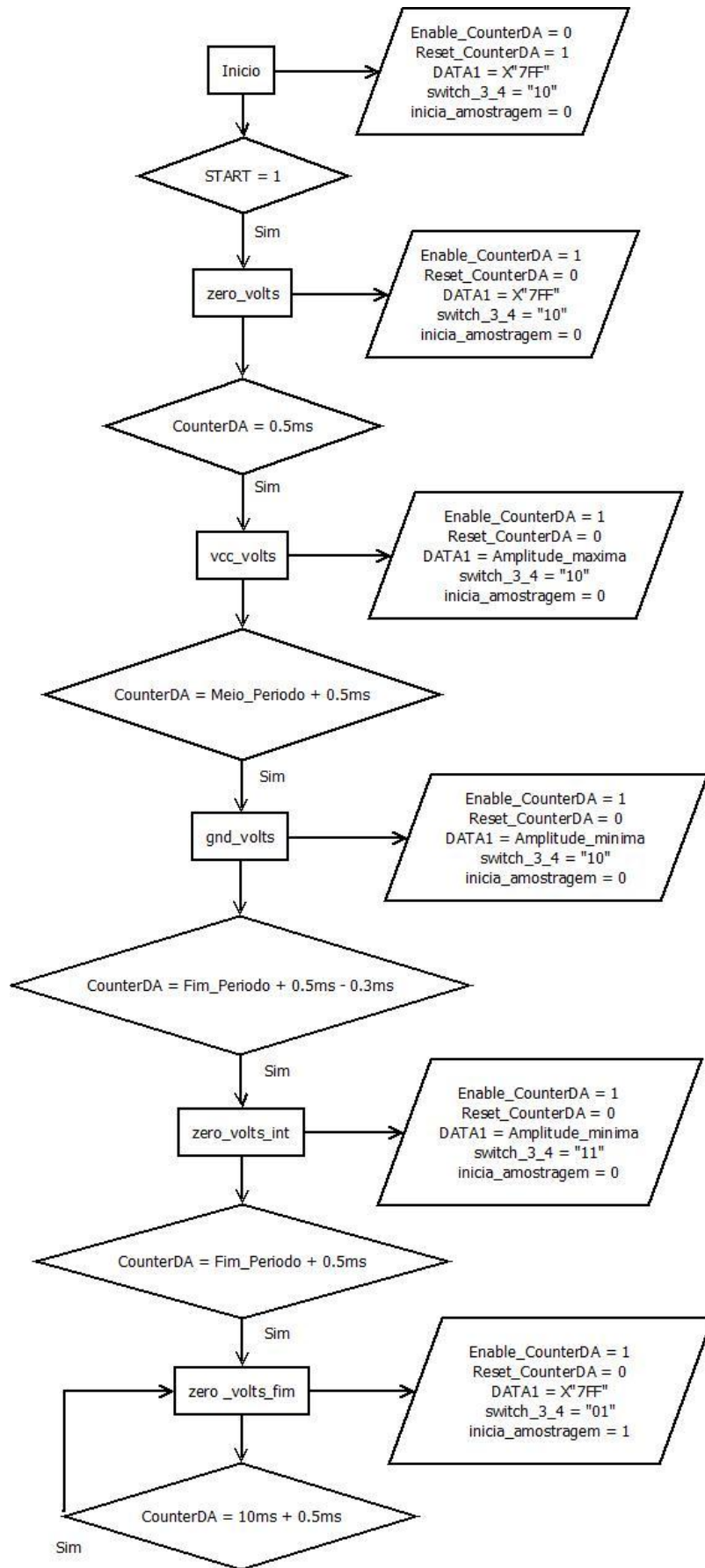
Paralelamente à máquina de estados que controla a produção do impulso analógico existe outra máquina de estados que controla a amplitude do impulso analógico, sincroniza os relés colocados antes e depois da célula do fio vibrante e dá a ordem para se iniciar a digitalização da resposta do fio vibrante.

- Esta máquina de estado é iniciada no estado *Inicio* onde permanece até que se recebam todos os dados da interface gráfica e se gravem na RAM de entrada da FPGA, após o qual é gerado um sinal  $START = 1$ .

- No estado *Início* é produzido um sinal analógico com amplitude 1.65 V e é dada a ordem para fechar o relé colocado antes da célula de fio vibrante (*switch\_3\_4* = "10").
- Quando receber o sinal *START* = 1 a indicar que pode iniciar a conversão DA, a máquina de estados evolui para o estado *zero\_volts* onde é dada a ordem para se iniciar um contador que controla o tempo de permanência em cada estado (*CounterDA*).
- No estado *zero\_volts*, após 0.5ms de produção de 1.65 V, a máquina de estados evolui para o estado *vcc\_volts*.
- No estado *vcc\_volts* após se produzir a amplitude máxima do impulso analógico durante metade da duração do impulso, a máquina de estados evolui para o estado *gnd\_volts*.
- No estado *gnd\_volts* após se produzir a amplitude mínima do impulso analógico até ao final da duração do impulso analógico menos 0.3ms, a máquina de estados evolui para o estado *zero\_volts\_int*.
- No estado *zero\_volts\_int* após se produzir a amplitude mínima do impulso analógico durante 0.3ms, a máquina de estados evolui para o estado *zero\_volts\_fim*. Neste estado é dada a ordem para também se fechar o relé colocado depois da célula de fio vibrante (*switch\_3\_4* = "11"). Esta ordem é dada 0.3ms antes do final do período porque este é o tempo de ativação do relé.
- No estado *zero\_volts\_fim* produz-se 1.65 V e dá-se ordem para se abrir o relé colocado antes da célula de fio vibrante (*switch\_3\_4* = "01"). Dá-se ainda a ordem para se iniciar a amostragem (*inicia\_amostragem* = 1).
- No estado *zero\_volts\_fim*, quando o contador contar 10 ms (mais os 0.5ms iniciais) a máquina de estado permanece no estado *zero\_volts\_fim*. Caso se pretenda obter um sinal periódico (em vez de um único impulso) basta alterar o próximo estado para o estado *Início*. Caso se pretenda um sinal com 10ms de período ou outro valor desejado basta alterar este valor.

Ambas as máquinas de estados são controladas pelo relógio *clk\_div* que tem um período de 80ns, assim como o contador *CounterDA*. Para controlar o tempo de permanência em cada estado, efetua-se o seguinte cálculo:

$$Contador = \frac{\text{Tempo de permanência no estado}}{80 \times 10^{-9}} \quad (4.2)$$



Fluxograma 4.9 - Máquina de estados para decidir quais as saídas do circuito *DA2\_controller*.

## 4.10 Circuito de sincronização

O circuito de sincronização foi denominado por *circuito3*. Este circuito é responsável por ler os dados da memória RAM de entrada da FPGA e por gravar as amostras na memória RAM de saída da FPGA.

Neste circuito lê-se os dados da memória RAM de entrada da FPGA (*RDdata*) no endereço selecionado (*RDaddrCIR*) e guardam-se em registos, ligando estes registos aos circuitos onde vão ser utilizados. Como foram enviados diferentes dados com diferente número de dígitos (diferentes tamanhos) da interface gráfica para a RAM de entrada na FPGA e cada dígito hexadecimal corresponde a 4 bits e cada endereço da RAM de entrada na FPGA tem 32 bits, vai haver endereços com mais do que um conjunto de informação. Os dados foram enviados da interface gráfica para a RAM de entrada na FPGA pela seguinte ordem e com o seguinte tamanho:

- Número de ciclos de relógio do DAC para indicar metade da duração do impulso analógico (*Meio\_Periodo*) – 8 dígitos hexadecimais = 32 bits.
- Número de ciclos de relógio do DAC para indicar a duração total do impulso analógico (*Fim\_Periodo*) – 8 dígitos hexadecimais = 32 bits.
- Amplitude mínima do impulso (*Amplitude\_minimo*) – 4 dígitos hexadecimais = 16 bits.
- Amplitude máxima do impulso (*Amplitude\_maxima*) – 4 dígitos hexadecimais = 16 bits.
- Se se pretende utilizar a resistência de 10 Ohm ou a resistência de 100 Ohm (*Switch\_escolhido*) – 2 dígitos hexadecimais = 8 bits.
- Se se pretende a resistência de 0.1Ohm ou não (*Resistência\_pequena*) – 2 dígitos hexadecimais = 8 bits.
- Valor da amplificação escolhida (*Amplificação*) – 4 dígitos hexadecimais = 16 bits.
- Valor da taxa de amostragem (*Taxa\_amostragem*) – 4 dígitos hexadecimais = 6 bits.

Para se guardar a informação da RAM de entrada na FPGA para registos implementou-se um contador (*addrCnt\_leitura*) que é incrementado pelo relógio do *circuito3* (que é o mesmo que o do circuito *AD1\_controller*, com período de 0.48 $\mu$ s) e que seleciona o endereço da RAM de entrada na FPGA. Para cada endereço de memória, guarda-se os valores em registos da seguinte maneira:

- No endereço '0' lê-se o valor de metade da duração do impulso.
- No endereço '1' lê-se o valor da duração total do impulso.

- No endereço '2', do bit '15' ao '0' lê-se o valor da amplitude mínima, do bit '31' ao bit '16' lê-se o valor da amplitude máxima.
- No endereço '3', do bit '7' ao '0' lê-se a resistência escolhida (de 10 ou 100 Ohm), do bit '15' ao bit '8' lê-se se foi escolhida a resistência de 0,1 Ohm, do bit '31' ao '16' lê-se a amplificação escolhida.
- No endereço '4', do bit '15' ao '0' lê-se a taxa de amostragem.
- No endereço '5' sabe-se que já se leu toda a informação que havia para ler e deixa-se essa informação disponível para se utilizar noutros circuitos.

Por outro lado, este circuito é também responsável por sincronizar o início da amostragem (*inic\_amostragem*) e gravar as amostras (*Amostra*) na memória de saída da FPGA (*WRdata*) no endereço selecionado (*WRaddrCIR*).

Tendo em conta que o ADC produz amostras de 12 bits e a memória de saída da FPGA tem 32 bits por endereço, optou-se por se colocar duas amostras por endereço, adicionando "0000" à esquerda de cada amostra ("0000" & Amostra) de modo a que cada amostra tenha 16 bits sem alterar o seu valor. Para isso, utilizaram-se três registos temporários, em que um deles (*amostra\_1\_16\_bit*) guarda uma amostra de 16 bits assim que é produzida uma amostra ímpar (a 1ª, a 3ª, a 5ª amostras, ...) outro (*amostra\_2\_16\_bit*) guarda uma amostra de 16 bits quando é produzida uma amostra par (a 2ª, a 4ª, a 6ª amostras, ...) e outro (*amostra\_32bit\_memSaida*) que guarda estas duas amostras num registo de 32 bits quando é produzida uma amostra ímpar para garantir que já foram produzidas duas amostras. É a informação deste registo que é gravada em cada endereço da memória de saída da FPGA, controlado por um contador (*addCnt*) a cada ciclo de relógio do *circuito3*.

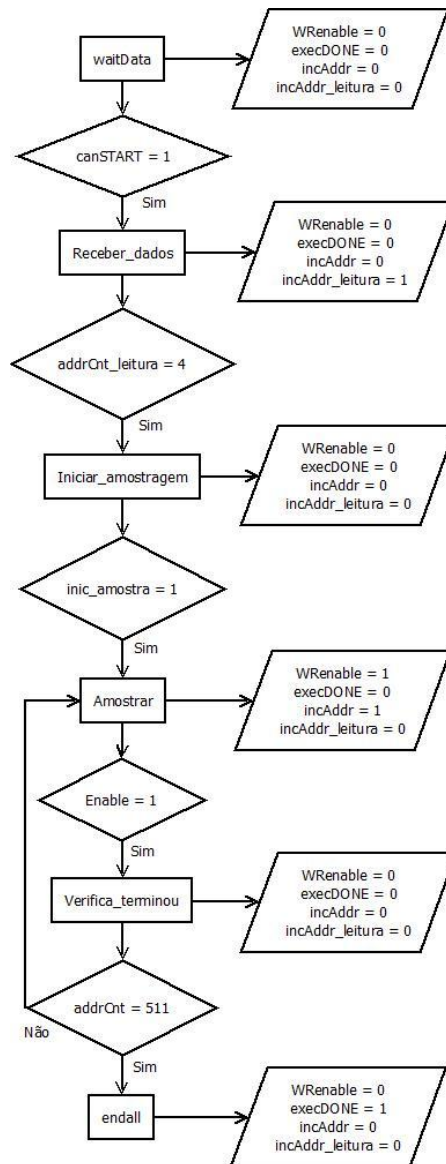
É importante ter em conta que este contador é diferente do contador que incrementa o endereço da memória RAM de entrada na FPGA.

Para se saber se a amostra produzida é par ou ímpar implementou-se um contador que é incrementado quando o sinal *DONE* do circuito *AD1\_controller* é produzido. Deste modo, lendo o bit menos significativo consegue-se saber se o número de amostras produzidas é par ou ímpar.

Para sincronizar estas operações foi implementada uma máquina de estados.

- A máquina de estados é iniciada no estado *waitData* onde permanece até receber a ordem para iniciar o processamento (*START = 1*) que indica que já chegaram todos os dados à memória RAM de entrada na FPGA.
- Quando se receber a ordem para iniciar o processamento, a máquina de estados evolui para o estado *Receber\_dados* onde é dada a ordem para incrementar o contador que seleciona o endereço da memória RAM de entrada na FPGA (*incAddr\_leitura*).

- Quando se chegar ao endereço 4, ou seja, quando se tiver lido toda a informação da memória RAM de entrada na FPGA, a máquina de estados evolui para o estado *Iniciar\_amostragem* .
- No estado *Iniciar\_amostragem* verifica-se se já se recebeu a ordem para se iniciar a amostragem (*inic\_amostragem*) que é produzida quando se acabar de produzir o impulso com a duração selecionada pelo utilizador.
- No estado *Iniciar\_amostragem*, quando for recebida a ordem para iniciar a amostragem, a máquina de estados evolui para o estado *Amostrar* onde é incrementado o contador que seleciona o endereço da memória RAM de saída da FPGA (*incAddr*) e autoriza a escrita na mesma (*WRenable*).
- No estado *Amostrar*, quando se produzir uma amostra ímpar, a máquina de estados evolui para o estado *Verifica\_terminou* onde se deixa de incrementar o endereço de memória RAM de saída da FPGA e de permitir escrever na mesma.
- No estado *Verifica\_terminou* verifica-se se o endereço de memória RAM de saída da FPGA é o último.
- Se não for, a máquina de estados volta ao estado *Amostrar* para gravar mais duas amostras.
- Se for o último endereço da memória, a máquina de estados evolui para o estado *endall* onde é produzido um sinal a indicar que se guardaram todas as amostras (*execDONE*).



Fluxograma 4.10 - Máquina de estados do circuito de sincronização.

## 4.11 Circuito de interligações

O circuito *FPGA3* é o circuito que contém todos os restantes circuitos apresentados anteriormente e é o circuito responsável por interligar as entradas e saídas entre os circuitos. É neste circuito que se ligam as entradas e saídas da FPGA, nomeadamente as saídas digitais que vão controlar os relés, os *leds*, os *switches*, os botões, os *displays* de 7 segmentos, se recebe o relógio da FPGA (de 50MHz), recebe e envia o bit da comunicação série através do cabo *Série\_USB* e os sinais que vão controlar o DAC e o ADC.

No circuito *FPGA3* foi implementada a capacidade de se verificar os dados que chegam à memória RAM de entrada na FPGA e os valores que estão na memória RAM de saída da FPGA. Para isso, utiliza-se os *switches* da placa para introduzir o endereço a visualizar e observa-se o seu valor nos *displays* de 7 segmentos. Por omissão são visualizados os 2 bytes menos significativos da memória de entrada. Pressionando o botão 2, observam-se os 2 bytes mais significativos da memória de entrada. O botão 3 permite visualizar os 2 bytes menos significativos da memória de saída. Pressionando em simultâneo os botões 2 e 3 pode-se observar os 2 bytes mais significativos da memória de saída.

