

Arquiteturas FPGA para Cômputo da Matriz de Covariância Amostral em Sensoriamento Espectral

Elivander Judas Tadeu Pereira, Dayan Adionel Guimarães

Resumo—Várias estatísticas de teste para sensoriamento espectral em redes de rádio cognitivo fazem uso da matriz de covariância amostral do sinal recebido, como é o caso do detector baseado no índice de Pietra-Ricci (*Pietra-Ricci index detector*, PRIDE). Este artigo compara duas arquiteturas de hardware para cômputo de tal matriz em arranjos de portas programáveis em campo (*field programmable gate array*, FPGA). O algoritmo do computador digital de rotação de coordenadas (*coordinate rotation digital computer*, CORDIC) com poucas iterações surge como alternativa à arquitetura convencional para a multiplicação de amostras complexas, prometendo trazer grande economia de hardware na implementação do PRIDE.

Palavras-Chave—Sensoriamento espectral, aritmética de ponto fixo, FPGA, ASIC, CORDIC.

Abstract—Many of the test statistics for spectrum sensing in cognitive radio networks make use of the received signal sample covariance matrix (SCM), which is the case of the Pietra-Ricci index detector (PRIDE). This article compares two hardware architectures for computing the SCM in field programmable gate array (FPGA). The coordinate rotation digital computer (CORDIC) algorithm with few iterations arises as an alternative to the conventional architecture for the multiplication of complex samples, promising to bring significant hardware savings to the implementation of the PRIDE.

Keywords—Spectrum sensing, fixed-point arithmetic, FPGA, ASIC, CORDIC

I. INTRODUÇÃO

A escassez de espectro em determinadas bandas de radio-frequência (*radio frequency*, RF) tem motivado pesquisas para melhorar a eficiência da utilização espectral com base no acesso dinâmico ao espectro. Nesse cenário um usuário não licenciado, chamado de usuário secundário (*secondary user*, SU), faz uso do espectro de um usuário licenciado, chamado de usuário primário (*primary user*, PU), sem causar interferências em suas transmissões. Técnicas baseadas no conceito de rádio cognitivo (*cognitive radio*, CR) [1], como o sensoriamento espectral vem sendo estudadas para tentar solucionar este problema.

O desempenho requerido e a complexidade das técnicas de sensoriamento podem ser vistos como um grande empecilho para a sua adoção prática. Atualmente, a cooperação entre vários terminais secundários, o emprego de estatísticas de teste cegas (que não dependem de conhecimento prévio de

parâmetros do ruído ou do sinal primário) [2], [3] e o uso de chips mais capazes e reconfiguráveis, destacando-se os arranjos de portas programáveis em campo (*field programmable gate array*, FPGA), têm se tornado viabilizadores para estudos de implementação em hardware de técnicas de sensoriamento espectral [4], [5].

Várias estatísticas de teste cegas dependem do cálculo da matriz de covariância amostral (*sample covariance matrix*, SCM) do sinal [3]. Esse tipo de cálculo, quando executado em equipamentos que fazem o processamento de forma sequencial, como uma unidade de processamento central (*central processing unit*, CPU), pode causar grande latência no sensoriamento espectral. O processamento em paralelo, viabilizado por circuitos integrados de aplicação específica (*application specific integrated circuit*, ASIC) ou por FPGAs, permite execução rápida dessa tarefa, minimizando a latência.

Este artigo compara duas arquiteturas de hardware para cômputo da matriz de covariância amostral em FPGA. O computador digital de rotação de coordenadas (*coordinate rotation digital computer*, CORDIC) surge como alternativa à arquitetura convencional para a multiplicação de amostras complexas, prometendo trazer economia de hardware na implementação de estatísticas de teste que fazem uso da SCM, como é o caso do detector baseado no índice de Pietra-Ricci (*Pietra-Ricci index detector*, PRIDE) [6].

