A Low Complexity Cooperative Spectrum Sensor for Cognitive –Radio Network Based on Approximate Computing

1st Aiswarya P B
M.Tech Student
Nehru College of Engineering and Research Centre
Pampady,Thrissur,Kerala,India

2nd Anooja B
Electronics and Communication Engineering
Nehru College of Engineering and Research Centre
Pampady,Thrissur,Kerala,India

***Abstract*— With the rapid growth of new wireless technologies, the issue of spectrum scarcity is becoming more prominent due to the limited availability of radio spectrum. Unutilized spectrum bands designated for primary users are known as spectrum holes or white spaces. Assigning these white spaces to secondary users can enhance spectrum utilization. Dynamic spectrum access is crucial for sharing licensed spectrum with secondary users, as opposed to static allocation policies which impede effective spectrum utilization. Spectrum sensing aims to detect licensed users in the spectrum to prevent interference from secondary users. Approximate computing has emerged as a strategy to enhance efficiency by leveraging the fault resilience of various applications. It introduces trade-offs in design optimizations, allowing for significant reductions in VLSI circuit area and energy consumption. A cooperative spectrum-sensing (CSS) algorithm based on approximate computing for cooperative cognitive-radio networks is proposed to reduce computational complexity for efficient hardware design while maintaining detection performance. The hardware-efficient architecture of the cooperative spectrum sensor (CSR) based on the CSS algorithm, combined with resource-sharing architectural optimization, outperforms existing implementations by occupying less area, achieving shorter sensing times, and displaying superior hardware efficiency.**

Keywords—Approximate Computing, Cooperative Spectrum Sensor(CSR), very-large scale integration(VLSI)

I. INTRODUCTION

Advances in technology have created a surge in the demand for wireless communication, leading to a scarcity of available spectrum resources. This challenge has prompted the need for dynamic spectrum access techniques to replace traditional static spectrum allocation policies. Cognitive Radio (CR) networks have emerged as a solution to this issue by allowing secondary users to access unused spectrum bands without disrupting primary users. Spectrum sensing, a crucial process that monitors the radio spectrum for available opportunities, is central to the operation of CR networks. Efficient spectrum sensing is vital for maximizing network throughput while minimizing interference. However, conventional techniques face challenges like high computational complexity, limiting their practicality in hardware-limited CR devices. To overcome this hurdle, there is a growing interest in developing hardware-efficient spectrum sensing algorithms using approximate computing. This project aims to create a low-complexity Cooperative Spectrum Sensor (CSS) based on approximate computing principles for cognitive radio networks. The CSS algorithm is designed to reduce computational complexity while maintaining high detection performance, enabling efficient hardware design and implementation. The project also includes a hardware-efficient architecture for the CSS, optimizing resource sharing to enhance hardware efficiency. By advancing spectrum sensing techniques in CR networks, this project aims to improve the utilization of limited radio spectrum resources.

II. SYSTEM ARCHITECTURE

The diagram below illustrates the top-level design of the cooperative spectrum sensor using the N-PRIDe algorithm. It is designed to support four secondary users with four digital input signals, each represented by y1[n], y2[n], y3[n], and y4[n] for all n={0, 1, 2, ..., N-1} where N=32 samples. The received digital signal samples from the four secondary users are initially processed by the Sample Covariance Matrix formulation module (SFM) to generate diagonal (r1 - r4) and upper off-diagonal (UOD) elements (u1 - u6). These outputs are then used in the real diagonal-element summation module (RDSM) to calculate the rd or rdb value. The upper off-diagonal elements and rd value are further utilized in the numerator computation module (NCM) to determine the Numerator (Tη) value, while also being used in the Denominator Computation Module (DCM) to calculate the denominator (Tδ) value. The Tη and Tδ values are then used in a divider to compute the test statistics TN-PRIDe. The N-PRIDe algorithm offers good detection performance with lower computational complexity than traditional methods. Finally, the TN-PRIDe value is compared against a predetermined threshold γ-value to produce the overall decision output PUdet.