Foi ainda implementada a capacidade de se verificarem os sinais que indicam que chegaram todos os dados à memória de entrada na FPGA (*canSTART*), que foram gravadas todas as amostras na memória de saída da FPGA (*execDONE*) e que foram enviados todos os dados da memória de saída da FPGA para o computador (*canDebug*). Nos leds 7 a 5 são indicados os sinais *canDebug*, *execDONE* e *canSTART*, respetivamente.

As verificações apresentadas podem ser feitas após os dados terem sido enviados para o computador (no estado *stEND* do circuito *interface\_serie*).

É neste circuito que se seleciona qual a saída digital a ativar. Foram reservadas cinco saídas digitais para a placa blindada e 8 saídas digitais para a placa não blindada. Estas saídas digitais estão ligadas a relés, permitindo ativá-los em função das escolhas do utilizador (gravadas na memória de entrada na FPGA e lidas no *circuito3*). As saídas digitais de estão ligados a relés para ativar:

- *saida\_digital\_1* – resistência de 10  $\Omega$ .
- *saida\_digital\_2* – resistência de 100  $\Omega$ .
- *saida\_digital\_3* – relé colocado antes da célula do fio vibrante.
- *saida\_digital\_4* – relé colocado depois da célula do fio vibrante.
- *saida\_digital\_5* – resistência de 0.1  $\Omega$ .
- *saida\_digital\_10* – amplificação de 1 vez (terceiro andar de amplificação).
- *saida\_digital\_11* – amplificação de 2 vezes (terceiro andar de amplificação).
- *saida\_digital\_12* – amplificação de 5 vezes (terceiro andar de amplificação).
- *saida\_digital\_13* – amplificação de 10 vezes (terceiro andar de amplificação).
- *saida\_digital\_14* – amplificação de 1 vez (segundo andar de amplificação).
- *saida\_digital\_15* – amplificação de 10 vezes (segundo andar de amplificação).
- *saida\_digital\_16* – amplificação de 1 vez (primeiro andar de amplificação).
- *saida\_digital\_17* – amplificação de 10 vezes (primeiro andar de amplificação).

Utilizando o esquema de codificação apresentada na secção 3.2.2, as saídas digitais são de 10 a 17 são ativadas quando a amplificação escolhida for:

- *saida\_digital\_10* – 1, 10 e 100 vezes.

- `saida_digital_11` – 2, 20 e 200 vezes.
- `saida_digital_12` – 5, 50 e 500 vezes.
- `saida_digital_13` – 1000 vezes.
- `saida_digital_14` – 1, 2 e 50 vezes.
- `saida_digital_15` – 10, 20, 50, 100, 200, 500 e 1000 vezes.
- `saida_digital_16` – 1, 2, 5, 10, 20 e 50 vezes.
- `saida_digital_17` – 100, 200, 500 e 1000 vezes.

É no circuito *FPGA3* que estão implementados o divisor do relógio da FPGA para definir a taxa de amostragem (*ckSample*) e o divisor de relógio da FPGA para definir o relógio dos circuitos *AD1\_controller* e do *circuito3* (*Shift\_clk*).

## 4.12 Ficheiro UCF

O ficheiro UCF (*User Constrain File*) especifica os pinos de entrada e de saída da FPGA. Para além de especificar os pinos referentes aos *displays* de 7 segmentos, dos botões, dos *leds*, dos *switches* e do relógio de 50 MHz relativos à FPGA utilizada é ainda necessário especificar onde se ligam os sinais que vão comandar o DAC, o ADC, as saídas digitais e os bits de enviados e recebidos pelo cabo *Série\_USB*.

```
#para o AD1:
NET "bitSelkAdc" LOC = M6;

NET "bitDIAdc" LOC = D15;
NET "bitCsAdc" LOC = E15;

#para o DA2:
NET "bitClkDac" LOC = T3;
NET "bitD1DAC" LOC = N11;

NET "bitnSyncDac" LOC = R10;

#para os relés da placa blindada
NET "saida_digital_1" LOC = A13; # resistência de 10 Ohm
NET "saida_digital_2" LOC = A12; # resistência de 100 Ohm
NET "saida_digital_3" LOC = B11; # Entrada do vio vibrante
NET "saida_digital_4" LOC = B10; # Saída do vio vibrante
NET "saida_digital_5" LOC = A7; # resistência de 0.1 Ohm
#para os relés da placa NÃO blindada
#3° bloco
NET "saida_digital_10" LOC = B5; # Amplifica 1X
NET "saida_digital_11" LOC = B4; # Amplifica 2X
NET "saida_digital_12" LOC = D10; # Amplifica 5X
NET "saida_digital_13" LOC = D8; # Amplifica 10X
#2° bloco
NET "saida_digital_14" LOC = D7; # Amplifica 1X
NET "saida_digital_15" LOC = E7; # Amplifica 10X
#1° bloco
NET "saida_digital_16" LOC = D6; # Amplifica 1X
NET "saida_digital_17" LOC = D5; # Amplifica 10X

#PACE: Start of PACE I/O Pin Assignments
NET "CLKB" LOC = "T9" ;
NET "RXD" LOC = "T13" ;
NET "TXD" LOC = "R13" ;
```

Figura 4.27 - UCF com localização dos sinais para DAC, ADC, saídas digitais, relógio e bits enviados e recebidos do cabo *Série\_USB*.

# Capítulo 5

## 5 Condicionamento dos sinais analógicos

### 5.1 Visão global

Neste capítulo vão ser apresentadas as opções efetuadas para a produção do impulso de corrente a estimular o fio vibrante, para a amplificação e aquisição da resposta do fio vibrante e para a blindagem utilizada.

Depois de o utilizador definir na interface gráfica a duração e amplitude do impulso de corrente a estimular o fio vibrante e de esses parâmetros serem recebidos pela FPGA para controlar o PmodDA2 na produção do impulso, é necessário condicionar esse impulso por forma a excitar o fio vibrante no modo desejado.

O impulso de corrente a estimular o fio vibrante deve ter o formato de uma onda quadrada simétrica com a primeira metade do período positiva e a segunda metade do período negativa [3]. Os pinos de E/S da FPGA e do PmodDA2 apenas fornecem tensões entre 0V e 3.3V. Deste modo apenas é possível produzir um impulso com o formato de uma onda quadrada simétrica, centrado em 1.65V com a primeira metade do período com amplitude superior a 1.65V (mas inferior a 3.3V) e a outra metade do período com amplitude inferior a 1.65V (mas superior a 0V). Como se pretende que este impulso seja centrado em torno de 0V (e não de 1.65V) teve que se conceber um bloco responsável por fazer o deslocamento de nível deste impulso sem o distorcer. Esse bloco está representado na Figura 5.1 por A.

De seguida, o impulso entra numa blindagem eletrostática, sendo que para tal é necessário isolar o sinal do exterior através de isoladores óticos (*IO*) que impedem o fluxo de correntes de ruído e permitem que o nó de terra não esteja a flutuar e seja ligado à blindagem eletrostática que envolve todo o circuito e que está ligado à parede externa da célula de medição, proporcionando uma proteção contra o ruído externo.

Como o fio vibrante é estimulado por corrente, é necessário utilizar um amplificador de potência que permita obter um impulso com amplitude eficaz na ordem das centenas de miliamperes. Este amplificador de potência está representado na Figura 5.1 por *B*.

Tal como em [2] [3] [10] é necessário utilizar uma resistência em série, colocada entre o impulso produzido e a célula de medida para limitar a corrente que atravessa a célula de medida. Tendo em conta que se podem utilizar fios vibrantes com diferentes raios, dependendo da viscosidade do líquido em estudo, optou-se por se colocar duas resistências  $R_9$  e  $R_{10}$  diferentes em paralelo e dois relés  $S_1$  e  $S_2$  que permitem selecionar qual das resistências é percorrida pelo impulso, condicionando assim a corrente que atravessa o fio vibrante.

Durante a aplicação do impulso no fio vibrante *FV* é necessário sincronizar dois relés de modo a que o relé  $S_3$  esteja a conduzir durante a duração do impulso e o relé duplo  $S_4$  esteja aberto de modo a evitar a influência de correntes que levem a falsas leituras ou erros de medida. Após o final da duração do impulso o relé  $S_3$  deixa de conduzir e o relé duplo  $S_4$  conduz a resposta do fio vibrante para um andar de amplificação *C* com ganho de mil. Quando não se está a fazer nenhum ensaio experimental todos os relés estão abertos de modo a evitar a passagem de corrente para o fio vibrante sem que seja desejado.

De seguida, a resposta do fio vibrante amplificada “sai” da blindagem eletrostática através de um isolador ótico (*IO*) onde vai ser novamente amplificada *D* com ganho selecionado pelo utilizador de modo a ter uma amplitude máxima de 3.3V pico a pico para ser convertida para digital pelo PmodAD1, enviada para a FPGA e por fim para o computador, para a interface gráfica o reconstruir. Para isso é necessário deslocar o sinal à saída de *D* de modo a que fique centrado em 1.65V, ou seja, todo positivo *E*. A montagem implementada pode ser observada no anexo D.

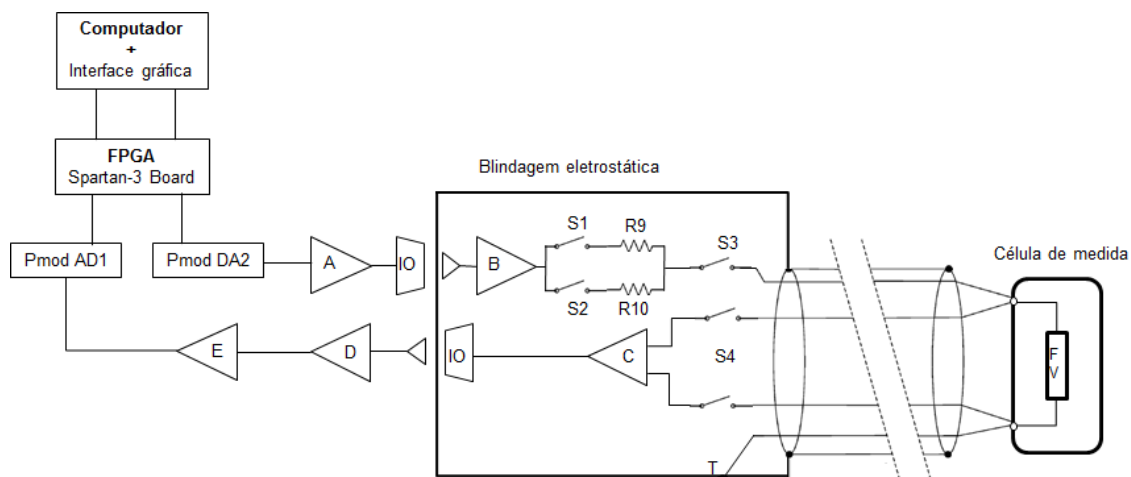


Figura 5.1 - Sistema global.

## 5.2 Esquema elétrico da montagem

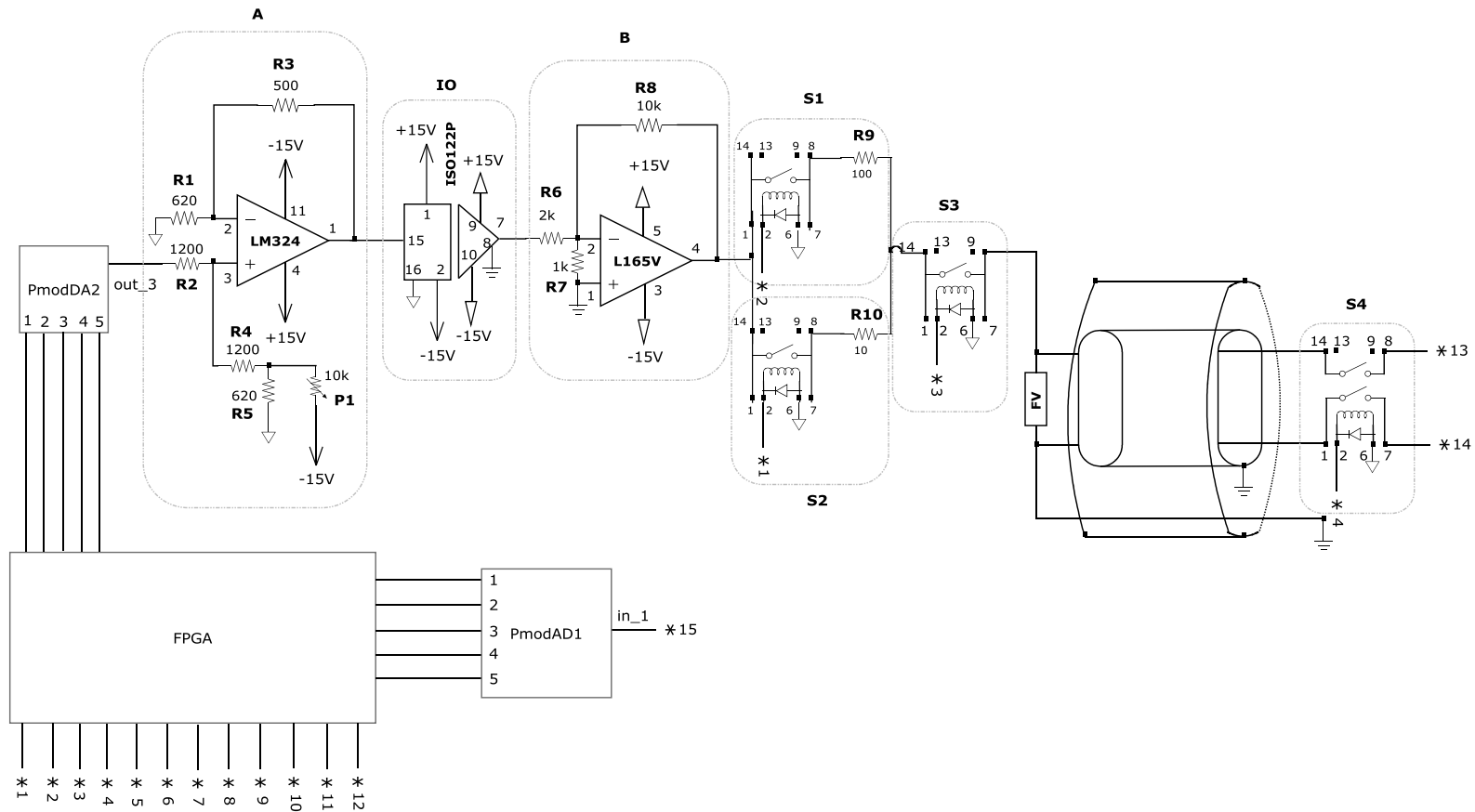


Figura 5.2 - Esquema elétrico. Parte 1.