O restante do artigo está assim estruturado: a Seção II apresenta o modelo de sensoriamento espectral, enquanto a Seção III descreve o formato da aritmética de ponto fixo adotada. A Seção IV traz as arquiteturas de hardware propostas e os resultados comparativos são dados na Seção V. Por fim, as conclusões são apresentadas na Seção VI.

II. MODELO DE SENSORIAMENTO

A. Modelo matricial do sinal

No modelo de sensoriamento cooperativo centralizado com fusão de dados, cada um dos m terminais secundários em cooperação coleta N amostras do sinal emitido por p transmissores primários em cada intervalo de sensoriamento, digitaliza tais amostras e as encaminha para o centro de fusão (*fusion center*, FC) através de um canal de controle. No FC, tais amostras serão processadas para que se forme a estatística de teste que permitirá a decisão acerca do estado de ocupação da banda sensorizada.

O FC é responsável por organizar os sinais recebidos em uma matriz $\mathbf{Y} \in \mathbb{C}^{m \times N}$ dada por

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{V}, \quad (1)$$

Elivander J. T. Pereira, Dayan A. Guimarães, Instituto Nacional de Telecomunicações (Inatel), Santa Rita do Sapucaí - MG. E-mail: elivander@mtel.inatel.br, dayan@inatel.br. Este trabalho foi financiado em parte pela RNP, com recursos do MCTIC, processo N° 01245.010604/2020-14, sob o projeto Brasil 6G do Centro de Referência em Radiocomunicações (CRR) do Instituto Nacional de Telecomunicações – Inatel, e em parte pelo CNPq e pela FAPEMIG.

na qual cada linha representa as amostras colhidas por um SU, como ilustrado na Fig. 1, considerando apenas 1 transmissor primário por simplicidade. A matriz $\mathbf{H} \in \mathbb{C}^{m \times p}$ representa os ganhos complexos do canal entre os transmissores primários e cada SU. A matriz $\mathbf{X} \in \mathbb{C}^{p \times N}$ contém as amostras dos sinais primários e a matriz $\mathbf{V} \in \mathbb{C}^{m \times N}$ contém as amostras do ruído aditivo Gaussiano branco (*additive white Gaussian noise*, AWGN) que contamina cada SU, com média zero e variância σ_v^2 . Sob a hipóteses \mathcal{H}_1 e \mathcal{H}_0 que denotam a presença e a ausência dos sinais primários na banda sensoriada, tem-se $\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{V}$ e $\mathbf{Y} = \mathbf{V}$, respectivamente.

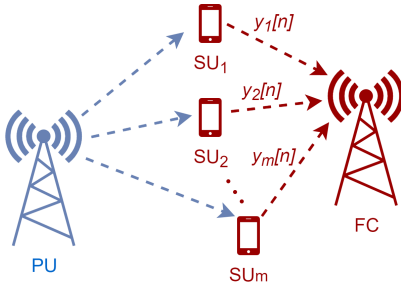


Fig. 1. Sensoriamento cooperativo centralizado com fusão de dados.

Algumas técnicas de sensoriamento espectral cooperativo dependem do cálculo da SCM de ordem $m \times m$, dada por

$$\mathbf{R} = \frac{1}{N} \mathbf{Y} \mathbf{Y}^\dagger, \quad (2)$$

em que \dagger representa o operador Hermitiano (complexo-conjugado transposto). Uma vez que a SCM é computada no FC, forma-se a partir dela uma estatística de teste para decidir acerca do estado de ocupação do espectro. Neste trabalho, o foco não será a análise de desempenho de uma estatística de teste em específico, mas sim em arquiteturas para cômputo da SCM. No entanto, a estatística de teste do detector PRIDE será utilizada para comparações de desempenho das arquiteturas avaliadas, através de simulações computacionais. A estatística de teste do PRIDE é dada por

$$T_{\text{PRIDE}} = \frac{\sum_{i=1}^{m^2} |r_i|}{\sum_{i=1}^{m^2} |r_i - \bar{r}|}, \quad (3)$$

em que $\bar{r} = \frac{1}{m^2} \sum_{i=1}^{m^2} r_i$, é o valor médio dos elementos do vetor \mathbf{r} formado pelo empilhamento das colunas de \mathbf{R} , e r_{ij} é o elemento na i -ésima linha e j -ésima coluna de \mathbf{R} .

