



RECUPERAÇÃO DE SINAIS GERADOS VIA AMOSTRAGEM COMPRESSIVA EM FPGA

Oeslle Alexandre Soares de Lucena¹, Edmar Candeia Gurjão²

RESUMO

A digitalização de dados é largamente utilizada para o uso em dispositivos eletrônicos em diversas áreas, tais como: sistemas de radio recepção, processamento de imagens médicas, entre outros. Seu uso é baseado na conversão de sinais analógicos para sinais digitais seguindo a regra apresentada pelo Teorema de Nyquist. De forma peculiar, a técnica da Amostragem Compressiva propõe representações mais compactas do sinal a ser amostrado promovendo recuperação deste por meio de medições feitas no sinal original, contrariando os métodos tradicionais. Para isto, o conceito de esparsidade é levado em consideração, restringindo a aplicação da técnica. Neste projeto, foi implementado em hardware um dos blocos, para recuperação gerados via Amostragem Compressiva usando o algoritmo *Orthogonal Matching Pursuit*.

Palavras-chave: Amostragem Compressiva, Esparsidade, Implementação

RECOVERING SIGNALS GENERATED FROM COMPRESSES SENSING TECHNIQUE IN FPGA

ABSTRACT

Data digitalization is used in electronic devices in many different fields, such as radio systems, medical image processing, etc. Digitalization converts analog signals into digital signals based on the Nyquist theorem. Unusually, the Compressed Sensing technique comes compressing signal representations and recovering the original signal using fewer samples than the traditional methods. In this project, it was implemented a hardware implementation of one of the blocks responsible to recover signals generated through the compressed sensing technique using the algorithm called *Orthogonal Matching Pursuit*.

Keywords: Compressed Sensing, Sparsity, Implementation

¹Aluno do Curso de Engenharia Elétrica, Departamento de Engenharia Elétrica, UFPG, Campina Grande, PB, e-mail: oeslle.lucena@ee.ufcg.edu.br

²Engenharia Elétrica, Professor. Doutor, Departamento de Engenharia Elétrica, UFPG, Campina Grande, PB, e-mail: ecandeia@dee.ufcg.edu.br

REVISÃO BIBLIOGRÁFICA

Amostragem Compressiva

A Amostragem Compressiva do inglês *Compressed Sensing*, ou simplesmente CS, é uma técnica de compressão que possibilita a recuperação de um sinal com menos amostras ou medidas que os métodos tradicionais (CANDÈS; WAKIN, 2008; DONOHO, 2006). Este método não se utiliza do conceito apresentado pelo Teorema de Nyquist no qual a taxa de amostragem precisa ser duas vezes maior que a maior frequência do sinal analógico (NYQUIST, 1928). Desta forma, a Amostragem Compressiva se apresenta como um poderoso algoritmo de recuperação de sinais para poucas medições do sinal original que possuem informações que aparentam ser incompletas (CANDÈS; WAKIN, 2008).

Esta técnica é suportada por dois conceitos básicos que tornando-a realizável, os quais são: Esparsidade e Incoerência. Estes são explicados a seguir.

Esparsidade

Um sinal \mathbf{x} com N componentes é esparso no domínio Ψ se sua representação nesse domínio tem no máximo S valores diferentes de zero, ou seja, $\|\mathbf{x}\|_0 \leq S$. A noção de esparsidade é referente a algum domínio de representação ao qual o sinal está (GURJÃO, 2013). Como exemplo um sinal senoidal é não esparso no tempo, porém ao aplicar a transformada de Fourier verifica-se dois valores de frequência os quais o sinal não é zero como visto na Figura 1.

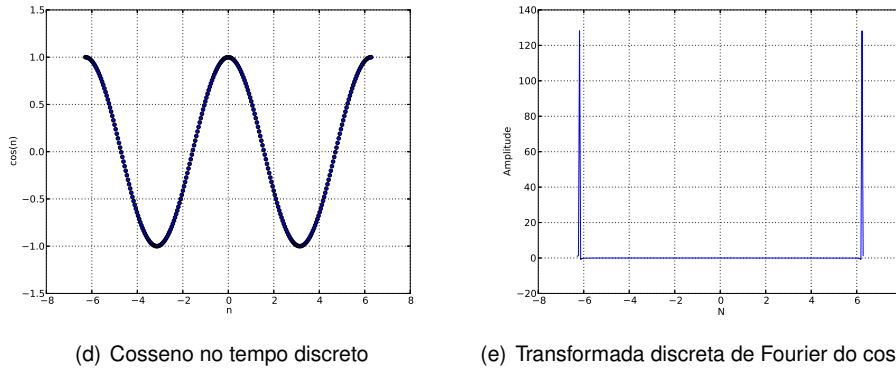


Figura 1: Representação do cosseno no domínio do tempo discreto e da frequência.