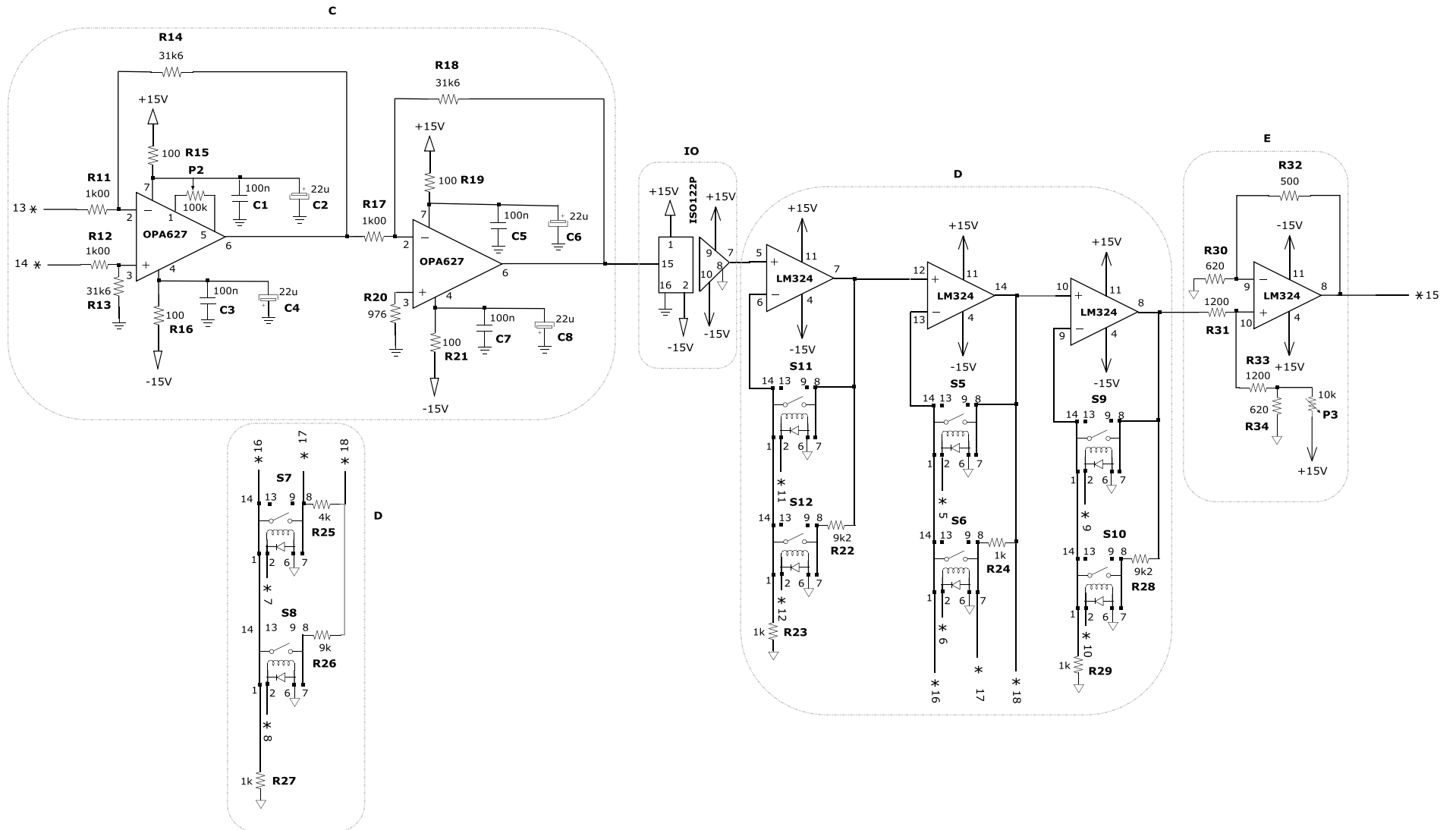


Figura 5.3 - Esquema elétrico. Parte 2.

Como se pode observar, o circuito pode dividir-se numa parte “mais sensível” ao ruído e em outra parte “menos sensível” ao ruído. Como tal, optou-se por se conceber duas placas separadas fisicamente.

A placa responsável pela zona “menos sensível” ao ruído é aquela em que os sinais de trabalho tem amplitude mínima na ordem das dezenas/centenas de milivolts e contém o DAC PmodDA2, o deslocador de nível *A* que vai centrar o impulso a estimular o fio vibrante em torno de 0V, o bloco de amplificação com ganho selecionável *D*, o deslocador de nível *E* que vai centrar a resposta do fio vibrante amplificado em torno de 1.65V e por fim o ADC PmodAD1 que está ligado à FPGA. Esta placa está ligada à FPGA e à outra placa “mais sensível” ao ruído.

A placa responsável pela zona “mais sensível” ao ruído é aquela em que o sinal de trabalho (resposta do fio vibrante) tem amplitude máxima na ordem das dezenas/centenas de microvolts e contém os isoladores óticos *IO* de entrada e de saída dos sinais na placa, o amplificador de potência *B*, as resistências colocadas antes do fio vibrante e dois amplificadores com um ganho total de 1000 vezes que amplificam a resposta do fio vibrante *C*. Esta placa está ligada à outra placa “menos sensível” ao ruído, ao fio vibrante e às saídas digitais da FPGA que controla os relés. Esta placa está envolvida por uma blindagem eletrostática.

### 5.2.1 Deslocador de nível

O deslocador de nível (Figura 5.4) foi realizado utilizando um amplificador operacional *LM324* [39] em que a componente DC é regulada através de um potenciómetro ligado aos -15V quando se pretende colocar o impulso em torno de zero *A* ou ligado aos +15V quando se pretende colocar a resposta do fio vibrante em torno de 1.65V *E*. Utilizou-se um potenciómetro e não uma resistência de valor fixo porque poderia haver a necessidade de se pretender alterar a componente DC sem se alterar o circuito.

Admitindo que o amplificador é ideal e a corrente de entrada aos terminais do amplificador é zero e que existe um curto de circuito virtual aos terminais de entrada do amplificador, pode-se admitir que:

$$I_1 = I_2 = \frac{V_0 - V^+}{R_3} = \frac{V^+ - 0}{R_1} \quad (5.1)$$

Desenvolvendo a expressão anterior, obtém-se:

$$V^+ = V_0 \frac{R_1}{R_1 + R_3} \quad (5.2)$$

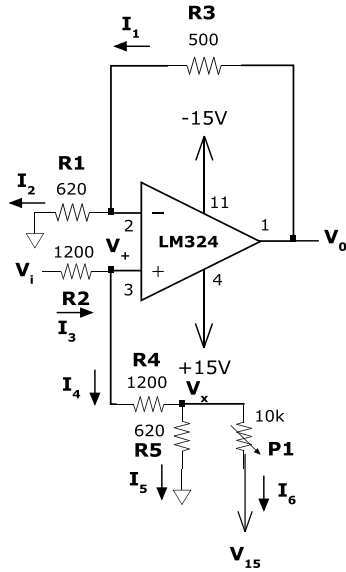


Figura 5.4 - Deslocador de nível.

Sabendo que:

$$I_3 = I_4 = \frac{V_i - V^+}{R_2} = \frac{V^+ - V_x}{R_4} \quad (5.3)$$

Desenvolvendo, obtém-se:

$$V_x = \frac{V_0 \frac{R_1}{R_1 + R_3} (R_2 + R_4) - V_i R_4}{R_2} \quad (5.4)$$

Sabendo também que:

$$I_5 = I_6 = \frac{V_x - 0}{R_5} = \frac{V_x - V_{DC}}{P_1} \quad (5.5)$$

Obtém-se:

$$V_0 = \frac{\left( \frac{R_4(P_1 + R_5)}{R_2 R_5 P_1} + \frac{1}{R_2} \right) V_i + \frac{V_{DC}}{P_1}}{\frac{R_1(R_2 + R_4)(P_1 + R_5)}{R_2(R_1 + R_3)} + \frac{R_1}{R_2(R_1 + R_2)}} \quad (5.6)$$

Se se atribuírem:

- $R_1 = R_5 = 620\Omega$
- $R_2 = R_4 = 1200\Omega$
- $R_3 = 500\Omega$
- $V_{DC} = 15V$

Pode-se simplificar a expressão anterior para:

$$V_0 = V_i + V_{DC} \frac{R_5}{R_1 + R_5} \quad (5.7)$$

Para se obter uma componente DC de 1.65V, pode-se substituir o potenciômetro por uma resistência de valor fixo de 5000Ω.

$$V_0 = V_i + 15 \times \frac{620}{620 + 5000} = V_i + 1.65 \quad (5.8)$$

Como se pode observar na expressão anterior, que define o ganho da montagem do deslocador de nível, quando  $V_{DC}$  é positivo a componente DC é positiva, mas quando  $V_{DC}$  é negativo a componente DC é negativa, permitindo assim utilizar a mesma montagem para centrar o impulso  $V_i$  em torno de 0V ou em torno de 1.65V.

### 5.2.2 Ganho de amplificação selecionável pelo utilizador

Para seleccionar o ganho total de amplificação da resposta do fio vibrante utilizaram-se três amplificadores operacionais em cascata. Uma vez que a resposta do fio vibrante já chega a esta placa amplificado 1000 vezes, optou-se por permitir ao utilizador amplificar a resposta do fio vibrante mais 1, 2, 5, 10, 20, 50, 100, 200, 500 ou 1000 vezes. Para isso, o primeiro andar de amplificação (desta placa) permite seleccionar ganho 1 ou 10, o segundo andar de amplificação permite seleccionar ganho 1, 2, 5 ou 10 e o terceiro andar de amplificação permite seleccionar ganho 1 ou 10. Estes ganhos são seleccionados através de relés que são ativos com as saídas digitais da FPGA. O esquema de amplificação está explicado na secção 3.2.

Para realizar estes andares de amplificação utilizou-se o circuito integrado LM324 que disponibiliza 4 amplificadores operacionais LM324, permitindo realizar os 3 andares de amplificação com um único circuito integrado. Como o ganho máximo necessário é 10 e a largura de banda do circuito é 50kHz, consultando o manual deste componente [39], verifica-se através da resposta de frequência em malha aberta deste componente que é possível obter um ganho em tensão de 10 vezes até uma frequência de 100kHz, acima do necessário.

Os três andares de amplificação foram implementados utilizando uma montagem inversora. Quando, por exemplo o relé S12 da Figura 5.3 está a conduzir, o ganho dessa montagem é dado por:

$$Ganho = \frac{R_{22}}{R_{23}} + 1 = \frac{9200}{1000} + 1 = 10.2 \quad (5.9)$$

Caso seja o relé *S11* da Figura 5.3 que esteja a conduzir, o ganho da montagem é dado por:

$$Ganho = \frac{0}{R_{23}} + 1 = 1 \quad (5.10)$$

O mesmo se passa para os restantes andares de amplificação.

### 5.2.3 Amplificador de potência

O amplificador de potência utilizado [40] foi implementado com ganho de 5. Deste modo, a corrente máxima a estimular o fio vibrante é 0.825 A. Esta corrente é produzida quando o utilizador escolhe a amplitude de 1.65V na interface gráfica que multiplicando pelo ganho 5 e fazendo passar o impulso pela resistência de 10Ω obtém-se:

$$I = \frac{1.65 \times 5}{10} = 0.825A \quad (5.11)$$

O amplificador de potência está ligado a um dissipador de calor que deve ficar em contacto com a parede da blindagem eletrostática para toda esta funcionar como dissipador de calor.

## 5.3 Largura de banda

### 5.3.1 Largura de banda do sinal de resposta do fio vibrante

A resposta do fio vibrante é uma senoide amortecida que é descrita no domínio do tempo pela equação (1.3). O espectro de um sinal sinusoidal amortecido tem uma infinidade de harmónicas que se podem exprimir por:

$$U(\omega_0) = \frac{1}{\sqrt{(\Delta^2 + (\frac{\omega}{\omega_0} - 1)^2)}} \quad (5.12)$$

Como as harmónicas possuem intensidades que decrescem com o aumento da frequência, considera-se apenas a parte do espectro onde se concentra a maioria da energia

do sinal, designando-se esta faixa de frequências por largura de banda efetiva ou simplesmente por largura de banda.

Para definir a largura de banda do sinal de resposta do fio vibrante, admite-se a frequência de oscilação do fio vibrante de 1300Hz e um decaimento de  $\Delta = 0.1$  e verifica-se qual a harmónica que tem uma amplitude cerca de 90 vezes inferior à harmónica de amplitude máxima. Com o auxílio do gráfico seguinte, pode observar-se que a décima harmónica  $10\omega_0$  tem uma amplitude cerca de 90 vezes inferior à harmónica de amplitude máxima (Figura 5.5). Assim sendo, pode-se concluir que a largura de banda do sinal de resposta do fio vibrante é 13 kHz.

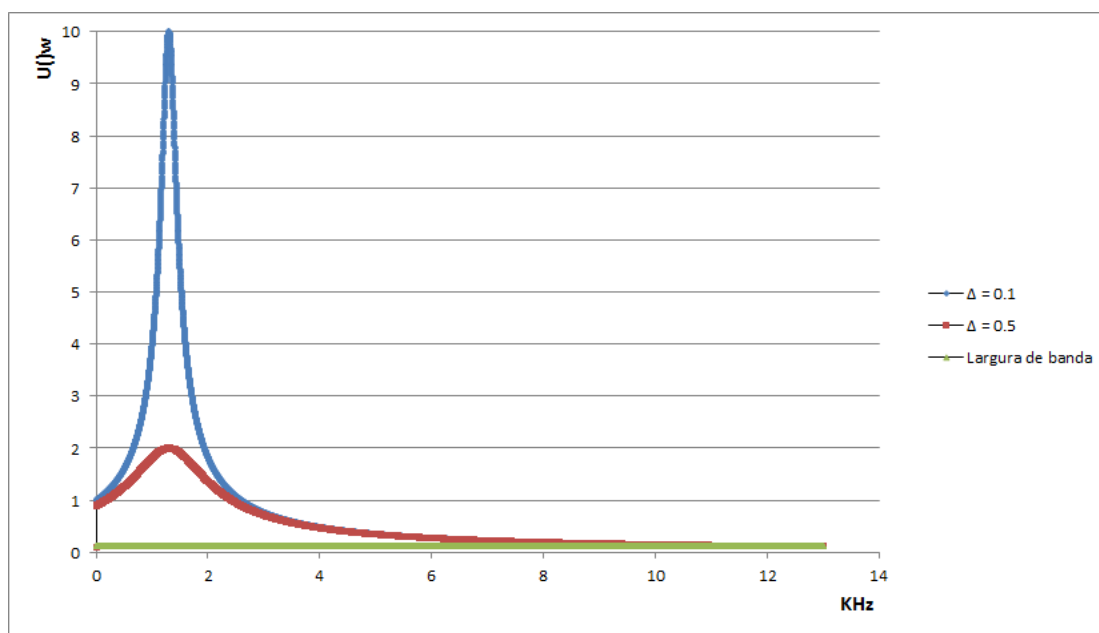


Figura 5.5 - Espectro do sinal de resposta do fio vibrante para diferentes amortecimentos.

### 5.3.2 Largura de banda do circuito

A largura de banda do circuito é imposta pelo componente com menor largura de banda que é o isolador ótico ISO122P e que tem uma largura de banda de 50 kHz [41].

Uma vez que a largura de banda do circuito é superior à largura de banda do sinal, pode-se garantir uma reprodução do sinal sem distorção, mas com aumento do ruído.

## 5.4 Ruído

Os componentes eletrônicos estão sujeitos a diversos tipos de ruído e com diversas origens. Designa-se por ruído todos os sinais que não transportam informação útil.

O ruído térmico resulta da agitação dos elétrons acima do zero absoluto (0 K) e é maior quanto maior for a temperatura. A sua densidade espectral de potência é aproximadamente constante ao longo do todo o espectro e apresenta uma distribuição gaussiana. O ruído de impulsos consiste em picos de energia muito intensos e de curta duração que podem ser provocados por equipamentos externos como como lâmpadas fluorescentes e motores elétricos.

Para reduzir o ruído externo utilizou-se uma blindagem que consiste numa gaiola de Faraday que envolve a célula de fio vibrante e está ligada a uma malha condutora que envolve os dois fios condutores que transportam a resposta do fio vibrante desde a célula do fio vibrante até à placa blindada onde a resposta vai ser amplificada mil vezes. Toda a blindagem, ou seja, a gaiola de Faraday, a malha que envolve os dois condutores da resposta o fio vibrante e a blindagem da placa que vai amplificar a resposta do fio vibrante estão ligadas ao mesmo potencial.

