Mixed-Mode Signal Processing for Implementing MCMC MIMO Detector

Amin Aghighi, Behrouz Farhang-Boroujeny, and Armin Tajalli(✉)

Electrical and Computer Engineering Department, University of Utah, Salt Lake City, USA
armin.tajalli@utah.edu
https://lcas.ece.utah.edu/

Abstract. A hybrid analog/digital signal processor has been proposed to implement energy-efficient multi-input-multi-output (MIMO) detectors. A sub-optimum MIMO detector based on Markov Chain Monte Carlo (MCMC) algorithm for a 4 × 4 MIMO system is presented. A careful partitioning between analog and digital domains has been made to reduce system power consumption. The outputs of the proposed analog signal processing unit are being converted to digital using a low-resolution analog-to-digital converter (ADC), to deliver the signals to the digital portion of the detector system. The proposed 4 × 4 MCMC MIMO detector is designed in a standard 45 nm CMOS technology, that consumes 29.3 mW from 1.0 V supply. A throughput of 235.3 Mbps is achieved, while operating at 1.0 GHz clock frequency. The design occupies a 0.11 mm^2 silicon area.

Keywords: Optimal detectors · sub-optimal detectors · Markov Chain Monte Carlo (MCMC) · VLSI MIMO · Mixed-mode MIMO · Mixed-mode circuits

1 Introduction

Modern wireless communications use the Multi-Input Multi-Output (MIMO) approach to improve data throughput at a lower cost. Moreover, the ever-growing number of users makes MIMO systems even more desirable [1]. Since MIMO systems use the same frequency band for transmitting parallel data streams, data transfer bandwidth improves with the number of transmit antennas [2]. Therefore, receiver-joint detection is crucial for exploiting the full capacity of the system. Although the optimum detectors can harness the full channel capacity, their complexity increases exponentially with the number of transmit antennas [3]. As a result, improving the performance of sub-optimum detectors is a demanding research topic [4–7]. A few implementations for sub-optimum detectors operating based on the Markov-Chain Monte Carlo (MCMC) algorithm are reported

Published by Springer Nature Switzerland AG 2021
A. Calimera et al. (Eds.): VLSI-SoC 2020, IFIP AICT 621, pp. 21–37, 2021.
https://doi.org/10.1007/978-3-030-81641-4_2

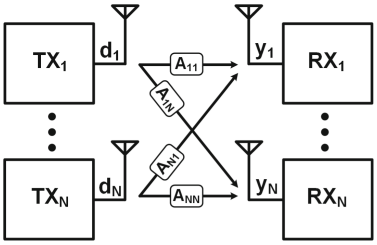

Fig. 1. MIMO system input-output description.

in [2,8–10]. Although this detector can achieve full channel capacity, the existing implementations, which are mainly based on Digital Signal Processors (DSP), result in fairly complex and power-hungry circuits.

This paper targets lowering power consumption and increasing throughput of the MCMC detectors by moving high-speed and energy-hungry operations from DSP to analog/mixed-mode domain. A set of system-level simulations are carried out to show the performance of the proposed analog/mixed-mode approach. Several analog building blocks are proposed to implement target signal processing schemes in a more energy-efficient way. The power and area cost of these blocks are calculated through simulations to have a good cost estimation of the proposed detector.

The rest of this paper is organized as follows: Sect. 2 provides a brief overview on MCMC detectors, and describes the high level implementation of the proposed MCMC detector. System level simulation results are presented in Sect. 3. Section 4 demonstrates circuit-level implementation, and Sect. 5 provides comparison between the proposed MCMC detector and the state-of-the-art.