A Amostragem Compressiva considera um sinal discreto e esparso representado por um vetor $\mathbf{x} \in \mathbb{R}^N$ e um conjunto de medidas \mathbf{y} dadas por

$$\mathbf{y}_m = \langle \mathbf{x}, \mathbf{a}_m \rangle, \quad m = 1, \dots, M,$$

sendo \mathbf{a}_m os vetores utilizados para as medições. Escrevendo em notação matricial temos $\mathbf{y} = \mathbf{A}\mathbf{x}$.

O conjunto de medidas (projeções) \mathbf{y} dadas por

$$\mathbf{y}_m = \langle \mathbf{x}, \mathbf{a}_m \rangle, \quad m = 1, \dots, M.$$

em notação matricial é dado por $\mathbf{y} = \mathbf{A}\mathbf{x}$, ou seja,

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

Todo vetor \mathbf{x} S -esparso de tamanho N pode ser recuperado a partir de $M \ll N$ medidas tomadas usando uma matriz \mathbf{A} pela solução do problema de otimização convexa

$$(P_1) \min_{\tilde{\mathbf{x}} \in \mathbb{R}^n} \|\tilde{\mathbf{x}}\|_1 \quad \text{sujeito a } \mathbf{A}\tilde{\mathbf{x}} = \mathbf{y}. \quad (1)$$

Amostragem Compressiva mostra que é possível recuperar o sinal \mathbf{x} a partir das suas observações \mathbf{y} . A ideia básica é que sendo o sinal esparso pode-se tomar poucas medições e ainda recuperar o sinal original (GURJÃO, 2013).

Incoerência

Formalmente, dado um par de bases ortonormais, $(\Phi, \mathbf{A}) \in \mathbb{R}^N$, define-se a coerência entre essas bases como

$$\mu(\Phi, \mathbf{A}) = \sqrt{n} \cdot \max_{1 \leq k, j \leq N} |\langle \phi_k, \mathbf{a}_j \rangle|,$$

e mostra-se que $\mu(\Phi, \mathbf{A}) \in [1, \sqrt{N}]$. A Amostragem Compressiva exige que as duas matrizes Φ e \mathbf{A} sejam incoerentes de modo a se obter uma representação esparsa e comprimir um sinal, logo, $\mu(\Phi, \mathbf{A}) = 1$ (CANDÈS; WAKIN, 2008; GURJÃO, 2013).

Algoritmos de Recuperação

Algoritmos de recuperação são implementados de forma a se obter computacionalmente o que a Equação 1 propõe. Mais detalhes sobre estes podem ser encontrados em (TROPP, 2010), todavia neste trabalho explicaremos apenas um algoritmo de perseguição chamado *Orthogonal Matching Pursuit*(OMP).

Orthogonal Matching Pursuit

A definição do algoritmo abaixo é um resumo do que se encontra em (TROPP; GILBERT, 2007) que está descrita em (GURJÃO, 2013).

Partindo de uma sequência de M vetores $\{\mathbf{x}_1, \dots, \mathbf{x}_M\}$ em \mathbb{R}^N , que não dependem de s , são obtidas as medidas

$$\langle s, \mathbf{x}_1 \rangle, \langle s, \mathbf{x}_2 \rangle, \dots, \langle s, \mathbf{x}_M \rangle$$

e a partir dessas medidas deseja-se recuperar o sinal s . Para isso monta-se a matriz Φ cujas linhas são os vetores \mathbf{x}_i , denominada de matriz de medidas com colunas $\varphi_1, \dots, \varphi_N$.

Desse forma, a medida pode ser escrita como

$$\mathbf{v} = \Phi \mathbf{s}$$

e o vetor \mathbf{v} será uma combinação linear das linhas de Φ . Na linguagem de aproximação esparsa \mathbf{v} tem uma representação de m termos sobre o dicionário Φ .

Para recuperar s é necessário determinar que colunas de Φ participam da composição de \mathbf{v} . Para isso seleciona-se colunas de Φ de uma maneira gulosa, ou seja, em cada iteração escolhe-se as colunas de Φ que são mais fortemente correlacionadas com a parte restante de \mathbf{v} , então subtrai-se essa contribuição de \mathbf{v} e repete-se a iteração sobre o resíduo. Mostra-se em m iterações (esparsidade de s) o algoritmo converge para uma boa aproximação.