### 5.4.1 Ruído de entrada do primeiro andar do bloco de amplificação

Sendo a resposta do fio vibrante uma diferença de potencial induzida aos terminais do fio, é essa diferença de potencial que se pretende amplificar. Para tal, utilizou-se um amplificador OPA627AP [42] com uma montagem de amplificador diferencial (montagem C do esquema elétrico da Figura 5.3), utilizando para tal resistências de precisão de 1% de forma a minimizar uma diferença de potencial provocado pela diferença das resistências de entrada, conhecida como rejeição de modo comum. O amplificador diferença amplifica a diferença entre dois sinais aplicados à entrada do amplificador  $V_{id}$  e rejeita o sinal comum à entrada  $V_{icm}$ .

Na Figura 5.6 são apresentados os dois andares de amplificação com ganho 1000.

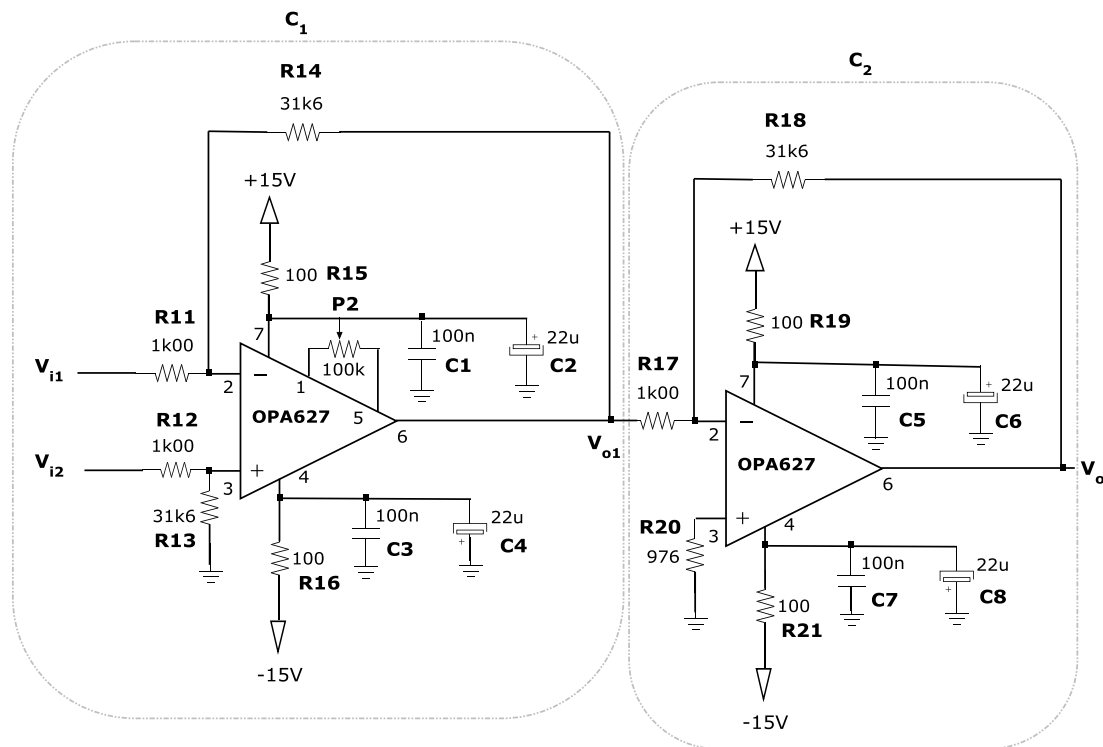


Figura 5.6 - Andares de amplificação com ganho total de 1000.

O ganho da montagem  $C_1$  é dado por:

$$V_{o1} = \frac{R_{14}}{R_{11}} (V_{i2} - V_{i1}) = 31.6 \times (V_{i2} - V_{i1}) \quad (5.13)$$

O ganho da montagem  $C_2$  é o ganho total da montagem e é dado por:

$$V_o = -\frac{R_{18}}{R_{17}} V_{o1} = 31.6 \times 31.6 \times (V_{i1} - V_{i2}) \approx 1000 \times (V_{i1} - V_{i2}) \quad (5.14)$$

Do ponto de vista de ruído, o andar de entrada do bloco de amplificação é o que tem maior influência e é o que deve introduzir menor ruído. No circuito implementado, o andar de entrada do bloco de amplificação tem um ganho de 31.6 seguido de outro andar com ganho de 31.6, o que faz um ganho total de aproximadamente 1000 vezes. Estes dois andares de amplificação são realizados utilizando amplificadores operacionais OPA627AP [42] e implementados na placa blindada. O primeiro andar de amplificação tem um potenciômetro de 100kΩ que permite ajustar a tensão de desvio (*offset*) e foi ajustado para uma tensão de desvio de 15mV na saída do segundo andar de amplificação, sendo este o valor de *offset* mínimo que se conseguiu obter.

À saída do segundo andar de amplificação a resposta do fio vibrante (sinusoide amortecida) possui uma amplitude máxima na ordem das dezenas/centenas de milivolts, pelo

que já é menos sensível ao ruído e é conduzida de volta para a placa não blindada onde vai sofrer a restante amplificação com o ganho selecionável. Esse ganho é selecionado através de relés que são ativados através das saídas da FPGA de acordo com as opções do utilizador. Como estes relés introduzem ruído, optou-se por colocá-los numa placa diferente da placa blindada (onde o sinal a amplificar tem uma amplitude de microvolts).

Através do cálculo da força eletromotriz induzida máxima aos terminais do fio vibrante (secção 1.3.3) podemos observar que a tensão máxima à entrada do primeiro andar de amplificação é da ordem das dezenas/centenas de microvolts e sabendo que a resposta do fio vibrante tem o formato de uma senoide amortecida, verifica-se que este valor máximo vai ser atenuado. Isto significa que a resposta do fio vibrante é detetável até que esta seja da mesma ordem de grandeza do ruído de entrada do amplificador do primeiro andar de amplificação.

Para determinar esse nível de ruído, utilizou-se o modelo típico que descreve a tensão de ruído e a corrente de ruído e as adiciona de modo a determinar a tensão equivalente de ruído de entrada do amplificador do primeiro andar de amplificação, da seguinte forma:

$$E_{ni} = \sqrt{e_{ni}^2 + I_{ni}^2 R_G^2 + 4KTR_G} \quad (5.15)$$

onde  $E_{ni}$  é a tensão de ruído de entrada do amplificador,  $e_{ni}^2$  é a tensão equivalente de ruído do amplificador,  $I_{ni}^2 R_G^2$  é a corrente equivalente de ruído do amplificador e  $4KTR_G$  exibe o ruído térmico produzido pelas resistências. Consultando [43], obtêm-se os seguintes valores:

$e_{ni}$ (10Hz) = 20nV/√Hz	$I_{ni}$ = 2.5fA/√Hz
$e_{ni}$ (100Hz) = 10nV/√Hz	$K$ = 1.38x10 <sup>-23</sup> joules/K
$e_{ni}$ (1kHz) = 5.6nV/√Hz	$T$ = 300°K = 27°C
$e_{ni}$ (10kHz) = 4.8nV/√Hz	

Como a corrente equivalente de ruído é  $I_{ni} = 2.5\text{fA}/\sqrt{\text{Hz}}$ , pode-se desprezar o efeito da corrente de ruído do amplificador.

Utilizando os valores anteriores constrói-se a Tabela 5.1 para auxiliar o cálculo da tensão de ruído de entrada do amplificador do primeiro andar de amplificação.

	10Hz	100Hz	1kHz	10kHz	100kHz
$e_{ni}^2$ (V <sup>2</sup> /Hz)	4.0x10 <sup>-16</sup>	1.0x10 <sup>-16</sup>	3.14x10 <sup>-17</sup>	2.30x10 <sup>-17</sup>	2.30x10 <sup>-17</sup>
4KTR <sub>G</sub> (V <sup>2</sup> /Hz)	1.08x10 <sup>-15</sup>	1.08x10 <sup>-15</sup>	1.08x10 <sup>-15</sup>	1.08x10 <sup>-15</sup>	1.08x10 <sup>-15</sup>
$E_{ni}^2$ (V <sup>2</sup> /Hz)	1.48x10 <sup>-15</sup>	1.18x10 <sup>-15</sup>	1.11x10 <sup>-15</sup>	1.10x10 <sup>-15</sup>	1.10x10 <sup>-15</sup>

Tabela 5.1 - Auxiliar de cálculo da tensão total de ruído de entrada do amplificador do andar de entrada do bloco de amplificação.

Como a largura de banda do circuito é 50kHz (ver secção 5.3.2) e o ganho do amplificador decresce 20dB por década, comportando-se como um filtro de primeira ordem [42], para saber a largura de banda efetiva do circuito multiplica-se os 50kHz por 1.57 [44] [45]. Como a largura de banda efetiva do circuito é 78.5kHz, para saber o ruído de entrada total no amplificador de entrada  $\sqrt{E_{ni}^2}$ , determina-se raiz quadrada da área do gráfico seguinte até aos 78.5kHz. Verifica-se assim que a tensão de ruído de entrada do primeiro andar de amplificação é  $\sqrt{E_{ni}^2} = 9.3\mu V_{RMS}$ .

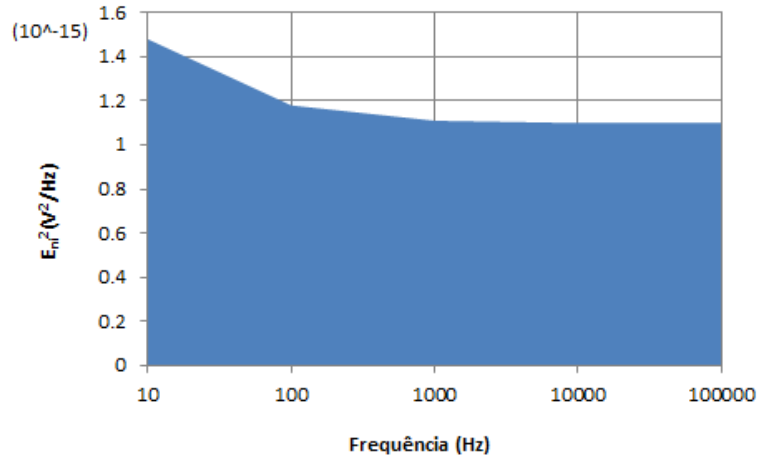


Figura 5.7 - Auxiliar de cálculo da tensão total de ruído de entrada do amplificador do andar de entrada do bloco de amplificação.

Sabendo ainda que o ganho em malha fechada deste andar de amplificação é 31.6, podemos saber a amplitude de ruído à saída do primeiro andar de amplificação fazendo  $\sqrt{E_{ni}^2} \times (\text{ganho do amplificador em malha fechada}) = 0.29mV_{RMS}$  que corresponde a  $1.76mV_{\text{pico-a-pico}}$  considerando  $6\sigma$ , ou seja, 99.7% de probabilidade da medida estar certa.

Uma vez que a amplitude de ruído à saída do primeiro andar de amplificação é  $1.76mV_{\text{pico-a-pico}}$  e o segundo andar de amplificação tem um ganho de 31.6, pode-se dizer que o ruído à saída do segundo andar de amplificação é cerca de  $55mV_{\text{pico-a-pico}}$ . Como a janela de medida é de 3.3V, dos quais  $55.6mV$  são ruído introduzido por estes dois andares de amplificação, significa que  $\frac{3.3}{0.055} = 60$  níveis de digitalização são ruído. Estes 60 níveis correspondem a 6 bits. Como cada amostra é codificada com 12 bits, conclui-se que 6 desses bits contêm ruído proveniente destes andares de amplificação.

Verifica-se através destes cálculos que o amplificador escolhido OPA627AP tem um desempenho adequado para a amplificação desejada, uma vez que o nível de ruído introduzido por este é aceitável tendo em conta a amplitude do sinal a amplificar.

# Capítulo 6

## 6 Testes e resultados

Neste capítulo vão ser apresentados os testes e os resultados que fazem a validação do funcionamento do sistema. Para tal, apresenta-se um conjunto de especificações do sistema e os respetivos resultados.

O sistema tem que:

- produzir um impulso simétrico com o formato de uma onda quadrada cuja duração e amplitude têm que ser controladas pelo utilizador através de uma interface gráfica.
- adquirir 1000 amostras com uma frequência de amostragem de 100kHz. O utilizador pode decidir o tempo total a amostrar, ou seja, poder escolher outra frequência de amostragem.
- após a aquisição do sinal, este deverá ser enviado para o computador onde uma interface gráfica recebe os dados e desenha um gráfico e permite guardar os pontos do gráfico e a imagem do gráfico.

Para testar e validar estes requisitos, optou-se por ligar o DAC diretamente ao ADC de modo a poder validar que a aquisição funciona como esperado e que o sinal produzido é o desejado.

O ambiente gráfico para o utilizador inserir a amplitude e a duração do impulso a estimular o fio vibrante é o da Figura 6.1.

Condicionalmento do impulso a excitar o fio vibrante

Duração do impulso (ms): (0.01 - 9.00) ms      Tempo de sinal a amostrar (ms):      Amplitude do impulso (V): (0.01 - 1.65) Volt

4.00      10      0.85

Escolha uma das resistências para controlar a corrente a passarno fio vibrante:

Resistência de 10 Ohm      Resistência de 100 Ohm

Frequência do impulso      Corrente do impulso

0.25 KHz      0.085 A

Se pretender fazer passar corrente na resistência de 0.1 Ohm carregue no botão seguinte.

Resistência de 0.1 Ohm

Escolha quantas vezes quer amplificar o sinal:

10<sup>3</sup>      2x10<sup>3</sup>      5x10<sup>3</sup>

10<sup>4</sup>      2x10<sup>4</sup>      5x10<sup>4</sup>

10<sup>5</sup>      2x10<sup>5</sup>      5x10<sup>5</sup>

10<sup>6</sup>

Iniciar

Figura 6.1 - Introdução das características do impulso a produzir.

A aquisição do impulso produzido tem o aspeto da Figura 6.2. À esquerda da figura pode-se observar a aquisição do impulso produzido quando o utilizador escolhe uma duração do impulso de 4ms e amplitude de 0.85V e seleciona a resistência de 10Ω (correspondente ao exemplo da figura anterior). A interface gráfica calcula a frequência do impulso em função da duração introduzida (neste caso  $f = \frac{1}{0.004} = 0.25kHz$ ) e a amplitude máxima da corrente a excitar o fio em função da amplitude do impulso em Volt e da resistência selecionada (neste caso  $I = \frac{0.85V}{10} = 0.085A$ ). À direita da figura seguinte pode-se observar a aquisição de um impulso produzido com duração de 2ms e amplitude de 1.65V.

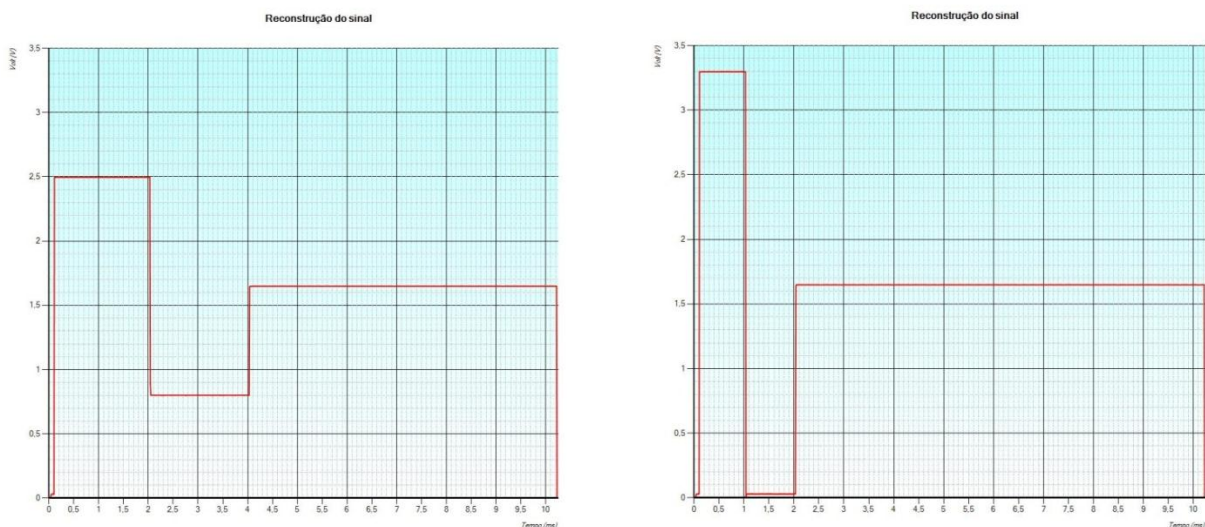


Figura 6.2 - Construção do gráfico com as amostras do impulso produzido.