2 System Level MCMC Detector

2.1 Conventional MCMC Detectors

Considering a flat fading channel, the input-output for each channel that is shown in Fig. 1, can be described by:

$$y = A \cdot d + n \tag{1}$$

where, $\mathbf{y}$ is the received data vector, $\mathbf{d}$ is the transmitted data bits, $\mathbf{n}$ is the channel additive noise vector, and $\mathbf{A}$ is the channel gain matrix. The MCMC detector is based on an iterative approach. While the detection algorithm is thoroughly described in [2], a brief concept review is presented here. The MCMC detector takes a random set of initial bits, $\mathbf{b_0}$, by a Gibbs sampler [11], and calculates the error function based on:

$$e_0 = y - A \cdot b_0 \tag{2}$$

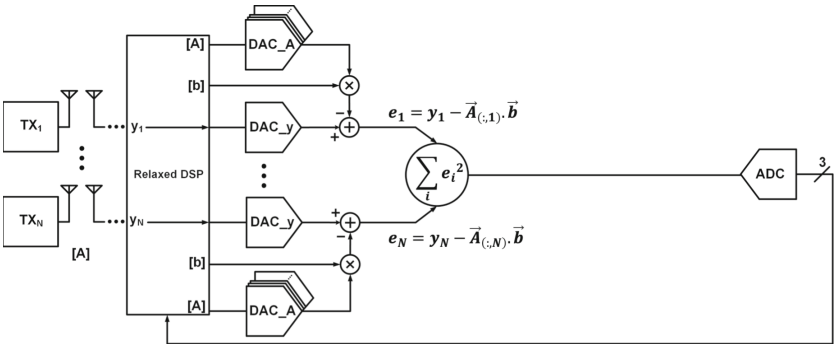

Fig. 2. The proposed MCMC detector block diagram.

where, each element of $\mathbf{e_0}$ represents the initial error associated with each channel.

In other words:

$$e_{0,i} = y_{0,i} - (A_{1,i} \cdot b_{0,1} + A_{2,i} \cdot b_{0,2} + \cdots + A_{N,i} \cdot b_{0,N}) \tag{3}$$

where, $\mathbf{i}$ is the desired channel index and $\mathbf{N}$ is equal to the number of transmit antennas. When all of the $\mathbf{e_0}$ elements are found, one of the bits in $\mathbf{b_0}$ will be flipped in each iteration, and the error vector, $\mathbf{e}$, will be recalculated. The new error vector in each iteration can be readjusted from the previous value:

$$e_k = e_{k-1} + A_{:,m} \cdot (2b_{(k-1),m}) \tag{4}$$

where, $\mathbf{k}$ is the index for the iteration steps, $\mathbf{m}$ is the index for the bit which is flipped, and $\mathbf{A}_{:,\mathbf{m}}$ represents the m-th column of $\mathbf{A}$. In order to compare the error functions between every two consecutive iterations, and decide about the m-th bit, the summation of squared components of $\mathbf{e}$ is calculated and defined as:

$$E = e_1^2 + e_2^2 + \cdots + e_N^2 \tag{5}$$

Based on the $\mathbf{E}$ value at each iteration step, it will be decided to keep either "+1" or "−1" for the m-th bit. When the decision has been made for all the bits in $\mathbf{b_0}$, the MCMC detector repeats this operation for $\mathbf{N_{gs}}$ times for different initial conditions. The performance (accuracy) of the MCMC detector improves when $\mathbf{T}$ parallel Gibbs samplers run with different initial random set of bits [2]. As including randomness to the decision process reduces the stalling problem, the MCMC detector introduced in [2] utilizes a random variable, $\mathbf{v}$, and considers both $\mathbf{E}$ and $\mathbf{v}$ for making decisions.

Usually, MCMC detectors employ Forward Error Correction (FEC), in order to reduce the Bit Error Rate (BER). In this work, we will analyze only the raw BER, before applying FEC.

2.2 Proposed MCMC Detector Circuit

Figure 2 shows the proposed MCMC detector, in which most of the speed-limited operations are moved into the analog domain. In order to calculate each com-

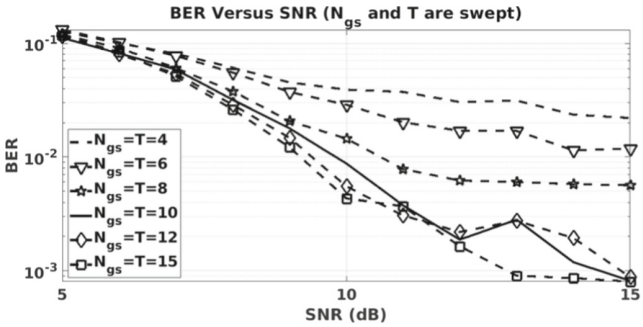

Fig. 3. BER versus SNR while sweeping $\mathbf{N_{gs}}$ and $\mathbf{T}$.