B. Modelo orientado à implementação

O modelo de sinal (1) implicitamente considera que as amostras enviadas ao FC não sofrem nenhum tipo de processamento adicional e no FC estarão disponíveis tal como foram colhidas nos SUs. No entanto, há que se considerar o efeito do processamento do sinal quando leva-se em conta a implementação em um dispositivo receptor real. Em [7], um modelo orientado à implementação em hardware é apresentado. Nele, o processo de sensoriamento realizado pelos SUs da rede secundária é modelado por um receptor de conversão direta, ilustrado na Fig. 2.

O sinal recebido é inicialmente amplificado por um amplificador de baixa figura de ruído (*low noise amplifier*, LNA) e

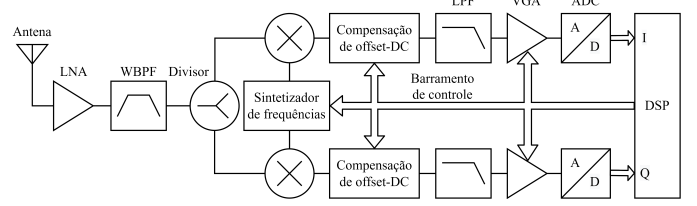


Fig. 2. Arquitetura de receptor de conversão direta (ou zero-IF).

filtrado por um filtro passa-faixas de banda larga (*wide band pass filter*, WBPf). O sinal resultante é dividido e rebatido para banda base, processo que insere um nível de tensão contínua residual (*DC-offset*) ao sinal, fazendo-se necessário um circuito para compensação desse nível. Após compensação do *DC-offset*, o sinal em cada ramo passa por um filtro passa-baixas (*low pass filter*, LPF) e por um amplificador de ganho variável (*variable gain amplifier*, VGA) que visa manter o sinal dentro da faixa dinâmica do conversor analógico-digital (*analog to digital converter*, ADC). Uma vez que os sinais em fase e em quadratura foram digitalizados, eles são processados por um processador digital de sinais (*digital signal processor*, DSP), que irá realizar o branqueamento (*whitening*) do sinal e preparar as amostras para transmissão para o FC.

Neste trabalho, o modelo de sinal leva em conta o modelo de receptor orientado à implementação de [7], com as amostras sendo digitalizadas nos SUs e enviadas para o FC utilizando um número N_b de bits. O impacto do número de bits utilizados por amostra no canal de controle também é avaliado no trabalho citado, demonstrando ser possível a utilização de $N_b = 3$ bits por amostra (3 bits para a parte real e 3 bits para a imaginária) sem degradar o desempenho.

III. PROCESSAMENTO DE AMOSTRAS EM PONTO FIXO

O FC, ao receber as amostras de 3 bits sem sinal, quantizadas e digitalizadas nos SUs, deve adotar uma nova representação binária para processá-las. Neste trabalho, o processamento da SCM foi implementado em um FPGA e, por isso, adotou-se a aritmética de ponto fixo com sinal através de palavras binárias em complemento de dois (*two's complement*). No FPGA, as amostras de sensoriamento recebidas são tratadas como dados com sinal (*signed*), o qual é definido pela biblioteca *ieee.numeric_std* juntamente com as funções necessárias para as operações aritméticas básicas (soma, subtração, multiplicação e divisão) com esse tipo de dado.