Deste modo, testa-se a interface gráfica no controlo da produção do impulso e na receção dos dados da FPGA. Testa-se também o correto funcionamento da FPGA no controlo dos ADC e DAC, a guardar as amostras digitalizadas e na comunicação com o computador.

Após produzir o impulso é necessário que o relé S3 esteja a conduzir e que o relé S4 não esteja a conduzir (consultar secção 5.2). Para testar e validar que isto se verifica, substitui-se o fio vibrante por um circuito RLC que provoca uma oscilação depois de o impulso ser aplicado. Nesta fase ainda não se introduz amplificação da resposta ao impulso, pelo que o circuito RLC está ligado diretamente ao ADC.

Quando os dois relés estão sempre a conduzir e não existe este sincronismo, o resultado é o seguinte:

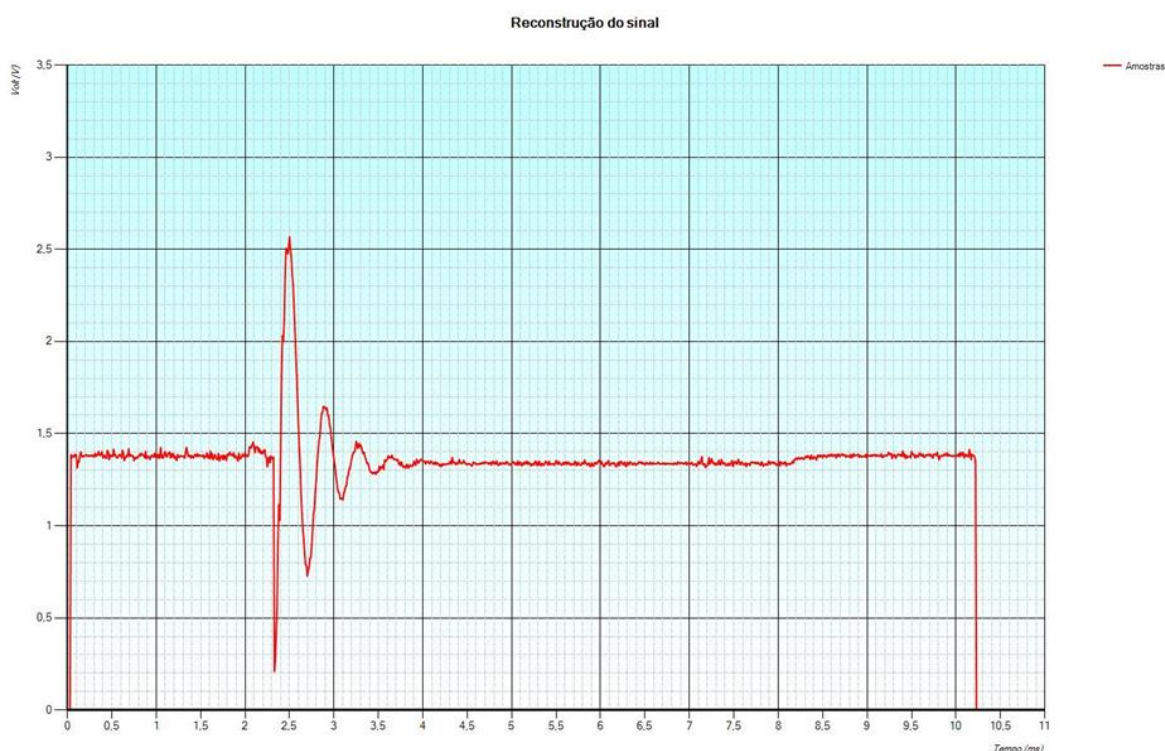


Figura 6.3 - Relés S3 e S4 sempre a conduzir, não sincronizados.

Neste teste o impulso (a excitar o circuito RLC) tem a duração de 2ms e uma amplitude de 3.3V pico a pico.

Quando o relé S3 conduz até ao final da duração do impulso e deixa de conduzir no final da duração do impulso, enquanto o relé S4 está sempre a conduzir, o resultado é o seguinte:

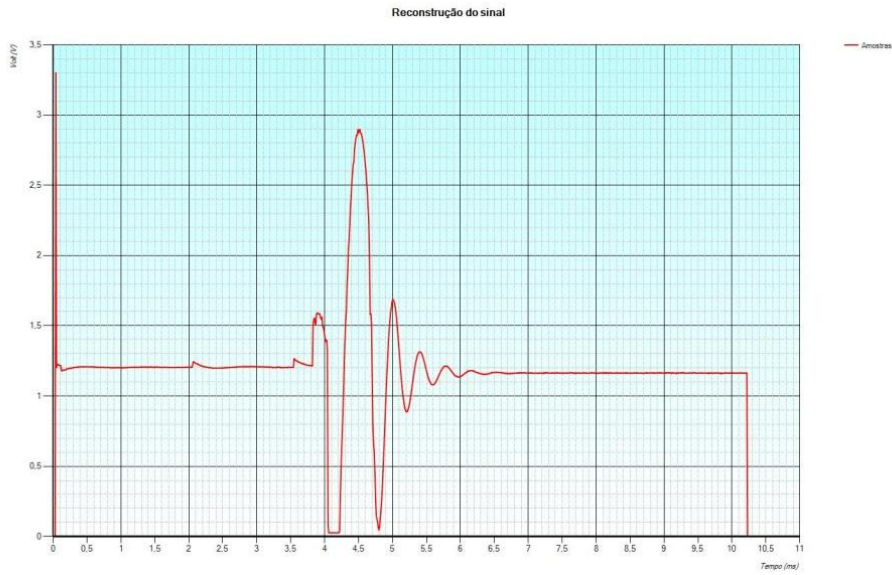


Figura 6.4 - Relé S4 sempre a conduzir e S3 só a conduzir durante a duração do impulso.

Neste teste, o impulso tem a duração de 4ms e uma amplitude de 3.3V pico a pico. Após o sincronismo dos relés S3 e S4, ou seja, durante a duração do impulso o relé S3 está a conduzir e o relé S4 não está a conduzir e após o final da duração do impulso o relé S3 deixa de conduzir e o relé S4 começa a conduzir. O resultado é o seguinte:

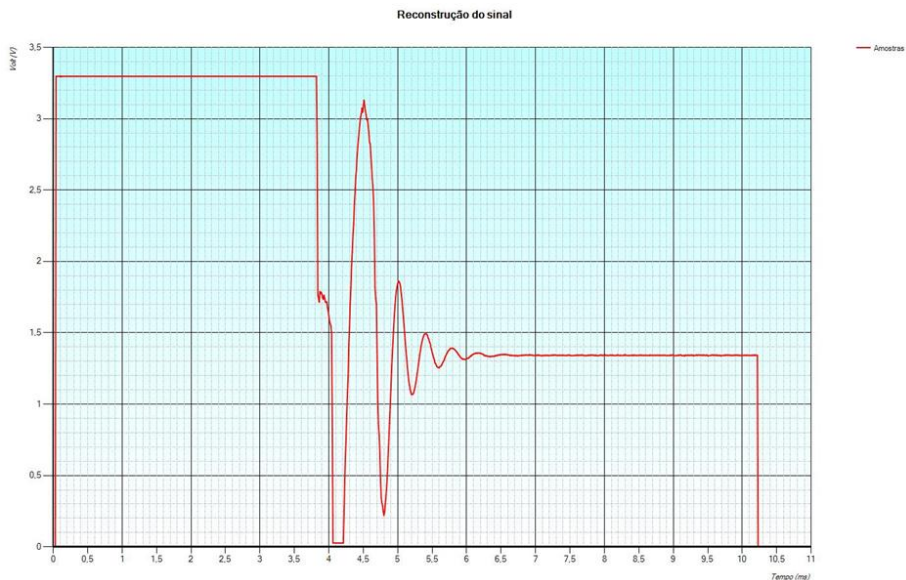


Figura 6.5 - Relés S4 e S3 sincronizados.

Como se pode observar, o relé S4 está ao corte durante a duração do impulso (neste caso são 4ms) pois o ADC satura quando não tem um sinal ligado à sua entrada. Após a duração do impulso o relé S4 conduz, como se pode observar na Figura 6.5. Depois de os relés

estarem síncronos, dá-se a ordem para se iniciar a aquisição do sinal só após o final da duração do impulso, ou seja, quando o relé *S4* começar a conduzir.

Para testar o funcionamento do deslocador de nível, efetuou-se uma simulação com recurso ao programa *PSpice* onde se introduziu a montagem Figura 5.4. Como se pode observar na Figura 6.6, a linha a verde representa o impulso quadrado centrado em 1.65V, com duração de 2ms e amplitude máxima de 1.65V à saída do DAC. A vermelho pode-se observar a saída do deslocador de nível (representado por *A* na Figura 5.2) que desloca o impulso 1.65V para o centrar em torno de 0V. A azul pode-se observar a saída do deslocador de nível (representado por *E* na Figura 5.3) que desloca o sinal de resposta do fio vibrante (mas nesta simulação desloca o impulso) novamente 1.65V para centrar o sinal em torno de 1.65V. Valida-se assim correto funcionamento dos deslocadores de nível.

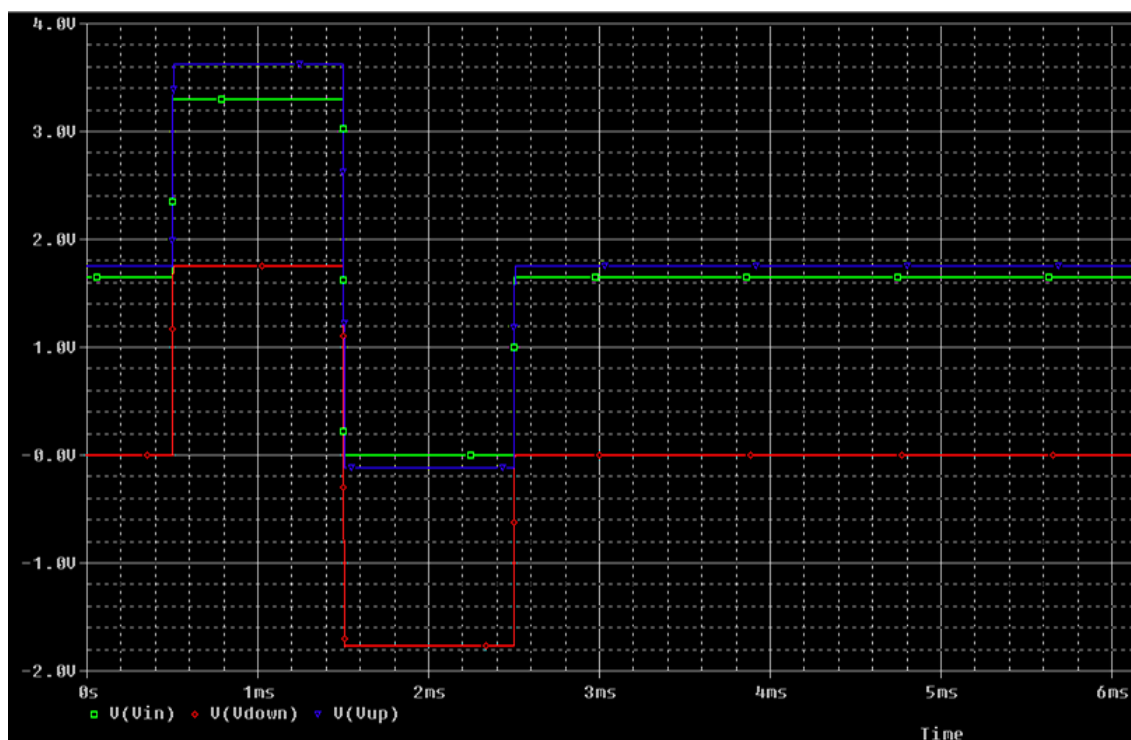


Figura 6.6 - Simulação em *PSpice* do comportamento do deslocador de nível.

Para se testar todo o sistema, faz-se a montagem da Figura 5.2 e Figura 5.3. Para adquirir a resposta do fio vibrante, substituindo-se o circuito RLC pelo fio vibrante e inclui-se o bloco de amplificação no sistema.

Depois do sistema fazer a aquisição da força eletromotriz induzida aos terminais do fio vibrante em ordem ao tempo recorreu-se ao programa *Table Curve 2D* para fazer o ajuste dos pontos experimentais ( $t$  e  $V(t)$ ) à seguinte equação que descreve o comportamento teórico da resposta do fio vibrante [3]:

$$V(t) = Ae^{-\Delta 2\pi ft} \cos(2\pi ft + \varphi) + V_1 \quad (6.1)$$

Antes de se introduzir a blindagem eletrostática, o sistema adquire a seguinte resposta do fio vibrante:

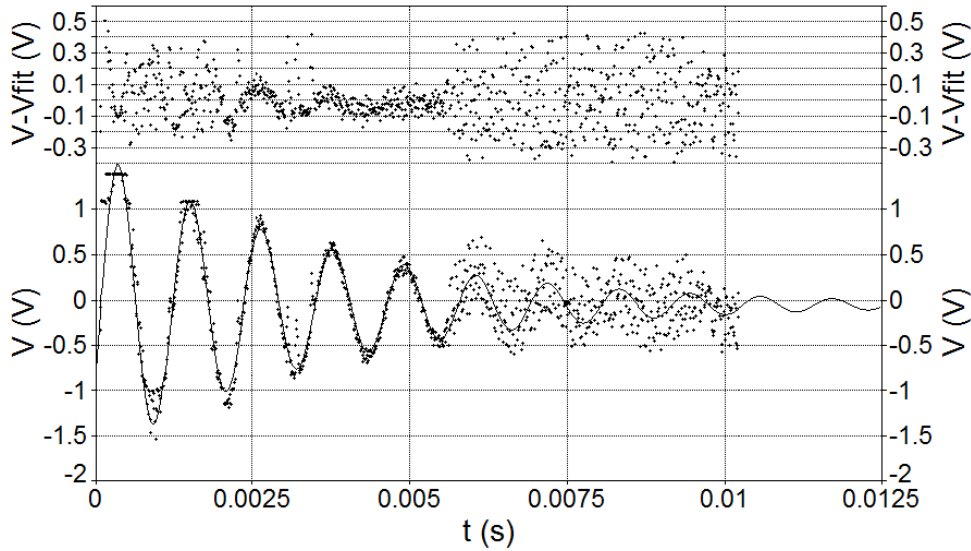


Figura 6.7 - Resposta do fio vibrante antes da blindagem eletrostática.