Dado o vetor esparsa de comprimento N , com somente K posições não zero ($\|\mathbf{x}\|_0 = K$) o número de medidas para recuperar x é $K \log_2(N/K)$.

O algoritmo 1 é uma repetição do apresentado em (TROPP; GILBERT, 2007). As estimativas \hat{s} fornecida por esse algoritmo para o sinal ideal tem os índices não zero listado em Λ_m . Os valores da estimativa \hat{s} são os componentes λ_j de \mathbf{x}_t .

Algorithm 1 Algoritmo OMP apresentado em (TROP; GILBERT, 2007).

[Algoritmo OMP para Recuperação de Sinais] **Entradas:**

- Uma matriz , de medição Φ , $M \times N$
- Um vetor M -dimensional \mathbf{v} de dados
- O nível m de esparsidade do sinal ideal

Saídas:

- Uma estimativa $\hat{\mathbf{s}}$ em R^d do sinal ideal
- Um conjunto Λ_m contendo m elementos de $\{1, \dots, d\}$
- Uma aproximação N dimensional \mathbf{a}_m do vetor \mathbf{v}
- Um resíduo N dimensional $\mathbf{r}_m = \mathbf{v} - \mathbf{a}_m$

Inicialização:

- Faça $\mathbf{r}_0 = \mathbf{v}$, $\Lambda_0 = \emptyset$, e o contador de inicialização $t = 1$

Iteração: Enquanto $t < m$ faça

1. Encontre o índice λ_t que resolve o problema de otimização

$$\lambda_t = \arg \max_{j=1, \dots, d} \langle \mathbf{r}_{t-1}, \varphi_j \rangle$$

2. Amplie o conjunto índice e a matriz com os átomos escolhidos: $\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$ e $\Omega_t = [\Omega_{t-1} \ \varphi_{\lambda_t}]$.
Inicie com Ω_0 como a matriz vazia.
3. Resolva o problema dos mínimos quadrados para obter a nova estimativa do sinal

$$\mathbf{x}_t = \arg \min_{\mathbf{x}} \|\mathbf{v} - \Phi_t \mathbf{x}\|_2$$

4. Calcule a nova aproximação dos dados e o novo resíduo

$$\mathbf{a}_t = \Phi_t \mathbf{x}_t \text{ e } \mathbf{r}_t = \mathbf{v} - \mathbf{a}_t$$

5. Incremente t .
-

METODOLOGIA

Arquitetura do problema em FPGA

Com intuito de implementar um algoritmo de reconstrução em *hardware*, foi proposto uma arquitetura para o OMP a qual já foi realizada em (RABAH et al., 2014; STANISLAUS; MOHSENIN, 2014; SEPTIMUS, 2010). Na Figura 2 encontra-se ilustrada em diagrama de blocos proposto para este trabalho.

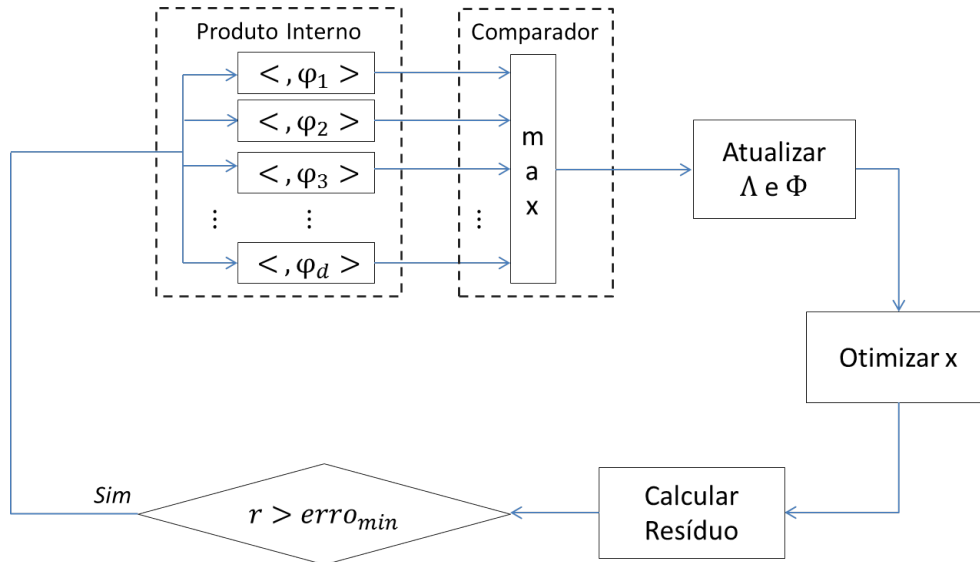


Figura 2: Arquitetura simplificada do *Orthogonal Matching Pursuit* para Hardware