O número de bits utilizado na entrada do sistema para representar em ponto fixo as partes real e imaginária é N_p . Esse número e o seu formato (relação entre o comprimento da palavra e a parte fracionária em ponto fixo), impactam diretamente o desempenho, a complexidade e o consumo de recursos da arquitetura. A largura de bits de saída ao longo de cada operação pode ser alterada, mas há dependência entre o desempenho e os valores utilizados na entrada.

Na notação de ponto fixo $[W, F]$, tem-se os comprimentos da palavra binária (*wordlength*) e da parte fracionária, respectivamente. O bit mais significativo (*most significant bit*, MSB) é dedicado ao sinal (0 para números positivos e 1 para números negativos). Dessa forma, $F \leq W - 1$ e o número de bits dedicados à parte inteira será $W - F - 1$, podendo ser zero.

Neste trabalho avaliou-se o número N_p mínimo que poderia ser utilizado sem ser perceptível a degradação do desempenho do sistema, bem como a relação entre o comprimento total da palavra e da parte fracionária. Dessa análise foi definido que as amostras que chegam com $N_b = 3$ bits devem ser processadas com $N_p = 6$ bits no formato de ponto fixo com sinal [6, 5]. No formato [6, 5] é possível representar amostras com passo mínimo de 0,03125 volt e que variam entre -1 e $+0,96875$ volt. Resultados e análises de desempenho para essa representação são apresentados e discutidos na Seção V.

IV. ARQUITETURAS PARA CÔMPUTO DA SCM

Nesta seção apresentam-se duas arquiteturas para cômputo da SCM. Ambas foram dimensionadas para $m = 4$ rádios cognitivos. Na primeira arquitetura (denominada convencional), 4 multiplicadores reais são construídos utilizando-se dos recursos existentes no FPGA, enquanto a segunda arquitetura (CORDIC) baseia-se em apenas 1 multiplicador real e uma estrutura composta apenas de somadores, registradores ou tabelas de consulta (*look up tables*, LUTs) e deslocadores. Neste trabalho, o CORDIC é uma implementação dos autores sem uso de propriedade intelectual do fabricante do FPGA.

A. Arquitetura convencional

Para calcular os valores da SCM é necessário utilizar unidades multiplicador-acumulador (*multiply-accumulate*, MAC). A arquitetura do MAC é composta por um multiplicador complexo, configurado de tal forma a realizar o produto entre a n -ésima amostra do i -ésimo SU, $y_i[n]$, e a n -ésima amostra conjugada do j -ésimo SU, $y_j^*[n]$. As saídas do multiplicador são conectadas a acumuladores, que irão executar a soma entre o valor calculado e o valor armazenado em um registrador e armazenar o novo valor. Os dois acumuladores alimentam as parcelas real e imaginária da saída do MAC que é r'_{ij} . Note que esse ainda não é o valor final de um elemento da SCM definida em (2), pois a saída do MAC não possui os valores normalizados pelo número de amostras N . Sendo assim, os valores da SCM são $r_{ij} = r'_{ij}/N$.

A Fig. 3 ilustra a arquitetura convencional com os parâmetros utilizados neste trabalho. As amostras recebidas dos SUs no processo de sensoriamento, após serem convertidas para ponto fixo com 6 bits, [6, 5], alimentam as entradas do MAC. Ao entrar no MAC, as amostras são redimensionadas para 7 bits, [7, 5], com a adição de um bit de sinal estendido. Quatro multiplicadores reais realizam os produtos das parcelas gerando sinais intermediários com 14 bits, [14, 10], que em seguida são somados para se obter as partes real e imaginária da multiplicação. Considerando a escala numérica utilizada, não é necessário o uso de 4 bits para a parte inteira desses valores. Sendo assim, após a soma há um redimensionamento para 12 bits (um bit para parte inteira e outro para sinal), [12, 10], antes de alimentar os acumuladores. Nos acumuladores, as amostras são somadas e armazenadas em um registrador com 18 bits, [18, 10]. A representação de saída do MAC proposto tem passo mínimo de 2^{-10} e escala entre -128 e $+127,999$. Na figura em questão, os valores de largura de bits em vermelho indicam as operações de redimensionamento.