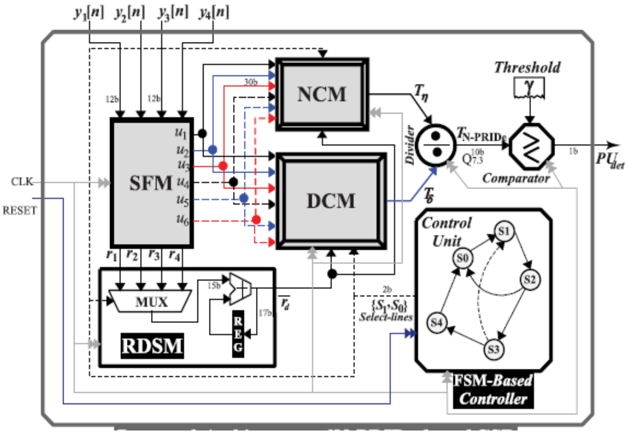


Figure 1 Propsed CSR Architecture based on N-PRIDe

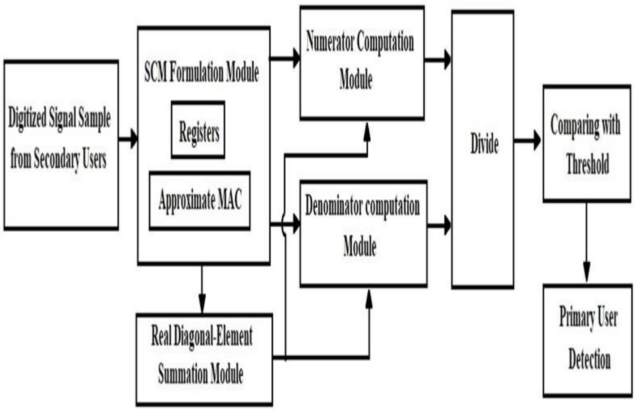

Digitized Signal Sample from Secondary Users
SCM Formulation Module
Registers
Approximate MAC
Numerator Computation Module
Denominator computation Module
Divide
Comparing with Threshold
Primary User Detection
Real Diagonal-Element Summation Module

Figure 2 Block Diagram of Proposed method

III. PROPOSED WORK

A. Proposed Approximate 4-2 Compressor

The Cooperative Spectrum Sensor (CSR) proposal involves the use of Sample Covariance Matrix Formulation Modulation (SFM) to calculate both upper off-diagonal (UOD) and diagonal elements of the Sample Covariance Matrix (SCM). A hardware architecture for SFM is detailed, featuring 10 multiply-and-accumulate (MAC) units and four input registers (REGs), supporting m=4 secondary users (SU) and N=32 signal samples. Approximate computing is seen as an effective method to reduce energy consumption in signal processing tasks with error-tolerant characteristics such as multimedia algorithms involving matrix multiplications. To minimize energy usage while allowing for computational errors, a cost-effective approximate multiply-accumulate (MAC) operator is needed. In this study, cost-effective approximate 4-2 compressors are developed by focusing on error directions rather than error amounts, creating two types: positive compressors (PC) and negative compressors (NC) that produce inaccuracies in opposite directions. By ensuring a value of 1 at $\tilde{y}_i$ for all input scenarios, hardware complexity and power consumption decrease significantly, particularly with zero-valued inputs leading to large relative errors. In order to ensure the precision of zero-valued inputs without incurring high hardware costs and minimizing performance loss in practical applications, the suggested approximate compressors will consistently output a value of 1 for $\tilde{y}_i$ except when all inputs are zero as follows :

$$\tilde{Y}_i^P = \tilde{Y}_i^N = a_i + b_i + c_i + d_i \qquad (1)$$

where $\tilde{Y}_i^P$ and $\tilde{Y}_i^N$ represent the approximate values of y_i from the proposed PC and NC models. Compared to the original equation, the proposed approximation of $\tilde{y}_i$ produces positive errors because we only change the value 0 to 1, regardless of the type of compressor. On the contrary, the approximation of $\tilde{y}_{i+1}$ is realized with a negative modification, changing the value of $\tilde{y}_{i+1}$ to 0. Specifically, in the proposed NC architecture, we must invert a sufficient number of values to change the direction of the compressor error, which is initially on the positive side, aggressively approximating the lower position of the bit $\tilde{Y}_i^N$. The hardware complexity of implementing an approximate compressor must be taken into account, which gives the following approximate equation:

$$\tilde{Y}_{i+1}^N = a_i b_i + c_i d_i \qquad (2)$$

On the other hand, the estimation of the upper-bit position for the positive compressor is created by as it were considering the equipment costs. Thus, we have more opportunity for changing the values of the first condition where the surmised condition can be defined as:

$$\tilde{Y}_{i+1}^P = a_i b_i + b_i c_i + b_i d_i + c_i d_i \qquad (3)$$

B. Proposed Approximate Multiplier

Using the proposed approximation compressors, we also develop two types of approximation coefficients depending on the direction of the error, called positive multiplier (PM) and negative multiplier (NM). A proposed positive multiplier (PM) for approximately 8 × 8 multiplication was implemented by replacing all the exact compressors in approximate ranges with designed PC in all compressor stages. On the other hand, for the architecture proposed 16 × 16 positive multiplier (PM), we mix a positive compressor (PC) and a negative compressor (NC) to reduce the peak errors, resulting in an additional biased error distribution in the positive direction.We use the proposed PCs in stages 1 and 3 of the compressor, while the proposed NCs are only used to estimate the 2nd stage to compensate for the internal error. We carefully refine each compressor input to minimize the amount of error by considering input patterns that produce inaccurate results for a given compressor architecture. The simplified multiplication process then generates the error value EMUL = AM(A, X)−A × X, where AM(A, X) is the approximate product of the product multiplier A and multiplier X. Based on the expected error for each approximate compressor, the EMUL probability can also be accurately analyzed.Specifically, we first determine the expected column-level error of the multiplier by accumulating all the expected errors of the approximation compressors in each bit position.

The approximation multiplier is then assumed to generate errors by considering the weight of each bit position. For a given approximate interval w, we first observe E(EPMUL) and E(ENMUL) to determine the mixing ratio of the two approximate multipliers are denoted by $\rho = E(ENMUL)/E(EPMUL)$. During MAC operations, after one PM-based multiplier, we perform an intermediate ρ-multiplier with NM architecture. A balanced error distribution can be made by adjusting the number of PM and NM modules according to the mixing ratio ρ. The proposed interleaving method allows for more approximations without degrading the algorithm level performance, resulting in energy-efficient MAC processing. The proposed work transforms the existing exact coefficients into only approximations with opposite error directions, so that the application of the proposed interleaving concept does not attract additional overhead for the last parallel MAC operators that use multiple coefficients. . If we aim at a resource-limited computing platform containing only one multiplier hardware according to the proposed interleaving concept, it is possible to use positive and negative compressors inside the multiplier to balance the error distributions as much as possible.

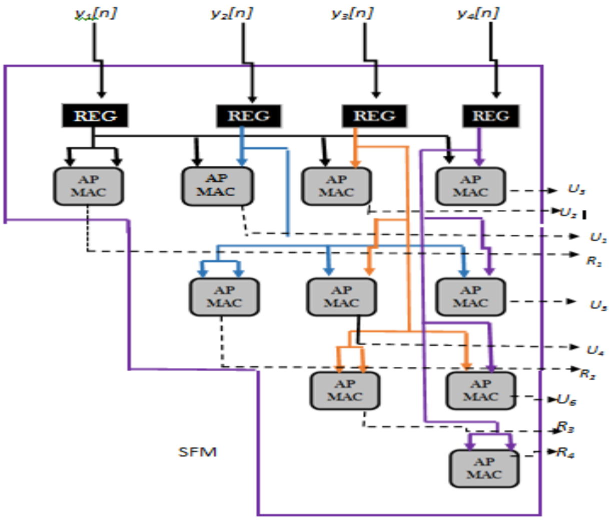


Figure 3 Proposed VLSI Architecture of SFM

C. Restoring Square Root Algorithm

The restoring square root algorithm operation is similar to that of division operation with changing divisor. In the Denominator Computation Module (DCM) and the Numerator Computation Module (NCM) ,the bit trimming output will be the input for the square root . Let us consider the bit trimming output as $x_{i.}$ The primary step is the multiplication of x_i by 2 where x_i is the partial remainder of the previous calculation. That is , multiplying x_i by 2 means shifting to the left . After that Q_i is multiplied by 2 and 2Qi is added with 2^{i+1}using adder where the adder is designed using the verilog code.The next step is calculated by subtracting 2Qi + 2^{i+1} from $2x_{i,}$where the subtractor is designed using the verilog code. If the sum $2Q_I + 2^{i+1}$ is greater than $2x_i$, the next calculation is considered as negative and the sign of the remainder will be checked by the controller. If the remainders sign is positive,then x_i+1 will be the remainder provided by the multiplier or else if the sign is negative $2x_i$ will be provided for the next iteration. So the subtract block has two outputs. For the first output The new remainder x_{i+1} will be used and for the second output the restore remainder $2x_i$ will be used. In the final step the final remainder is determined only if the the square root of the radicand has been calculated by the restoring algorithm.