A Figura 2 é apenas uma simplificação em blocos do algoritmo descrito na seção anterior. Porém, faz-se necessário destacar que o bloco de Otimização de x envolve encontrar a matriz pseudo-inversa e em hardware para se obter uma solução realizável os métodos QR e decomposição de Cholesky são utilizados, como apresentados em (RABAH et al., 2014; STANISLAUS; MOHSENIN, 2014; SEPTIMUS, 2010).

De forma inicial, optou-se por projetar o bloco de produto interno, esta operação representada em diagrama de blocos na Figura 3 e, utilizando-se de multiplicadores (representados em "X") e os somadores (representados em Σ).

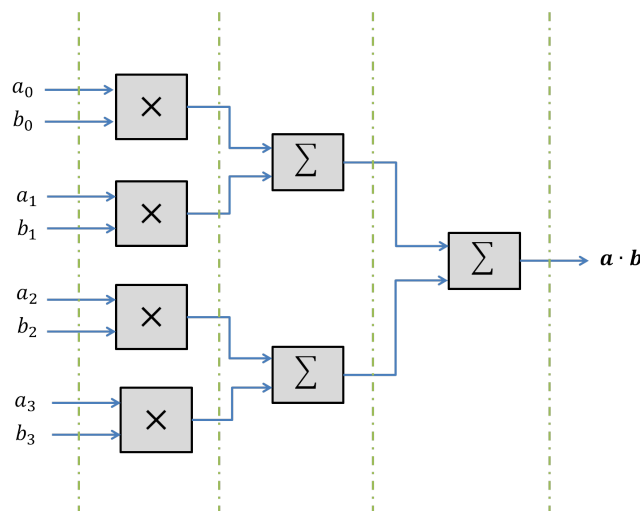


Figura 3: Arquitetura para o cálculo de produto interno para um vetor de 4 elementos

Arquitetura do produto interno no ambiente Quartus II

A implementação do produto interno foi feita no Quartus II 9.0 Web Edition com o uso da ferramenta da *MegaWizard Plug-In Manager*, considerando vetores de 4 elementos, cada elemento com representação em ponto flutuante e precisão simples de 32 bits. A Figura 4 ilustra a representação da caixa preta do produto interno.

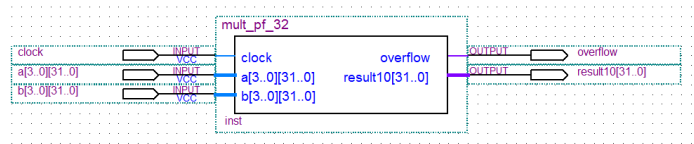


Figura 4: Bloco do produto interno no Quartus II

Utilizando-se do *MegaWizard Plug-In Manager* projetou-se os blocos multiplicador e somador utilizados no projeto os quais são ilustrados nas figuras 5(a) e 5(b), os quais o número de pulsos de *clock* para realização de cada operação são 5 para o multiplicador e 7 para o somador, ou seja, uma operação seguida dos dois resultaria em 12 pulsos de *clock*. Assim como, o projeto utilizando-se dos blocos construídos é apresentado na Figura 6 e o número de pulsos de *clock* para realização de toda operação é 19.

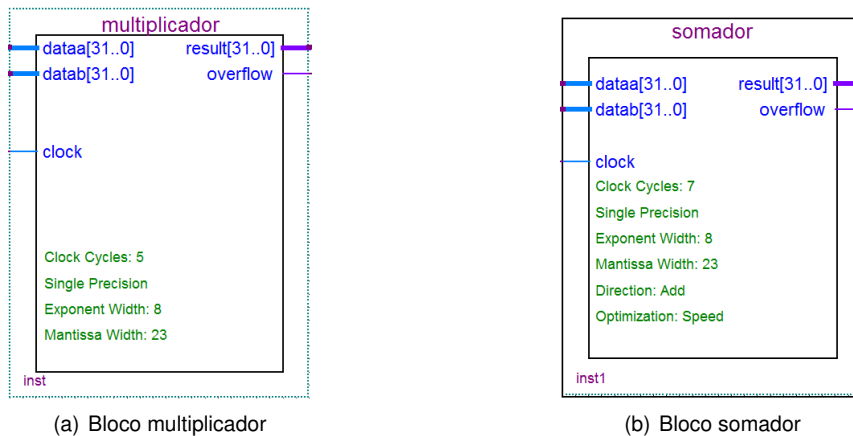


Figura 5: Representação do blocos somador e multiplicador.