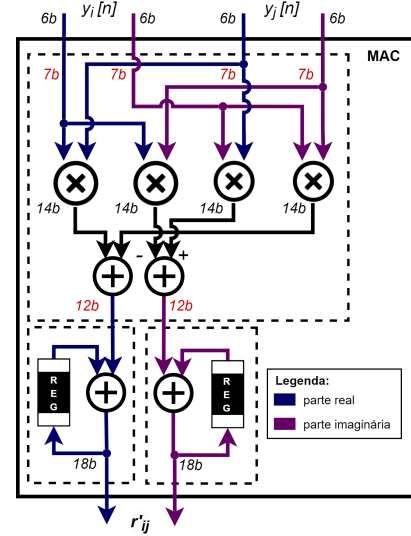


Fig. 3. Arquitetura do MAC convencional.

Como a SCM tem características Hermitianas, o seu cômputo pode ser reduzido para apenas uma região fora da diagonal principal (acima ou abaixo), mais a própria diagonal principal, fazendo uso de $m(m+1)/2$ MACs. Uma vez computados os r'_{ij} , todos os sinais são multiplexados e passam por um divisor por N , obtendo-se finalmente r_{ij} . A Fig. 4 apresenta a arquitetura completa de cômputo dos elementos da SCM, de forma convencional.

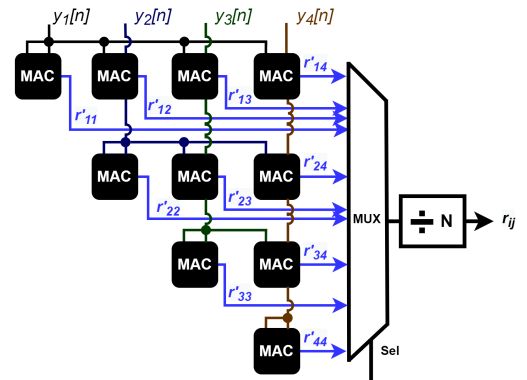


Fig. 4. Arquitetura do módulo de cálculo da SCM convencional.

B. Arquitetura CORDIC

Alternativamente à arquitetura anteriormente descrita, que representa uma abordagem convencional do ponto de vista de implementação, outra arquitetura é proposta utilizando-se do algoritmo CORDIC. Trata-se de um algoritmo que opera rotações iterativas em vetores que vão da origem dos eixos até as coordenadas fornecidas em sua entrada [8].

O CORDIC possui dois modos de operação. No primeiro, o funcionamento objetiva minimizar a componente residual no eixo vertical (a parcela imaginária, nesse caso), chamado de vetorização ou translação, servindo então como um módulo para cálculo de valor absoluto (módulo) ou para conversão de coordenadas Cartesianas para polares. No outro modo de operação, chamado de rotação, o vetor é rotacionado de

acordo com o ângulo de entrada (valor que se deseja eliminar a componente residual, nesse caso). O modo de rotação é útil para cálculo de funções trigonométricas (seno e cosseno por exemplo) e para conversão de coordenadas polares para Cartesianas.

O CORDIC é construído utilizando-se apenas somadores, deslocadores, registradores e LUTs, recursos existentes nos módulos lógicos adaptativos (*adaptive logic module, ALM*) do FPGA.

Na arquitetura proposta, ilustrada na Fig. 5, o módulo de cálculo da SCM insere, antes das operações nos MACs, 4 unidades CORDIC em modo de translação, que convertem as amostras de 6 bits no sistema Cartesiano para coordenadas polares com 8 bits, [8, 5] (na Fig. 6, $d_i[n] = 1,642|y_i[n]|$ é a magnitude vezes o ganho do CORDIC e $\theta_i[n]$ o ângulo).