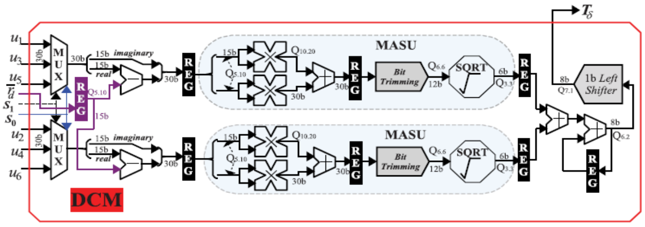


Figure 4 Proposed VLSI Architecture of NCM

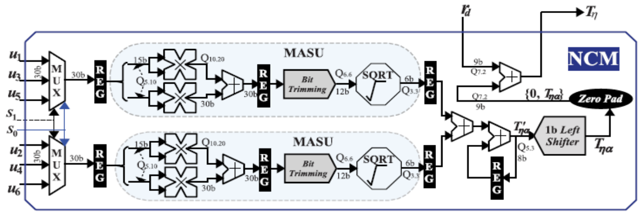


Figure 5 Proposed VLSI architecture of DCM

D. Restoring Division Algorithm

The restoring division algorithm is the method used to perform binary division which is used in digital arithmetics. It is an iterative process that resembles the long division method used in decimal arithmetic but operates on binary numbers .It operates by repeatedly subtracting the divisor from the dividend that is the method involves a sequence of shifts and subtractions to calculate the quotient and remainder . It restores the partial remainder if the result is negative. The algorithm uses a sign bit to track the sign of the partial remainder and adjusts accordingly. The process continues until the division is comlete and the quotient and remainder are obtained.

IV. SIMULATION RESULTS

The evaluation of the proposed hardware-efficient RDSM, DCM, and NCM architectures was conducted using ModelSim for simulation and Xilinx for synthesis. The key metrics compared include the total number of 4-input Look-Up Tables (LUTs), the number of bounded Input/Output Blocks (IOBs), total equivalent gate counts, power consumption, and delay.

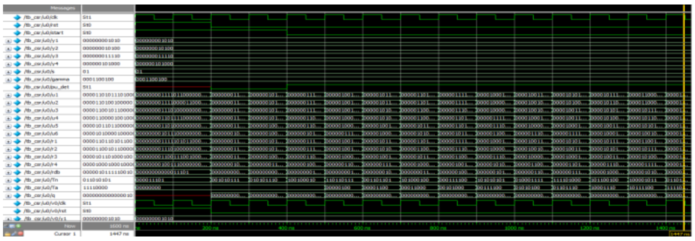


Figure 6 Simulation result of propsed CSR

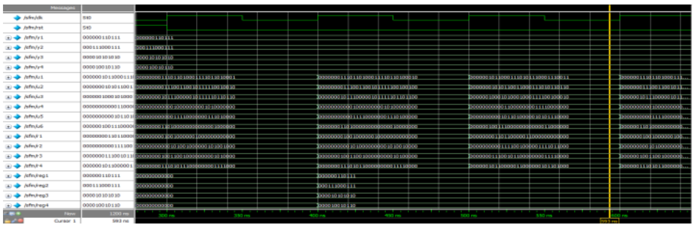


Figure 7 Simulation result of the proposed SFM

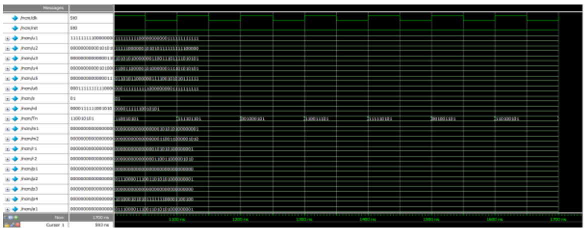


Figure 8 Simulation result of the proposed NCM

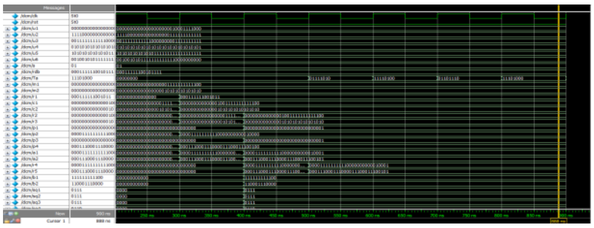
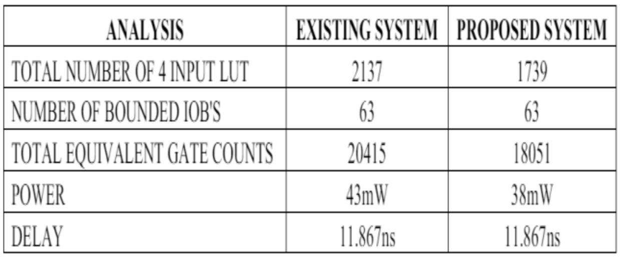