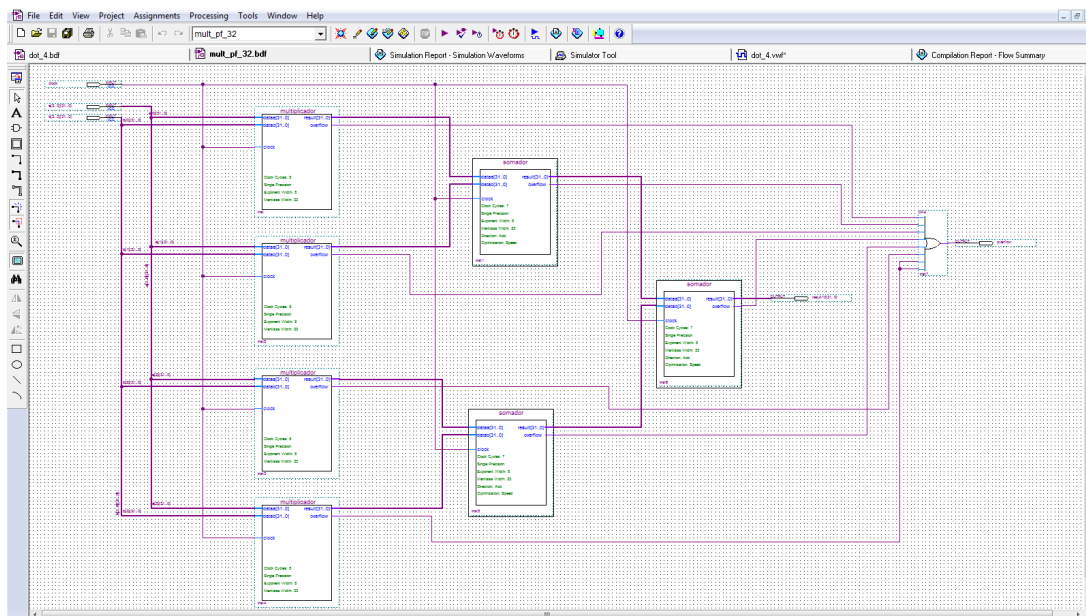


Figura 6: Projeto no Quartus II para o cálculo de produto interno em ponto flutuante para um vetor de 4 elementos

RESULTADOS E DISCUSSÃO

Os elementos dos vetores estão em representação de ponto flutuante que pelo padrão IEEE implica em 1 bit de sinal, seguido por 8 bits de expoente e 23 de mantissa. Este fato precisa ser levado em conta para validação do sistema projetado, para isso foi utilizado da ferramenta em <http://www.h-schmidt.net/FloatConverter/IEEE754.html> para tal conversão.

Os resultados da simulação são encontrados na Figura 7, os dados utilizados para esta encontram-se na Tabela 1. Foi verificado um erro pequeno causado pela imprecisão na representação binária, todavia para valores com maior precisão binária o resultado foi validado sem erros.

a	1.5000	10.0750	250.8000	-7.4597
b	342.42	-0.11111111	5.1E-4	8.7878
a · b				-2.9158936