Observando a figura anterior verifica-se que a distribuição do ruído apresenta uma amplitude de cerca de  $0.9V_{pico-a-pico}$ . Depois de se introduzir a placa “mais sensível” ao ruído numa caixa metálica que funciona como uma blindagem eletrostática, obteve-se a seguinte resposta do fio vibrante onde se verifica que a distribuição do ruído apresenta uma amplitude de cerca de  $0.1V_{pico-a-pico}$ , ou seja, cerca de 8 vezes menor:

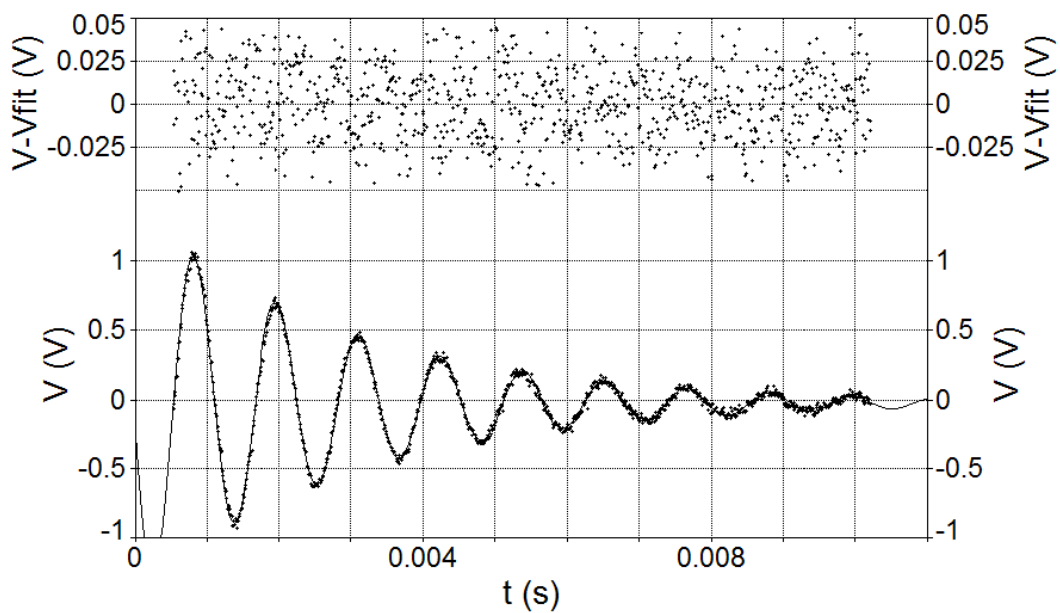


Figura 6.8 - 1ª Resposta do fio vibrante depois da blindagem eletrostática.

De seguida apresenta-se outra aquisição da resposta do fio vibrante utilizando a blindagem eletrostática e uma tabela resumo dos parâmetros da equação (6.1) após o ajustamento destas e de outras medidas presentes no Anexo E.

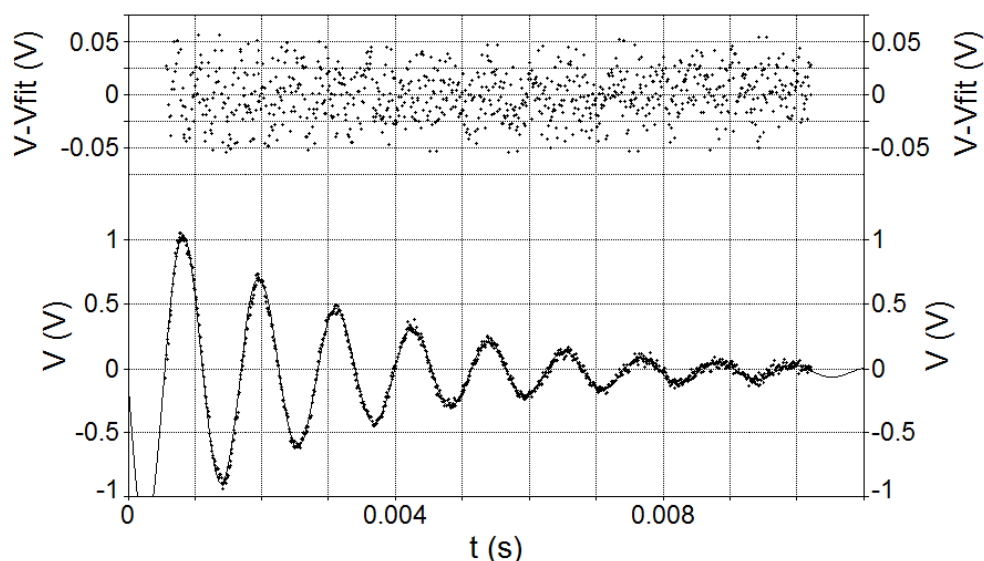


Figura 6.9 - 2ª Resposta do fio vibrante depois da blindagem eletrostática.

Tendo em conta que os parâmetros relevantes para o cálculo da viscosidade do líquido em estudo é a frequência e a decaimento da resposta do fio vibrante, calculou-se a média da frequência e do decaimento das medidas com blindagem e verifica-se que a  $f_{\text{média}} = 877.0985\text{Hz}$  com desvio padrão  $\sigma = 0.118\text{Hz}$  e  $\text{decaimento}_{\text{médio}} = 0.06075$  com desvio padrão  $\sigma = 5 \times 10^{-5}$ .

Ensaio	f (Hz)	$\Delta$ (decaimento)	$V_1$ (V)	A (V)	$\varphi$	$R^2$	Desvio de frequência (f-f <sub>médio</sub> )	Desvio do decaimento ( $\Delta$ - $\Delta_{\text{médio}}$ )
Sem blindagem	879.410	0.0499	-0.05	1.71	-0.44	0.910	2.3115	-0.01085
1º Com blindagem	876.972	0.0608	-0.02	1.38	-2.97	0.996	-0.1265	$-5 \times 10^{-5}$
2º Com blindagem	876.989	0.0607	-0.02	1.39	-3.05	0.994	-0.1095	$-5 \times 10^{-5}$
3º Com blindagem	877.210	0.0607	-0.02	-1.49	-1.40	0.995	0.1115	$-5 \times 10^{-5}$
4º Com blindagem	877.223	0.0608	-0.22	-1.48	-0.80	0.994	0.1245	$5 \times 10^{-5}$

Tabela 6.1 - Quadro resumo dos parâmetros da equação para medidas efetuadas.

Como se pode observar pela Tabela 6.1 e pela Figura 6.7, quando não existe blindagem eletrostática as medidas contêm um nível de ruído que não permite obter resultados utilizáveis porque, como se pode observar, o valor da frequência  $f$  da resposta do fio vibrante quando não existe blindagem varia na ordem cerca de duas unidades e o decaimento varia na terceira casa decimal das respetivas médias.

Quando se introduz a blindagem eletrostática sobre a placa “mais sensível” ao ruído o desvio padrão da frequência é cerca de  $0.1Hz$  e o desvio padrão do decaimento é cerca de  $5 \times 10^{-5}$ .

Para saber a relação sinal/ruído das medidas efetuadas calcula-se o valor eficaz do sinal (que é a resposta do fio vibrante) e o valor eficaz do ruído que é obtido curto-circuitando as entradas do andar de amplificação, ou seja, na ausência de sinal. Verifica-se que no pior caso, a sinal tem amplitude eficaz de  $S_{RMS} = 0.31921 (V)$  e o ruído tem amplitude eficaz de  $R_{RMS} = 0.02280 (V)$ . Verifica-se assim que no pior caso, a relação sinal/ruído é  $\frac{S}{R} = 22.9(dB)$ , ou seja, o sinal tem uma amplitude 14 vezes superior à amplitude do ruído.

A resposta do fio vibrante tem uma amplitude eficaz de  $0.32V_{RMS}$  que equivale a cerca de  $1.92V_{pico-a-pico}$  (seis vezes o valor eficaz) que é codificada em 12 bits, ou seja, 4096 níveis de quantificação. Por sua vez o ruído tem uma amplitude eficaz de  $0.0228V$  que equivale a cerca de  $0.13V_{pico-a-pico}$  que é corresponde a 161 dos 4096 níveis de quantificação, logo é codificado em 8 bits (Figura 6.10).

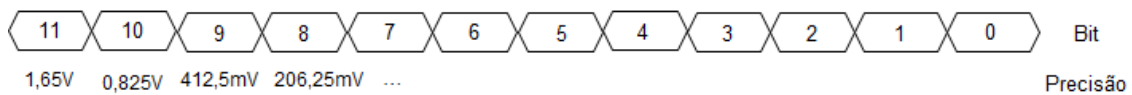


Figura 6.10 - “Precisão” dos bits de uma amostra.

Pode-se assim dizer que cada amostra tem uma precisão de 4 bits.

# Capítulo 7

## 7 Conclusões e trabalhos futuros

### 7.1 Conclusões

Tendo em consideração os objetivos e as especificações apresentadas na secção 1.3 pode-se dividir o trabalho em três grandes blocos: a interface gráfica, a implementação na FPGA do controlo do sistema e o *hardware* de condicionamento de sinal.

Uma vez que todas as especificações da interface gráfica foram cumpridas permitindo ao utilizador interagir com o sistema de modo pretendido, introduzindo a duração e amplitude do impulso a estimular o fio vibrante, introduzindo a frequência de amostragem, seleccionando o ganho de amplificação, comunicando corretamente com a FPGA, recebendo as amostras da resposta do fio vibrante e permitindo gravá-las num ficheiro externo, reconstruindo o gráfico do decaimento e permitindo guardá-lo, pode-se concluir que este bloco foi implementado com êxito.

Relativamente à implementação na FPGA do controlo do sistema pode-se concluir que as especificações foram cumpridas com êxito na medida em que é produzido o impulso a estimular o fio vibrante de acordo com as opções que o utilizador introduz na interface gráfica e a aquisição da resposta do fio vibrante está de acordo com as especificações do trabalho, ou seja, o número de amostras (adquire 1024 amostras sem interrupção), a frequência de amostragem (100kHz por defeito mas possível de ser alterado pelo utilizador na interface gráfica), a precisão de cada amostra (12 bits) e o sincronismo do sistema através de relés controlados pelas saídas digitais da FPGA, são os pretendidos.

Quanto ao *hardware* de condicionamento do sinal, relativamente à produção do impulso de corrente a estimular o fio vibrante as especificações foram cumpridas com êxito uma vez que o sistema permite produzir um impulso com amplitude eficaz máxima de  $I_{eficaz} = 0.825A$  que é cerca de oito vezes maior do que a amplitude de outros trabalhos realizados

anteriormente, permitindo assim utilizar fios vibrantes com um diâmetro maior e efetuar medições em líquidos mais viscosos.

No condicionamento da resposta do fio vibrante consegue-se obter uma relação sinal/ruído de cerca de  $23(dB)$  em que o sinal (resposta do fio vibrante) tem uma amplitude eficaz de  $0.32V_{RMS}$  que equivale a cerca de  $1.92V_{pico-a-pico}$  e o ruído tem uma amplitude eficaz de  $0.0228V_{RMS}$  que equivale a cerca de  $0.13V_{pico-a-pico}$ .

Comparando o ruído de aquisição antes de utilizar a blindagem eletrostática e depois de utilizar a blindagem eletrostática, pode-se concluir que a blindagem utilizada introduz uma melhoria significativa, reduzindo o ruído externo à montagem cerca de 8 vezes (ver secção 6). No entanto, apenas a placa “mais sensível” ao ruído estava no interior da blindagem eletrostática, ficando a outra placa, a FPGA e a fonte de alimentação das duas placas fora da blindagem e introduzindo ruído externo na montagem.

Se se realizasse uma blindagem eletrostática que incluísse a fonte de alimentação, a FPGA, a placa “menos sensível” ao ruído e a outra blindagem com a placa “mais sensível” ao ruído, o nível de ruído diminuiria garantidamente uma vez que se verificaram diferentes níveis de ruído em testes realizados durante o dia em que o laboratório onde os testes foram realizados continha máquinas em funcionamento (como motores, frigoríficos, exaustores, lâmpadas fluorescentes, computadores e outras máquinas normalmente utilizadas em laboratórios de química) e durante a noite em que grande parte destas máquinas estavam desligadas. Tendo em consideração esta variação dos níveis de ruído, teve-se a preocupação de se efetuar medidas num período intermédio em que nem estavam todas as máquinas a funcionar nem estavam todas desligadas, de modo a não apresentar resultados (níveis de ruído) que não correspondessem à realidade da futura utilização deste sistema.

Apesar do ruído observado nas medidas realizadas, pode-se concluir que os resultados obtidos são úteis para o cálculo da viscosidade, uma vez que o desvio máximo da frequência da resposta do fio vibrante é  $0.1Hz$  e do desvio máximo do decaimento é  $5 \times 10^{-5}$ . De salientar que as medidas foram feitas sem controlo de temperatura e sabendo que a viscosidade varia com a variação da temperatura era de esperar ligeiras variações nos valores da frequência e do decaimento da resposta do fio vibrante, sendo que os desvios aceitáveis são da ordem de grandeza dos verificados experimentalmente. Como tal, não foi possível calibrar o sistema e comparar com outras medidas realizadas com outros equipamentos.

Resumindo, o objetivo de realizar um protótipo funcional de todo o sistema foi cumprido com êxito na medida em que o sistema permite fazer o estímulo do fio vibrante e a aquisição da resposta do fio vibrante em regime livre, obtendo valores de frequência e amortecimento da resposta do fio vibrante com um desvio padrão de  $0.1Hz$  e  $5 \times 10^{-5}$  respetivamente (ver secção 6), permitindo com o auxílio destes valores calcular a viscosidade do líquido em estudo.

## 7.2 Contribuições originais

A principal contribuição original desta dissertação é o modo de controlo do estímulo e aquisição de sinal no contexto da medição de viscosidade utilizando o método do fio vibrante em regime livre, utilizando uma FPGA e uma interface gráfica.

A interface gráfica com o utilizador feita “à medida” e o controlo do sistema com recurso a uma FPGA são uma inovação no contexto da medição de viscosidade com o método do fio vibrante em regime livre.

A amplitude eficaz máxima de corrente a estimular o fio vibrante  $I_{eficaz} = 0.825A$  também é uma contribuição original fruto das opções tomadas, permitindo assim utilizar fios vibrantes com um diâmetro maior e efetuar medições em líquidos muito mais viscosos.

## 7.3 Propostas de trabalhos futuros

Uma proposta de trabalho futuro é o aperfeiçoamento da blindagem eletrostática da placa “mais sensível” ao ruído e a realização de outra blindagem que incorpore todo o sistema, permitindo deste modo diminuir o nível de ruído.

Outra proposta é a implementação das duas placas de condicionamento do sinal em circuito impresso ou PCB e não em Veroboard como foram implementadas, permitindo também reduzir o ruído da montagem e o espaço ocupado pelas placas.

Por fim, será de interesse avaliar a necessidade de incorporar novas funcionalidades na interface gráfica como a possibilidade de realização de medidas automáticas a horas programadas e de modo sucessivo com intervalos entre medidas programáveis.

# Referências

- [1] F. J. Caetano, J. L. Mata, J. M. Fareleira, C. M. Oliveira e W. A. Wakeham, "Viscosity Measurements of Liquid Toluene at Low Temperatures Using a Dual Vibrating-Wire Technique," *International Journal of Thermophysics*, vol. 25, pp. 1-11, 2004.
- [2] F. M. Janeiro, J. M. Fareleira, J. C. Diogo, D. R. Máximo, P. M. Ramos e F. J. Caetano, "Impedance spectroscopy of a vibrating wire for viscosity measurements," em *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE*, Évora, 2010.
- [3] J. L. Mata, F. J. Caetano, C. M. Oliveira e J. M. Fareleira, "Viscosity Measurements of Diisodecyl Phthalate Using a Vibrating Wire Instrument Operated In Free Decay Mode: Comparison with Results Obtained with the Forced Mode of Operation," *Journal of Chemical & Engineering Data*, 2009.
- [4] J. L. Mata, "Desenvolvimento de técnicas e instrumentação para medidas simultâneas de densidade e viscosidade de líquidos," Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal, 2004.
- [5] T. A. Alvesalo, C. N. Archie, C. N. Albrecht, J. D. Reppy e R. C. Richardson, "Viscosity Measurements In Superfluid  $^3\text{He-B}$  TO 1 mK," *Journal de Physique*, vol. 39, pp. C6-41 - C6-42, 1978.
- [6] T. L. Pereira, "Reologia Aplicada nas Indústrias: Farmacêutica, Alimentícia, Cosmética e Química," 10 5 2005. [Online]. Available: <http://amigonerd.net/trabalho/22894-reologia-aplicada-nas-industrias-farmaceutica>. [Acedido em 2012 8 1].
- [7] H. M. Avelino, J. M. Fareleira e W. A. Wakeham, "Simultaneous Measurement of the Density and Viscosity of Compressed Liquid Toluene," *International Journal of Thermophysics*, vol. 24, pp. 323-336, 2003.
- [8] B. Flávio, R. Higuti, C. M. Furukawa, E. C. Silva e J. C. Adamowski, "Medição de viscosidade de líquidos pelo método de múltiplas reflexões acústicas com conversão de modo," *Sba: Controle & Automação Sociedade Brasileira de Automatica*, vol. 14, 2003.
- [9] F. Buiochi, J. C. Adamowski e C. M. Furukawa, "Measurement of viscosity using wave mode conversion," *Ultrasonics Symposium, 1998. Proceedings., 1998 IEEE*, vol. 2, pp. 1193-1196, 1998.