ponent of the error vector, $\mathbf{e}$, a set of digital-to-analog converters (DACs) are employed to convert the digitized received data, $\mathbf{y}$, and the channel gain matrix, $\mathbf{A}$, to the analog domain. The DACs are implemented based on very low-power and low complexity circuits. Bit-wise multiplication is implemented in the analog domain to multiply $\mathrm{DAC_A}$ by $\mathbf{b_0}$, and then simply subtract the result from $\mathrm{DAC_y}$ to produce the error vector, $\mathbf{e}$. As will be shown in Sect. 4, the summation occurs in the current domain and no extra hardware is required. There are N parallel operators in total needed to produce all the components of the $\mathbf{e}$. A square generator produces the $\mathbf{e_i^2}$ for each channel separately and adds them together to estimate the updated value for $\mathbf{E}$. Finally, a low-resolution analog-to-digital converter (ADC) closes the loop by digitizing the $\mathbf{E}$ value, leaving the final step of decision, i.e., determining $\mathbf{b_0}$ to digital circuits. As will be shown later, the entire operation explained above is implemented using simple analog circuits that occupy a small area and need only one clock cycle to finish each step.

The proposed system needs one clock cycle to generate each component of $\mathbf{e}$. Therefore, considering the arrays of $\mathrm{DAC_A}$ and $\mathrm{DAC_y}$ are working in parallel, producing $\mathbf{N}$ components of $\mathbf{e}_0$ takes one clock cycle based on (2). Since all of the building blocks in Fig. 2 are synchronized and working at the same frequency, producing and digitizing $\mathbf{e_0}$ also occurs at the same clock rate. Hence, it takes one clock cycle to fulfill all the operations shown in Fig. 2. Based on (3), another clock cycle is required to calculate $\mathbf{e}_k$, and decide about k-th bit of $\mathrm{k} > 1$. Therefore, while making decision for the first bit takes two clock cycles, the next bits require only one clock cycle to be determined. Hence, it takes $(\mathrm{N}+1)$ clock cycles in order to determine the polarity of all the bits in $\mathbf{b}$.

3 System Level Simulation Results

High-level simulations have been carried out to determine the performance of the circuit shown in Fig. 2, and determine the required specifications for the key building blocks in this architecture. Some design parameters, such as the number

of parallel Gibbs samplers **T**, the number of iterations for each Gibbs sampler $\mathbf{N_{gs}}$, and the required resolution for each data converter, will be determined based on this study. For simplicity, it is assumed that each antenna transmits bits over only one carrier, i.e., there is no sub-carrier. The number of transmit antennas is set to **N**. Also, in order to have enough number of bits for calculating BER, the whole MCMC detector iterates for $N_{MC} = 500$ times, which results in $N_{MC} \times T \times N_{gs} \times N$ total number of bits. Here, **T** and $\mathbf{N_{gs}}$ are the most critical design parameters that determine the cost and the performance of the detector. This work uses the same **T** and $\mathbf{N_{gs}}$ values that have been employed in [2].

Figure 3 shows Bit-Error-Rate (BER) versus signal-to-noise ratio (SNR) for the received signal, while **T** and $\mathbf{N_{gs}}$ have been swept. As it is expected, BER improves by $\mathbf{N_{gs}}$ and **T**. However, improvement for $N_{gs} = T > 10$ is marginal. It is reported in [2] that $N_{gs} = T = 6$ to 8 is a good compromise. In order to determine the right resolution for each of the data converters (DAC_A, DAC_y and ADC), a two-step verification has been implemented. Initially, the resolution of each data converter gradually reduces, while all the other data converters are considered to be ideal. This approach prevents the other data converters to contribute to the total quantization noise and affect the system performance. In the second step, the entire system is simulated while all the data converters have been employed with their limited resolution. Figure 4(a), (b) and (c) represent the MCMC performance, while the resolution for DAC_A, DAC_y and ADC have been swept, respectively. As it is shown in Fig. 4(a), increasing DAC_A resolution higher than 6 bits, does not improve the MCMC performance. Therefore, the resolution of DAC_A needs to be 6b or more, in order not to lose much performance. When the resolution of DAC_A is more than 6b, the system needs to be simulated for much longer in order to produce a precise BER estimation. As can be seen in Fig. 4(a), the results for resolutions higher than 6b are not very accurate. Based on the results shown in Fig. 4(b) and (c), one can conclude that the minimum resolution for DAC_y and the ADC are 6b and 3b, respectively.