Tabela 1: Valores utilizados para simulação do produto interno

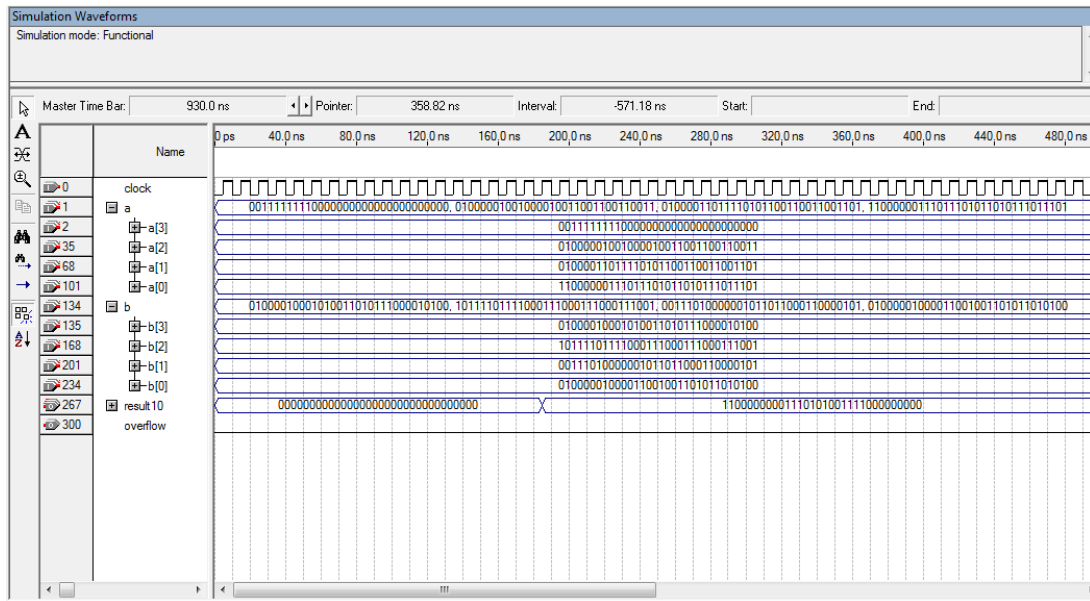


Figura 7: Simulação no Quartus II do cálculo de produto interno em ponto flutuante para um vetor de 4 elementos

CONCLUSÕES

Neste trabalho, foi realizado o estudo teórico sobre amostragem compressiva e uma arquitetura simples foi proposta para a implementação em FPGA do algoritmo OMP. De forma inicial, o bloco da operação de produto interno em ponto flutuante foi implementado e validado.

Algumas dificuldades durante o projeto podem ser citadas, como por exemplo o fato de que o Quartus II Web Edition não permite simulações para todas as famílias de FPGAs, logo, a complexidade dos projetos foi reduzida. Devido a esta limitação, o vetor utilizado para simulação teve o número de elementos reduzidos. Outra dificuldade relaciona-se com tempo hábil destinado ao projeto que por se tratar de um PIBITI teve uma ênfase maior no aprendizado e utilização das ferramentas que nos resultados obtidos.

Como trabalhos futuros, pretende-se criar os outros blocos da arquitetura e utilizar a licença do Quartus que garante maior complexidade no projeto. Como também, implementar o algoritmo em um FPGA e obter resultados experimentais.

AGRADECIMENTOS

Os autores agradecem a Universidade Federal de Campina Grande (UFCG) e ao CNPq por financiar este trabalho de PIBITI/CNPq-UFCG.

Referências

CANDÈS, E. J.; WAKIN, M. B. An introduction to compressive sampling. **IEEE SIGNAL PROCESSING MAGAZINE**, v. 25, p. 21–30, March 2008.

DONOHO, D. L. Compressed sensing. **IEEE TRANSACTIONS ON INFORMATION THEORY**, v. 52, p. 1289–1306, April 2006.

GURJÃO, E. C. Telecomunicações: Teoria, avanços e aplicações. In: _____. [S.l.]: SBrT, 2013. cap. Introdução a Amostragem Compressiva, p. 1–26.

NYQUIST, H. Certain topics in telegraph transmission theory. **AIEE**, v. 47, p. 617–644, April 1928.

RABAH, H. et al. Fpga implementation of orthogonal matching pursuit for compressive sensing reconstruction. **IEEE TRANSACTION ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS**, PP, p. 1, October 2014.

SEPTIMUS, A. High performance compressive sensing reconstruction hardware with qrd process. **PROCEEDINGS OF 2010 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS)**, p. 3316–3319, May 2010.

STANISLAUS, J.; MOHSENIN, T. High performance compressive sensing reconstruction hardware with qrd process. **PROCEEDINGS OF 2012 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS)**, p. 29–32, May 2014.

TROPP, J. A. Computational methods for sparse solution of linear inverse problems. **PROCEEDINGS OF THE IEEE**, v. 98, p. 948–958, May 2010.

TROPP, J. A.; GILBERT, A. C. Signal recovering from random measurements via orthogonal matching pursuit. **IEEE TRANSACTIONS ON INFORMATION THEORY**, v. 53, n. 12, p. 4655–4666, December 2007.