Figure 9 Simulation result of the proposed DCM



ANALYSIS	EXISTING SYSTEM	PROPOSED SYSTEM
TOTAL NUMBER OF 4 INPUT LUT	2137	1739
NUMBER OF BOUNDED IOB'S	63	63
TOTAL EQUIVALENT GATE COUNTS	20415	18051
POWER	43mW	38mW
DELAY	11.867ns	11.867ns

Table 1 Result Analysis

The proposed system uses 1739 LUTs compared to 2137 in the existing system, indicating a reduction of approximately 19%. This reduction highlights the efficiency of the proposed design in terms of logic resource utilization. Both the existing and proposed systems utilize 63 bounded IOBs. This indicates that the I/O requirements remain unchanged while achieving improvements in other areas. The proposed system has a total equivalent gate count of 18051, which is 12% less than the 20415 gate counts in the existing system. This reduction reflects the optimization in the proposed architecture, leading to lower hardware complexity.

CONCLUSION

The proposed hardware-efficient architectures for the Real Diagonal Summation Module (RDSM), Denominator Computation Module (DCM), and Numerator Computation Module (NCM) significantly enhance the performance and efficiency of cooperative spectrum sensing in cognitive radio networks. By optimizing the computational complexity and reducing the hardware resources required, our design demonstrates a substantial decrease in the total number of 4-input LUTs, equivalent gate counts, and power consumption compared to existing systems. These optimizations are achieved without compromising the delay, which remains consistent with the existing system, thus ensuring high-speed performance.

Overall, the proposed system showcases a balanced approach to improving hardware efficiency and energy consumption. The reductions in resource usage and power make it a viable solution for applications where power efficiency and hardware simplicity are critical. The consistent performance in terms of delay further underscores the practicality of our design in real-time applications. These advancements position our design as a highly effective and sustainable choice for future cognitive radio networks, offering a robust foundation for further innovations in spectrum sensing technologies.

REFERENCES

[1]. R. B. Chaurasiya and R. Shrestha, “A new hardware-efficient spectrum sensor VLSI architecture for data-fusion based cooperative cognitive radio network,” IEEE Trans. Very Large Scale Integr. (VLSI),2021, vol. 29, no. 4, pp. 760–773 .

[2]. R. B. Chaurasiya and R. Shrestha, “Area-efficient and scalable data-fusion based cooperative spectrum sensor for cognitive radio,” IEEE Trans. Circuits Syst. II, Exp. Briefs, Apr. 2021. vol. 68, no. 4, pp. 1198–1202.

[3]. R. B. Chaurasiya and R. Shrestha, “Hardware-efficient and fast sensing-time maximum-minimum-eigenvalue-based spectrum sensor for cognitive radio network,” IEEE Trans. Circuits Syst. I, Reg. Papers, Nov. 2019, vol. 66, no. 11, pp. 4448–4461.

[4]. Z. Chen, D. Guo, and J. Zhang, “Deep learning for cooperative spectrum sensing in cognitive radio,” in Proc. IEEE 20th Int. Conf. Commun. Technol. (ICCT), Oct. 2020, pp. 741–745.

[5]. K. Cichon, A. Kliks, and H. Bogucka, “Energy-efficient cooperative spectrum sensing: A survey,” IEEE Commun. Surveys Tuts., 2016,vol. 18, no. 3, pp. 1861–1886, 3rd Quart..

[6]. D. A. Guimaraes, “Pietra–Ricci index detector for centralized data fusion cooperative spectrum sensing,” IEEE Trans. Veh. Technol. , Oct. 2020, vol. 69, no. 10, pp. 12354–12358.

[7]. X. Hong, J. Wang, C.-X. Wang, and J. Shi, “Cognitive radio in 5G: A perspective on energy-spectral efficiency trade-off,” IEEE Commun. Mag., Jul. 2014, vol. 52, no. 7, pp. 46–53.

[8]. I. Qiqieh, R. Shafik, G. Tarawneh, D. Sokolov, and A. Yakovlev, “Energy-efficient approximate multiplier design using bit significancedriven logic compression,” in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2017, pp. 7–12.

[9]. C. Stevenson, G. Chouinard, Z. Lei, W. Hu, S. Shellhammer, and W. Caldwell, “IEEE 802.22: The first cognitive radio wireless regional area network standard,” IEEE Commun. Mag., Jan. 2009.vol. 47, no. 1, pp. 130–138.

[10]. M. Wang *et al.*, “An optimized compression strategy for compressorbased approximate multiplier,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.