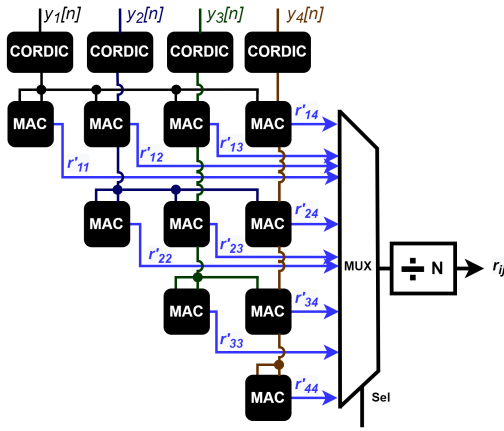


Fig. 5. Arquitetura do módulo de cálculo da SCM com o CORDIC.

A multiplicação de números complexos em coordenadas polares simplifica a arquitetura. A magnitude resultante é igual ao produto das magnitudes e, nesse caso, tem saída redimensionada para o formato com sinal de 12 bits. A fase resultante é igual a soma dos ângulos de entrada. Como o bloco deve calcular o produto entre conjugados complexos, a entrada j correspondente ao segundo SU tem o sinal do seu ângulo invertido. Uma vez que o produto complexo é calculado em coordenadas polares, um CORDIC em modo de rotação é utilizado para retornar ao sistemas de coordenadas Cartesianas antes da soma acumulativa. As componentes reais e imaginárias na saída desse CORDIC possuem 12 bits no formato [12, 8]. Esse formato é capaz de representar a saída do MAC com passo mínimo de 2^{-8} e escala entre -512 e $+511,996$. Essa arquitetura de MAC é ilustrada na Fig. 6.

Nota-se que a representação de saída nessa arquitetura reduz a largura de bits para a parte fracionária, enquanto aumenta a largura para a parte inteira do número em ponto fixo. Isso se deve ao fato de que algoritmo CORDIC produz em sua saída um ganho cujo valor exato é dado em função do número de iterações. Esse ganho é aplicado durante a conversão para coordenadas polares e no retorno às coordenadas Cartesianas, resultando em um ganho ao cubo (para 4 iterações, o ganho combinado é de 4,431) multiplicando todos valores da SCM. Como essa operação apenas altera linearmente a escala da SCM, isso não impacta em diferença de desempenho. Logo, não se faz necessária a compensação desse valor.

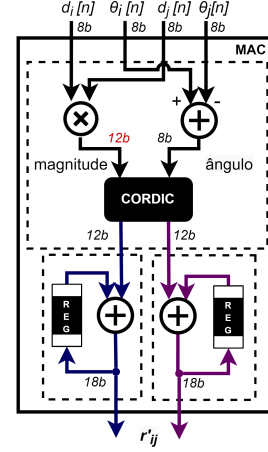


Fig. 6. Arquitetura do MAC utilizando o CORDIC.

V. RESULTADOS

Para avaliar o desempenho em relação à largura de bits na representação em ponto fixo, algumas simulações foram feitas em MATLAB. Partindo do código de simulação apresentado para o modelo orientado a implementação de [7], buscou-se encontrar o menor valor de bits que não causasse degradação perceptível no valor referência, calculado pelo MATLAB com valores do tipo *double* em ponto flutuante (64 bits de precisão). Os resultados apresentados na Fig. 7 foram calculados utilizando dos seguintes parâmetros: $p = 1$ PU, $m = 4$ SUs, $N = 100$ amostras por SU, $SNR = -8$ dB, $N_b = 3$ bits (números de bits de quantização utilizados para as transmissões dos SUs para o FC), $N_p = 6$ bits (número de bits para representação aritmética e processamento no FC), 4 iterações para as funções CORDIC e 10000 execuções de Monte Carlo. A estatística de teste empregada foi o PRIDE.