Figure 5 represents the second step of our analysis, in which the resolution of all the data converters have been limited to the values discussed above. Based on these results, the performance of the proposed MCMC detector, which is utilizing a realistic model for the data converters, is very close to those reported in Figs. 4(a), (b) and (c).

4 Circuit Level Simulations

This Section demonstrates circuit-level implementations and simulation results for the proposed detector shown in Fig. 2.

Figure 6 shows the merged DAC_A, DAC_y, and the multiplier for producing the error vector, **e**. Current mode logic (CML) based circuits have been used to simplify the design, and also make it possible to linearly operate at very high frequencies with a low level of consumption and complexity [12–16]. While DAC_A and multiplier are shown in the right-hand side of the schematic (Fig. 6), the programmable differential pair at the left-hand side represents DAC_y. Current-mode

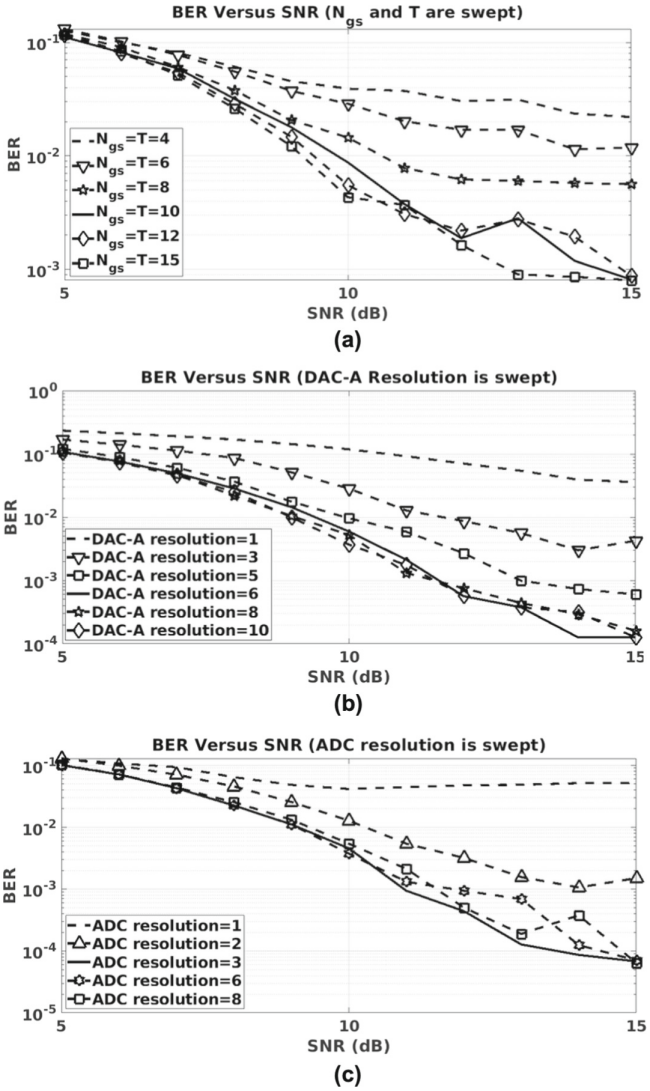

Fig. 4. BER versus SNR while sweeping (a) DAC_A resolution, (b) DAC_y resolution, and (c) ADC resolution.

DAC architecture has been used to simplify the multiplication and summation operations. Depending on sign of **b** values, the output of DAC_A is multiplied by "+1" or "−1". The outputs of the **N** parallel DACs are shorted properly to implement a summer, as required in (3).