- [10] J. P. Trusler, "Viscosity and Density Measurements of Compressed Liquids," Department of Chemical Engineering, Imperial College London, South Kensington Campus, , London, U.K., 2007.
- [11] "Microcontroller to Sensor Interfacing Techniques," BiPOM Electronics, Inc., Missouri City, Texas, 2006.
- [12] D. Singh, "Micro-controller for Sensor Networks," Department of Computer Science and Endeneering, Indian Institute of Technology, India, 2008.
- [13] M. Morsi, "A Microcontroller Based on Multi Sensors Data Fusion and Artificial Intelligent Technique for Gas Identification," em *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE, Arab Acad. for Sci. & Technol.*, Alexandria, 2007.
- [14] S. S. Ramprasad e V. S. Nandanwar, "Design and Implementation of Performance Monitoring of DC Motor Based on Microcontroller and Temperature Control of Heat Sink," em *International Conference on Advanced Computing, Communication and Networks'11* , Nagpur, India , 2011.
- [15] Avinash , "eXtreme Electronics," Microchip PIC Tutorials, 25 07 2010. [Online]. Available: <http://extremeelectronics.co.in/microchip-pic-tutorials/interfacing-lm35-temperature-sensor-with-pic-microcontroller/>. [Acedido em 2012 08 01].
- [16] T. K. Bhaumik e D. Sarkar, "A Microcontroller Based Ultrasonic Range Finding Module for Target Detection from a Mobile Robot," Variable Energy Cyclotron Centre, 2009.
- [17] A. M. Costa, "Projecto de um Controlador para um Viscosímetro de Fio Vibrante," Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal, 2009.
- [18] G. Chalivendra, R. Srinivasan e N. S. Murthy, "FPGA based re-configurable wireless sensor network protocol," em *Electronic Design, 2008. ICED 2008. International Conference*, Arizona State Univ., Tempe, AZ, 2008.
- [19] W. Yan, A. Bermak e F. Boussaid, "FPGA implementation of compressive sampling for sensor network applications," em *Quality Electronic Design (ASQED), 2010 2nd Asia Symposium*, ECE Dept., Hong Kong Univ. of Sci. & Technol., Hong Kong, China, 2010.
- [20] V. Vikrant e R. Mehra, "FPGA Based Kalman Filter for Wireless Sensor Networks," *International Journal of Computer Technology and Applications* , vol. 2, pp. 155-159, 2011.
- [21] D. Reichel, A. Druilhe e T. Dang, "An Innovative Wireless Sensor Network Protocol Implementation using a Hybrid FPGA Technology," em *World Academy of Science, Engineering and Technology*, 2011.
- [22] S. Velusamy, W. Huang, J. Lach, M. Stan e K. Skadron, "Monitoring temperature in FPGA based SoCs," em *Computer Design: VLSI in Computers and Processors, 2005. ICCD 2005. Proceedings. 2005 IEEE International Conference* , Departments of Comput. Sci., Virginia

Univ., USA, 2005.

- [23] M. S. SIMIN, "Weather Station Monitoring System Using FPGA," Universiti Kuala Lumpur British Malaysian Institute, 2011.
- [24] K. Ohtani, M. Baba e D. Notohara, "An intelligent position sensor for light spots using an analog scan circuit and fpga," em *Instrumentation and Measurement Technology Conference, 2003. IMTC '03. Proceedings of the 20th IEEE*, Japan, 2003.
- [25] B. MITSURU, "Intellectualized optical position sensor using FPGA.," *EMTAF Kenkyu Jyosei Seika Hokokusho Heisei 13 Nendo (Dai14ki) Josei Jigyobun*, vol. 14, 2004.
- [26] C. R. Donate, L. M. Velazquez, R. A. Rios, G. H. Ruiz e R. J. Troncoso, "MDPI - Open Access Journals," 26 04 2010. [Online]. Available: <http://www.mdpi.com/1424-8220/10/4/4114>. [Acedido em 2012 08 01].
- [27] B. Zeidman, "The Death of the Structured ASIC," *Chip Design Magazine*, 2006.
- [28] Electronics Engineering Herald, "EE Herald," 2006. [Online]. Available: [http://www.eeherald.com/section/design-guide/fpga\\_design.html](http://www.eeherald.com/section/design-guide/fpga_design.html). [Acedido em 01 08 2012].
- [29] Xilinx Inc., "Spartan-3 FPGA Family: Introduction and Ordering Information," 13 07 2004. [Online]. Available: <http://duteela.et.tudelft.nl/~elca/XilinxTutor/documents/ds099-1.pdf>. [Acedido em 01 08 2012].
- [30] Digilent Inc., "digilentinc/Products: Spartan-3 Board," Digilent Inc., 13 05 2005. [Online]. Available: <http://digilentinc.com/Products/Detail.cfm?Prod=S3BOARD>. [Acedido em 01 08 2012].
- [31] Xilinx Inc., "Comparing and Contrasting FPGA and Microprocessor System Design and Development," 21 07 2004. [Online]. Available: [http://www.xilinx.com/support/documentation/white\\_papers/wp213.pdf](http://www.xilinx.com/support/documentation/white_papers/wp213.pdf). [Acedido em 01 08 2012].
- [32] Future Technology Devices International Ltd. , "Virtual COM Port Drivers," FTDI, 2012. [Online]. Available: <http://www.ftdichip.com/Drivers/VCP.htm>. [Acedido em 01 08 2012].
- [33] Digilent Inc., "Digilent PmodAD1 Analog to Digital Module Converter Board Reference Manual," Digilent Inc., 06 12 2011. [Online]. Available: <http://www.digilentinc.com/Products/Detail.cfm?Prod=PMOD-AD1>. [Acedido em 01 08 2012].
- [34] Analog Devices Inc., "AD7476A / AD7477A / AD7478A," 2011. [Online]. Available: [www.analog.com](http://www.analog.com). [Acedido em 01 08 2012].

- [35] Ubicom Inc., “Serial Peripheral Interface (SPI) and Microwire/Plus implementation Using the SX Communications Controller,” 11 2000. [Online]. Available: <http://www.parallax.com/Portals/0/Downloads/appnt/sx/An20Spi.pdf>. [Acedido em 01 08 2012].
- [36] R. Dubey, Introduction to Embedded System Design Using Field Programmable Gate Arrays, Índia: Springer, 2009.
- [37] Digilent Inc., “Digilent PmodDA2 Digital to Analog Module Converter Board Reference Manual,” Digilent Inc., 25 09 2006. [Online]. Available: <https://www.digilentinc.com/Products/Detail.cfm?NavPath=2,401,487&Prod=PMOD-DA2>. [Acedido em 01 08 2012].
- [38] National Semiconductor Corporation, “DAC121S101/DAC121S101Q,” 19 02 2010. [Online]. Available: <https://www.national.com/ds/DA/DAC121S101.pdf>. [Acedido em 01 08 2012].
- [39] Semiconductor Components Industries, “Quad Differential Input, Low Power Operational Amplifiers - LM324,” Semiconductor Components Industries, 2001.
- [40] SGS-Thomson Microelectronics, “3A Power Operational Amplifier - L165V,” SGS-Thomson Microelectronics, 1991.
- [41] Burr-Brown Corporation, “Precision Lowest Cost Isolation Amplifier - ISO122P,” Burr-Brown Corporation, U.S.A., 1993.
- [42] Burr-Brown Corporation, “Precision High-Speed Difet Operational Amplifiers - OPA627AP,” Burr-Brown Corporation, U.S.A., 1998.
- [43] Burr-Brown Corporation, “Precision High-Speed Difet Operational Amplifiers - OPA627AP,” U.S.A, Burr-Brown Corporation, 1998, p. 2.
- [44] Intersil Corporation, “Operational Amplifier Noise Prediction,” Intersil Application Note 519, 1996.
- [45] Intersil Corporation, “Instrumental Amplifier - AN 1298,” Intersil, 2009.

# ANEXOS

## Anexo A

Este anexo apresenta um exemplo de como a interface gráfica processa a introdução da duração do impulso pelo utilizador, para a converter em ciclos de relógio do circuito *DA\_controller* da FPGA.

Por exemplo, se o utilizador introduzir:

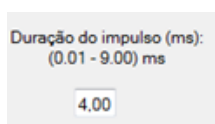


Figura A.1 - Duração do impulso introduzida pelo utilizador.

O cálculo efetuado seria:

- $\frac{4.00}{80 \times 10^{-6}} = 50000$  ciclos de produção de impulso analógico.
- $Round(50000) = 50000$
- $50000_{(10)} = C350_{(16)}$
- 0000C350
- 053C0000 é o número que chega à RAM implementada na FPGA e que indica o final da duração do impulso analógico.

De seguida divide-se a duração introduzida e efetua-se o mesmo raciocínio mas agora para indicar onde é metade da duração do impulso analógico a produzir. Continuando o exemplo anterior, o cálculo seria:

- $\frac{4.00}{2} = 2.00$
- $\frac{1.00}{80 \times 10^{-6}} = 25000$  ciclos de produção de impulso analógico.
- $Round(25000) = 25000$
- $25000_{(10)} = 61A8_{(16)}$
- 000061A8
- 8A160000 é o número que chega à RAM implementada na FPGA e que indica onde é metade da duração do impulso analógico.

## Anexo B

Este anexo apresenta um exemplo de como a interface gráfica processa a introdução da amplitude do impulso pelo utilizador, para a converter na escala utilizada pelo circuito *DA\_controller* da FPGA.

Por exemplo, se o utilizador introduzir:



Figura B.1 - Amplitude do impulso introduzida pelo utilizador.

O cálculo efetuado seria:

- $0.85 + 1.65 = 2.5V$  de amplitude máxima.
- $\frac{2.3 \times 4095}{3.3} = 3102.27$  nível de conversão, que tem que ser arredondado.
- $Round(3102.27) = 3102$  nível de conversão.
- $3102_{(10)} = C1E_{(16)}$
- $0C1E$
- $E1C0$  é o número que chega à RAM implementada na FPGA e que indica a amplitude máxima do impulso analógico.

Para saber a amplitude mínima, subtrai-se a amplitude introduzida a 1.65 V e de seguida segue-se o mesmo raciocínio. Continuando o exemplo anterior, o cálculo seria:

- $1.65 - 0.85 = 0.8$  Volt de amplitude mínima.
- $\frac{0.8 \times 4095}{3.3} = 1240.90$  nível de conversão, que tem que ser arredondado.
- $Round(992.72) = 992$  nível de conversão.
- $992_{(10)} = 3E0_{(16)}$
- $03E0$
- $0E30$  é o número que chega à RAM implementada na FPGA e que indica a amplitude mínima do impulso analógico.

## Anexo C

Este anexo apresenta diagramas temporais onde se pode observar a simulação da aquisição de uma resposta do fio vibrante, onde o bit de entrada proveniente da aproximação sucessiva é o *sdata1*, verifica-se que *cs* = 0 durante os 16 ciclos do estado *shiftin* e *done* = 1 durante o estado *Idle*.

A figura seguinte demonstra o diagrama temporal de aquisição da 1ª amostra.

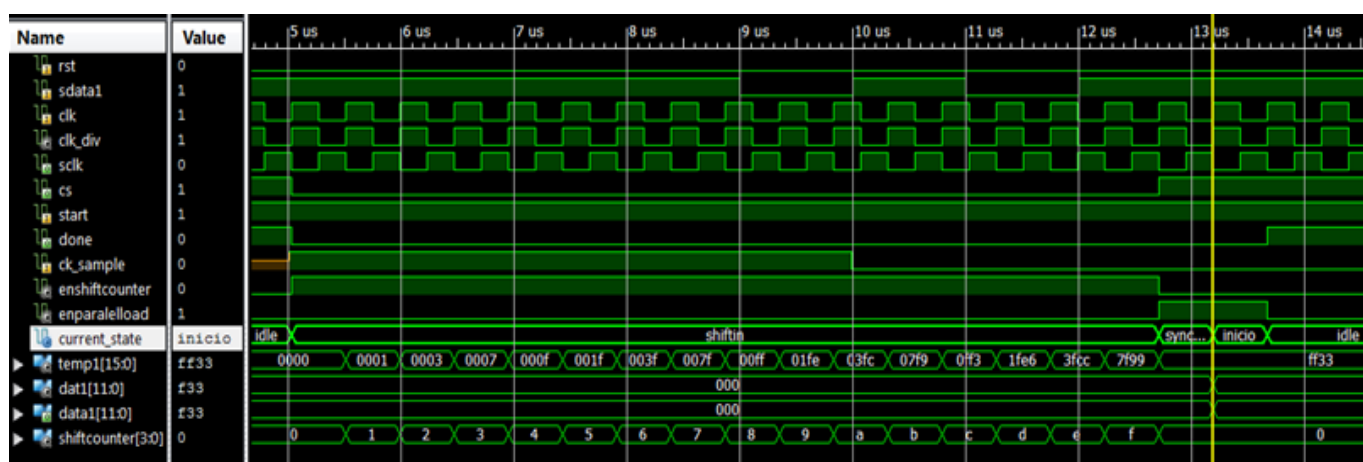


Figura C.1 - Diagrama temporal de aquisição da 1ª amostra.

A tabela seguinte resume a aquisição da 1ª amostra de acordo com a figura anterior.

sdata1	shiftCounter	Temp1 (em binário)	Temp1 (em hexadecimal)
	0	0000 0000 0000 0000	0000
1	1	0000 0000 0000 0001	0001
1	2	0000 0000 0000 0011	0003
1	3	0000 0000 0000 0111	0007
1	4	0000 0000 0000 1111	000F
1	5	0000 0000 0001 1111	001F
1	6	0000 0000 0011 1111	003F
1	7	0000 0000 0111 1111	007F
1	8	0000 0000 1111 1111	00FF
0	9	0000 0001 1111 1110	01FE
0	10	0000 0011 1111 1100	03FC
1	11	0000 0111 1111 1001	07F9

1	12	0000 1111 1111 0011	0FF3
0	13	0001 1111 1110 0110	1FE6
0	14	0011 1111 1100 1100	3FCC
1	15	0111 1111 1001 1001	7F99
1	16	1111 1111 0011 0011	<b>FF33</b>

Tabela C.1 - Sequência exemplo de aquisição da 1.<sup>a</sup> amostra.

Verifica-se que neste caso, o registo temporário  $tem1 = X"FF33"$ . Tendo em conta que a amostra são os 12 bits menos significativos de  $tem1$ , a 1.<sup>a</sup> amostra que sai do circuito  $AD1\_controller$  é  $DATA1 = X"F33"$ .

A seguinte demonstra o diagrama temporal de aquisição da 2.<sup>a</sup> amostra.