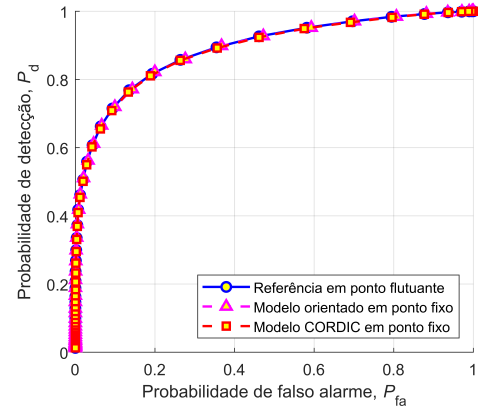


Fig. 7. Análise de ponto fixo com 6 bits para os modelos orientado a implementação [7] e CORDIC (com 4 iterações).

Por meio da curva característica de operação do receptor (*receiver operating characteristic, ROC*) é possível notar que o uso de uma representação aritmética com 6 bits, no formato [6, 5], não causa degradação perceptível ao desempenho do detector PRIDE. Além disso, como o algoritmo CORDIC é iterativo, avaliou-se o número mínimo de iterações necessárias para que a sua precisão não afetasse o desempenho. As simulações no MATLAB demonstraram que o uso de apenas 4 iterações é suficiente para que o desempenho não seja

degradado de forma perceptível.

Os resultados de simulação apresentados na Fig. 7 foram utilizados como parâmetros iniciais para a construção das duas arquiteturas anteriormente propostas em *very high speed integrated circuit (VHSIC) hardware description language (VHDL)*, no programa Quartus Prime 20.1 (ALTERA/Intel). A síntese do código foi feita tendo como alvo a placa de desenvolvimento ALTERA Cyclone® V SoC.

A Tabela I resume a utilização de recursos no FPGA para as duas arquiteturas. A arquitetura convencional mapeia cada multiplicador real para algoritmos de multiplicação de ponto fixo em complemento de dois, por exemplo, baseados no multiplicador de Booth [9]–[11]. No caso da arquitetura CORDIC, todos os blocos são construídos fazendo uso de somadores e registradores do FPGA e apenas um único multiplicador real.

Os dados de construção das duas arquiteturas se revelaram muito próximos entre si, com a arquitetura convencional demonstrando consumo aproximadamente 10% maior de recursos de ALM. Dos recursos utilizados, a arquitetura CORDIC mostra um número 25% a menos de *look-up tables* adaptativas (ALUT) na sua composição, mas um número muito maior de registradores comparado à arquitetura convencional. A Fig. 8 destaca a área ocupada pela construção das duas arquiteturas no FPGA.

TABELA I
UTILIZAÇÃO DE RECURSOS NO FPGA CYCLONE® V SOC
(5CSXFC6D6F31C6).

	Convencional	CORDIC
ALMs	1639	1495
ALUT combinacional	3180	2406
Registradores	360	820

Apesar das utilizações de recursos pelas duas arquiteturas estarem próximas uma da outra, vale ressaltar que o cálculo das operações de módulo da estatística de teste, segundo (3), é muito mais simples e eficiente de ser executado através do CORDIC no modo de vetorização (translação). O cálculo de valor absoluto convencional depende de elevar as partes reais e imaginárias ao quadrado, fazendo-se necessários dois multiplicadores reais e um cálculo de raiz quadrada, executado, por exemplo, por meio de um algoritmo como aquele proposto em [12].

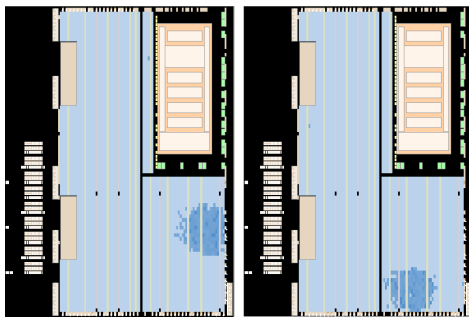


Fig. 8. Utilização e alocação de recursos lógicos no chip FPGA pela arquitetura convencional (esquerda) e CORDIC (direita).