The consumption of the DACs depends on the resolution as well as their speed of operation. In order to make sure that the DACs can operate properly at the desired clock frequency, the time constant at the output node should be chosen carefully:

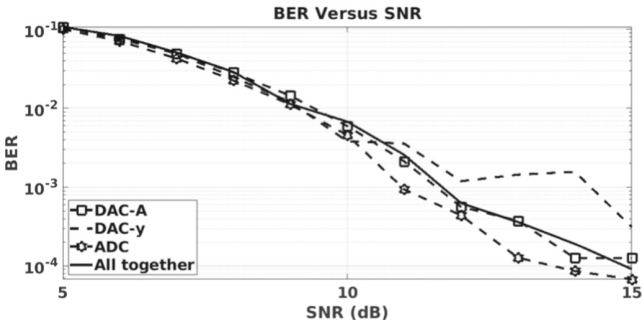

Fig. 5. BER versus SNR when including all the data converters.

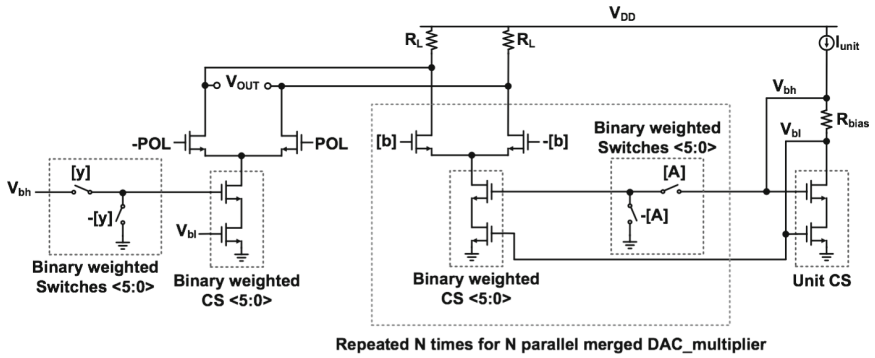

Fig. 6. DAC_A merged with multiplier and DAC_y for producing the e function for each channel.

$$R_L < 1/(2\pi C_{L,DAC} \times 5 f_{bit}) \tag{6}$$

where $C_{L,DAC}$ is the DAC load capacitance, and f_{bit} is the input bit frequency. Here, a factor of five is considered to assure settling with an error of less than 1%. The maximum DAC output swing will be achieved when all of the binary-weighted current sources are turned on in all of the $N + 1$ parallel DACs on each channel. Since there are a total of $2^6 - 1$ copies of the unit current source for each DAC, the minimum required unit current, I_{unit}, is equal to:

$$I_{unit} = \frac{V_{swing,sq}}{(2^6 - 1) \times (N + 1) \times R_{L,DAC}} \tag{7}$$

where, $V_{Swing,sq}$ is the squarer required input swing, and R_L is the load resistance determined by (6). Figure 7 represents the circuit level implementation of the squarer block. Given that all of the devices work in the saturation region, and assuming ideal long channel device characteristics, it can be shown that [17]:

$$I_{SQ} = I_{D1} + I_{D2} = kV_{id}^2 + 2I_B \tag{8}$$

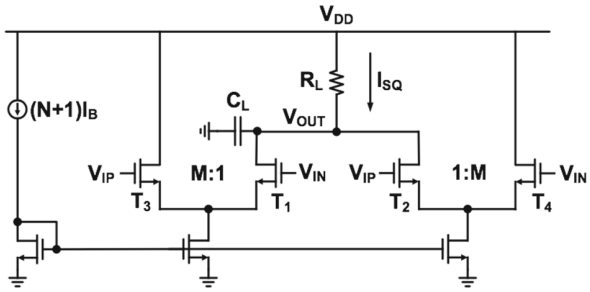

Fig. 7. Squarer circuit schematic.