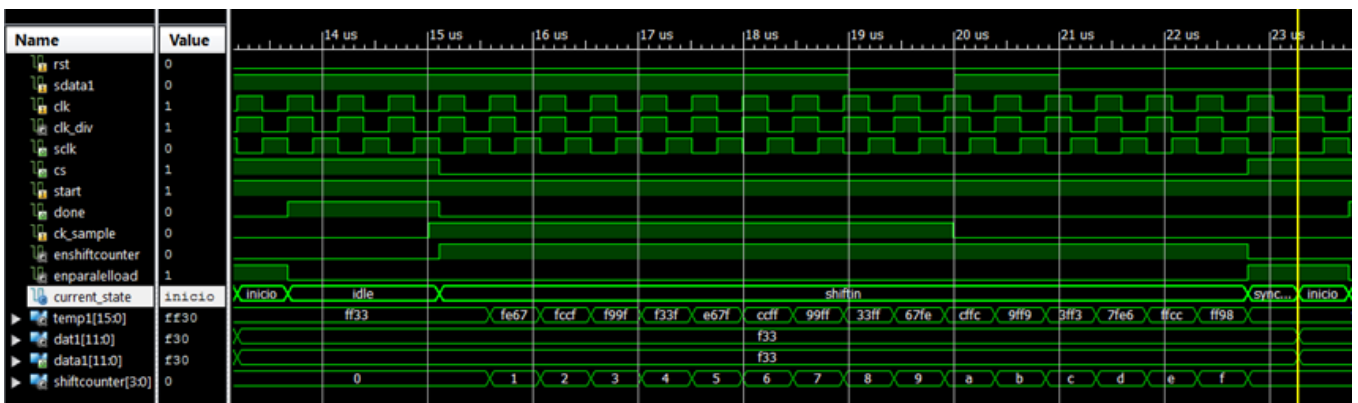


Figura C.2 - Diagrama temporal de aquisição da 2.<sup>a</sup> amostra.

A tabela seguinte resume a aquisição da 2.<sup>a</sup> amostra de acordo com a figura anterior.

sdata1	shiftCounter	Temp1 (em binário)	Temp1 (em hexadecimal)
1	0	1111 1111 0011 0011	FF33
1	1	1111 1110 0110 0111	FE67
1	2	1111 1100 1100 1111	FCCF
1	3	1111 1001 1001 1111	F99F
1	4	1111 0011 0011 1111	F33F
1	5	1110 0110 0111 1111	E67F
1	6	1100 1100 1111 1111	CCFF
1	7	1001 1001 1111 1111	99FF
1	8	0011 0011 1111 1111	33FF

0	9	0110 0111 1111 1110	67FE
0	10	1100 1111 1111 1100	CFFC
1	11	1001 1111 1111 1001	9FF9
1	12	0011 1111 1111 0011	3FF3
0	13	0111 1111 1110 0110	7FE6
0	14	1111 1111 1100 1100	FFCC
0	15	1111 1111 1001 1000	FF98
0	16	1111 1111 0011 0000	<b>FF30</b>

Tabela C.2 - Sequência exemplo de aquisição da 2.<sup>a</sup> amostra.

Verifica-se que neste caso, o registo temporário  $tem1 = X''FF30''$ . Tendo em conta que a amostra são os 12 bits menos significativos de  $tem1$ , a 2.<sup>a</sup> amostra que sai do circuito  $AD1\_controller$  é  $DATA1 = X''F30''$ .

## Anexo D

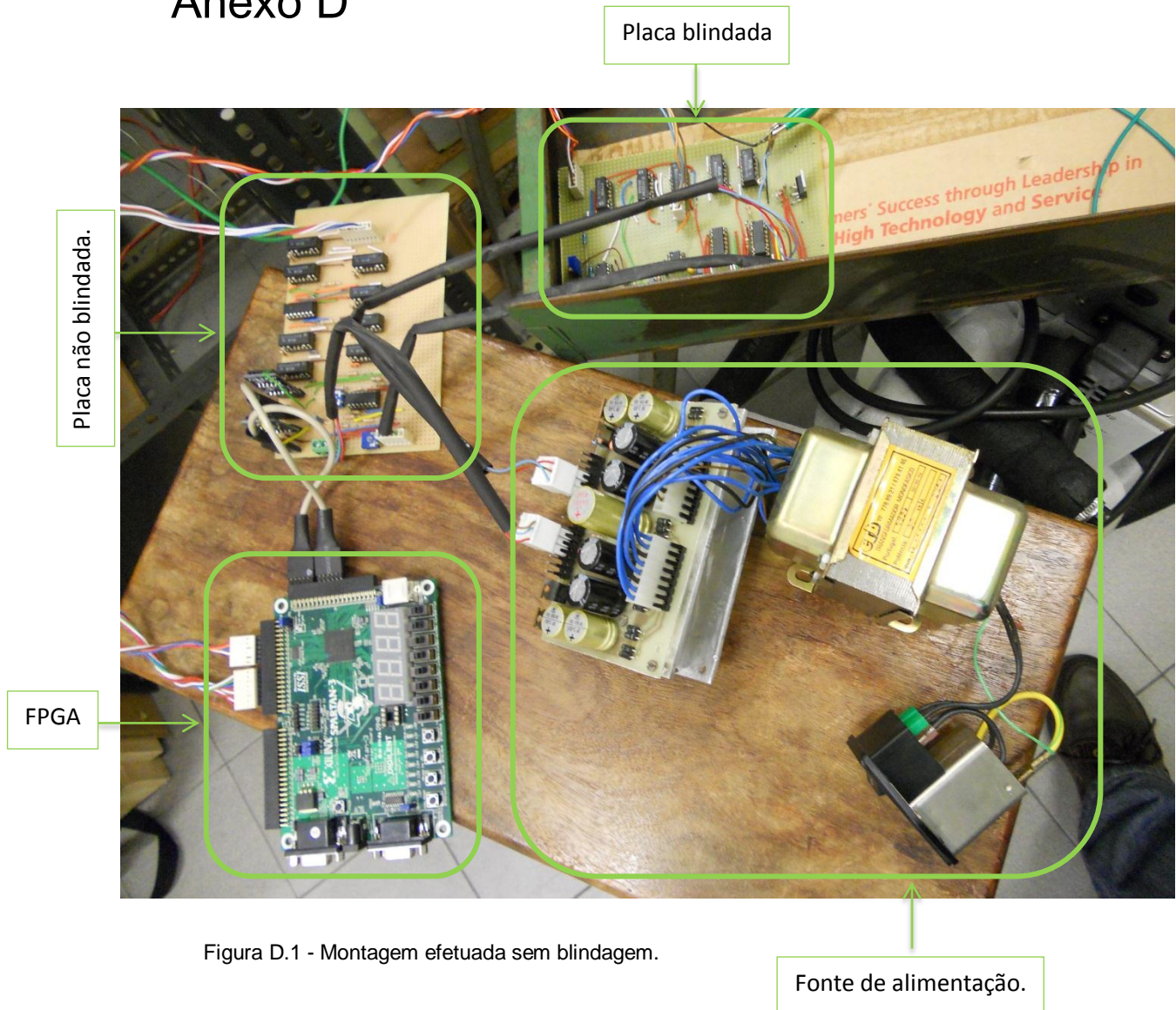


Figura D.1 - Montagem efetuada sem blindagem.

Na figura D.1 pode-se observar a montagem efetuada sem blindagem. Destacaram-se os diferentes grupos, nomeadamente a fonte de alimentação que alimenta separadamente a placa blindada e a placa não blindada, a placa blindada que se encontra no interior de uma caixa metálica mas sem a tampa e está ligada à placa não blindada, a placa não blindada que se encontra ligada à placa blindada e à FPGA e a FPGA que deveria estar ligada ao computador. O fio vibrante é ligado à placa blindada.

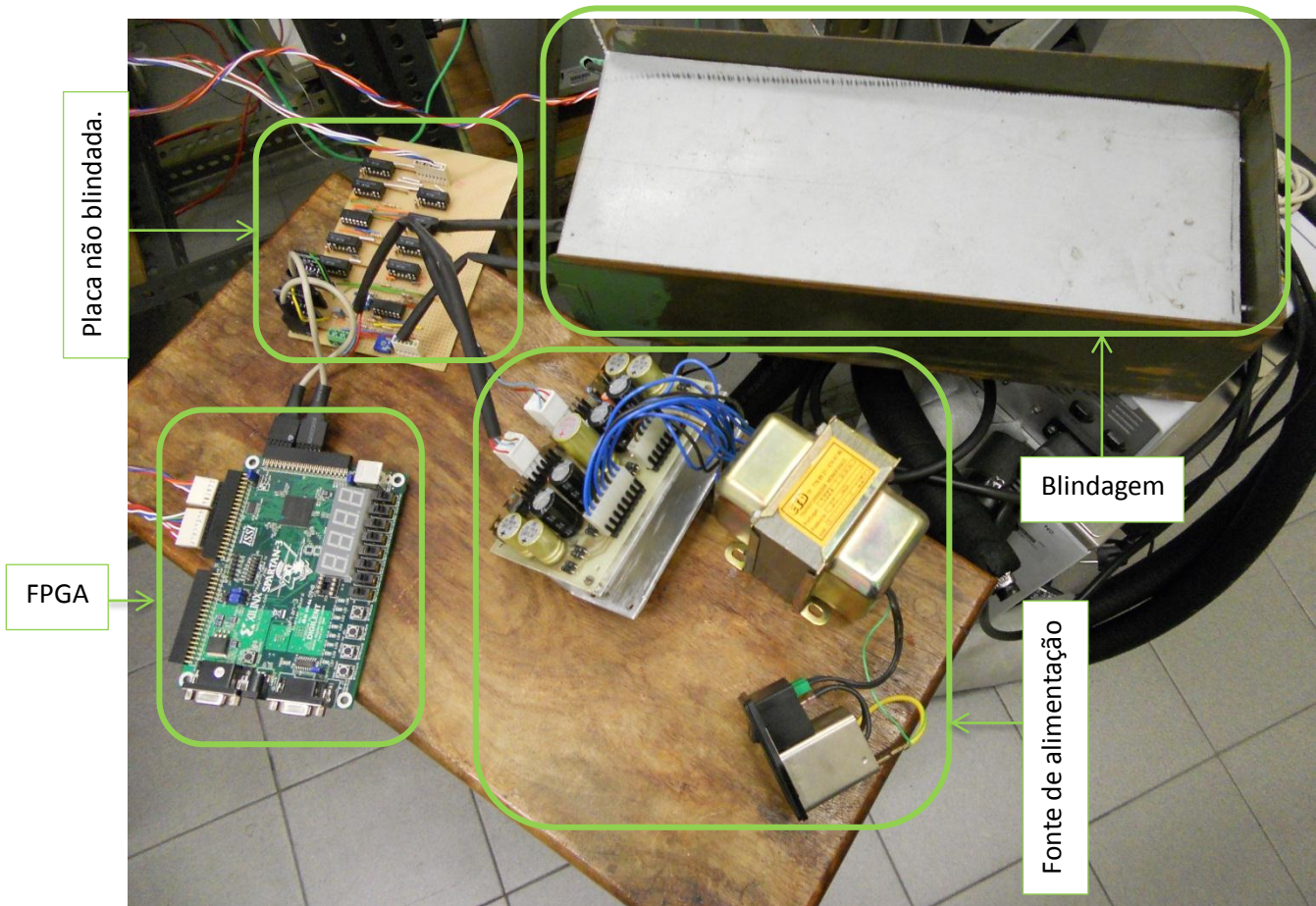


Figura D.2 - Montagem efetuada com a placa “mais sensível” blindada.

Na figura D.2 pode-se observar a montagem efetuada com a placa “mais sensível” ao ruído já blindada. Pode-se observar que a blindagem utilizada pode ser melhorada. Verifica-se ainda que o nível de ruído pode ser diminuído se se colocar a fonte de alimentação, a FPGA, a placa não blindada e a blindagem (com a placa blindada) dentro de uma caixa metálica construída especificamente para o efeito.

## Anexo E

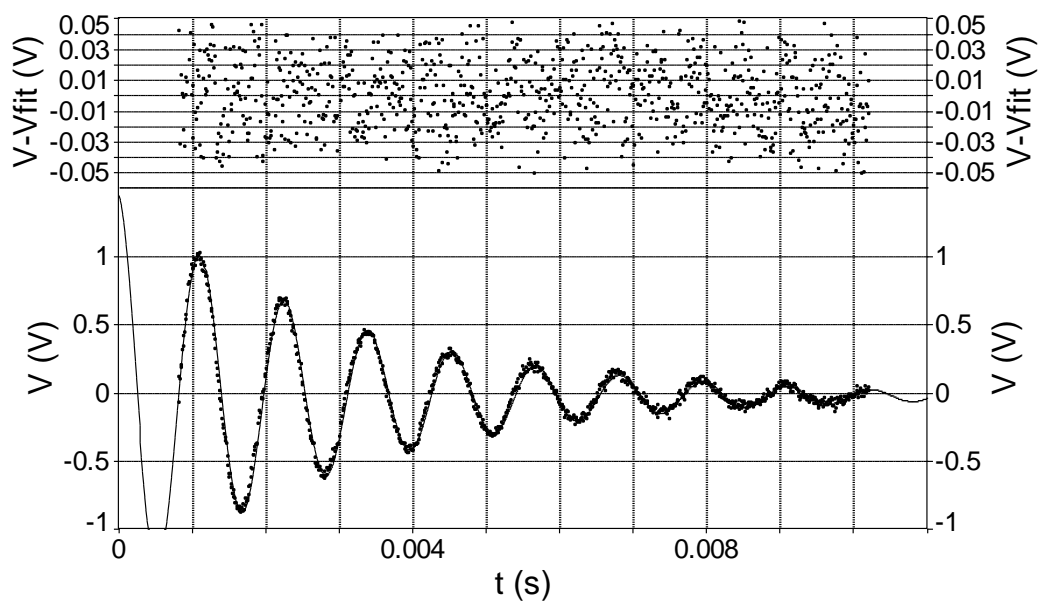


Figura E.1 - 3ª Resposta do fio vibrante depois da blindagem eletrostática.

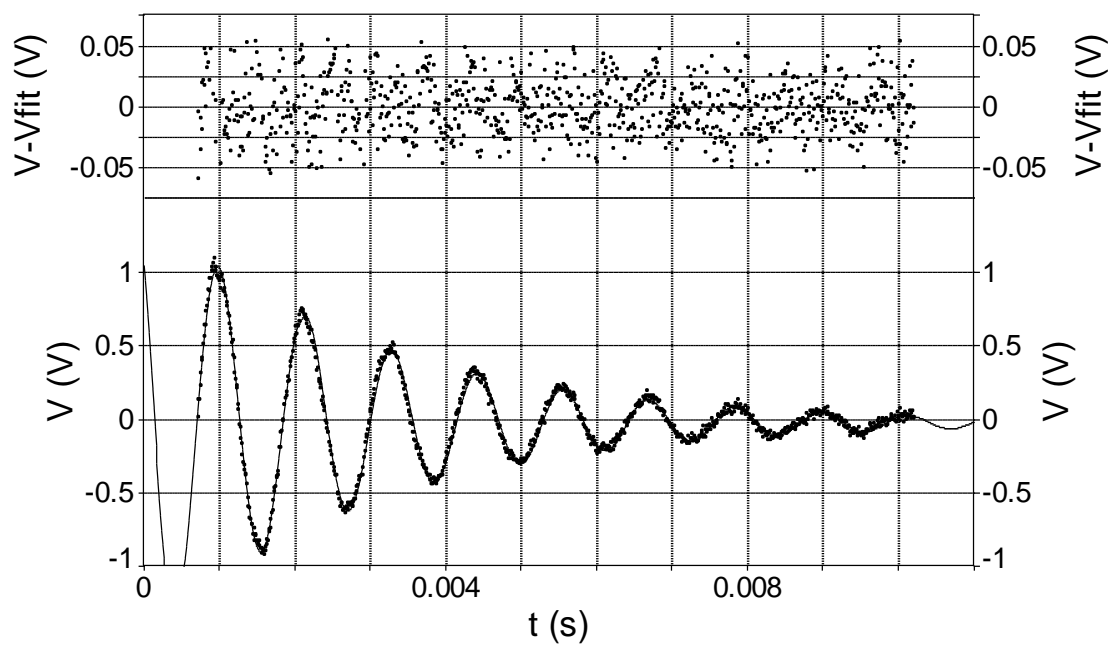


Figura E.2 - 4ª Resposta do fio vibrante depois da blindagem eletrostática.