VI. CONCLUSÕES

Neste trabalho foi estudado inicialmente o impacto da aritmética de ponto fixo com baixo número de bits para processamento de amostras no sensoriamento espectral. Simulações demonstraram que é possível assegurar o desempenho, mesmo nessas condições. Um algoritmo alternativo, o CORDIC, foi introduzido para arquiteturas com essa finalidade. Por ser um processo iterativo, o estudo também analisou o impacto do número de iterações utilizado e verificou-se por meio de simulações que quatro iterações asseguram desempenho sem degradação perceptível.

Do ponto de vista de implementação, foram apresentadas duas arquiteturas para cálculo da matriz de covariância amostral no FPGA. O CORDIC é uma opção pouco explorada para tais arquiteturas. Seu desempenho em consumo de recursos lógicos para a arquitetura proposta foi ligeiramente superior, mas a implementação completa de uma estatística de teste, como o PRIDe [6], pode resultar em significativa redução na utilização de recursos para o cálculo das operações de módulo, pois o modo de vetorização do CORDIC calcula diretamente o módulo do vetor de entrada.

Trabalhos futuros serão conduzidos para integrar as arquiteturas projetadas aos módulos de cálculo da estatística de teste para estender a análise aqui reportada.

REFERÊNCIAS

- [1] J. Mitola III and G. Q. Maguire Jr., “Cognitive radio: making software radios more personal,” *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, Aug 1999, doi: 10.1109/98.788210.
- [2] A. Kortun, T. Ratnarajah, M. Sellathurai, C. Zhong, and C. B. Papadias, “On the performance of eigenvalue-based cooperative spectrum sensing for cognitive radio,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 49–55, 2011.
- [3] D. A. Guimarães, “Robust test statistic for cooperative spectrum sensing based on the gerschgorin circle theorem,” *IEEE Access*, vol. 6, pp. 2445–2456, 2018.
- [4] M. Safavi and M. Safavi, “A real-time, low-power implementation for high-resolution eigenvalue-based spectrum sensing,” *Analog Integrated Circuits and Signal Processing/Springer*, 12 2013.
- [5] R. B. Chaurasiya and R. Shrestha, “Hardware-efficient vlsi architecture and asic implementation of grcr-based cooperative spectrum sensor for cognitive-radio network,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 2, pp. 166–176, 2022.
- [6] D. A. Guimarães, “Pietra-Ricci index detector for centralized data fusion cooperative spectrum sensing,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12 354–12 358, 2020, doi: 10.1109/TVT.2020.3009440.
- [7] D. Guimarães and E. Pereira, “Influence of a direct-conversion receiver model on the performance of detectors for spectrum sensing,” *Journal of Communication and Information Systems*, vol. 36, no. 1, p. 173–183, Nov. 2021. [Online]. Available: <https://jcis.sbrt.org.br/jcis/article/view/789>
- [8] J. E. Volder, “The cordic trigonometric computing technique,” *IRE Transactions on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, 1959.
- [9] A. M. H. Al-Bermani and R. K. Mohamed, “Design and implementation of high speed complex multiplier using FPGA,” *Al-Nahrain Journal for Engineering Sciences*, vol. 11, no. 1, pp. 91–97, 2008.
- [10] N. Sharma and R. Sindal, “Modified booth multiplier using wallace structure and efficient carry select adder,” *International Journal of Computer Applications*, vol. 68, no. 13, 2013.
- [11] R. Nikhil, G. Veerendra, J. Harsha, and D. Prabhakar, “Implementation of time efficient hybrid multiplier for FFT computation,” *International Journal of Engineering & Technology*, vol. 7, p. 409, 03 2018.
- [12] Y. Li and W. Chu, “A new non-restoring square root algorithm and its vlsi implementations,” in *Proceedings International Conference on Computer Design. VLSI in Computers and Processors*, 1996, pp. 538–544.