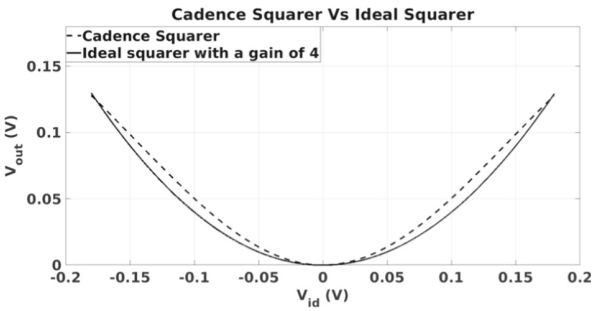

Fig. 8. Input-output transfer characteristics of the simulated squarer circuit.

where, V_{id} is the input differential voltage, $k = (W/L)\mu_n C_{ox}$ and (W/L) is the aspect ratio of T1 and T2. There should be **N** parallel squarer blocks in an N×N MIMO system, their outputs combined to produce **E**. Hence, **N** squarer output nodes have been shorted together to produce $I_{SQ,total}$:

$$I_{SQ,tot} = I_{SQ,1} + I_{SQ,2} + \ldots I_{SQ,N} \tag{9}$$

and **E**:

$$E = e_1^2 + e_2^2 + \cdots e_N^2 = I_{SQ,tot} \times R_L \tag{10}$$

While having large M as the ratio between transistors T1/T2 to T3/T4 increases the squaring accuracy [17], a ratio of M = 4 provides enough accuracy for our system. In order to prove this, the system is simulated while replacing the ideal squarer by the transistor level circuit. Figure 8 shows the input-output transfer characteristics of the squarer circuit with an ideal squarer, whose gain is 4 [V/V]. The offset introduced by the last term in (8) can be removed by comparing the output of the target circuit with a reference (replica) circuit. As can be seen in Fig. 8, the maximum error is limited to about 25%.

Figure 9 compares the expected system performance with a system that uses transistor-level squarer circuit. Here, a realistic model for all of the other building blocks including data converters has been included. Although we are using

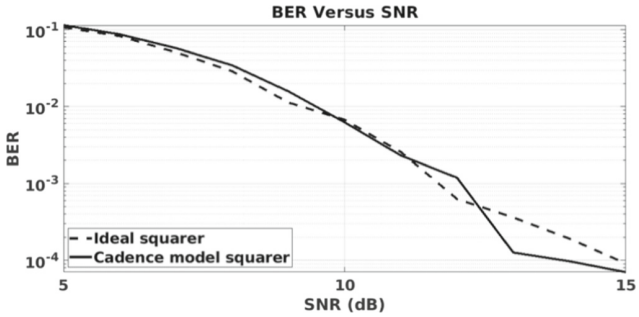

Fig. 9. System level simulation results with ideal and transistor level squarer circuit.

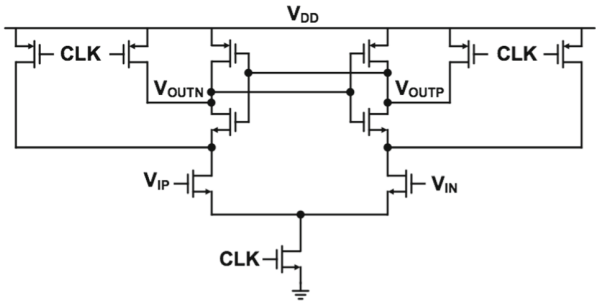

Fig. 10. StrongARM comparator used in the ADC implementation.

a replica circuit to eliminate the offset of the squarer circuit, system-level simulations show that the absolute value of the offset does not influence the system performance. Based on Fig. 9, the system performance with a transistor-level model of the squarer circuit is consistent whit that of an ideal squarer.

A careful design procedure needs to be employed to minimize the energy consumption of the circuit shown in Fig. 6, while the dynamic range is maintained. Based on this design procedure, the size of the load resistance, R_L, and the bias current of the DAC unit cells can be determined.

The 3b ADC block in Fig. 2 has been implemented using a conventional flash ADC structure, that employs StrongArm based comparator topology, shown in Fig. 10. Since PVT variations affect the output swing of the squarer, two replicas of the squarer are provided to determine the maximum and minimum reference levels for the ADC. While one of the replicas mimics the squarer when input swing is minimum, i.e., $V_{id} = 0$, the other replica produces the expected squarer output swing with $V_{id} = V_{id,max} = 180$ mV. These two voltages are then utilized to generate different reference levels for comparators using a resistor ladder.

Table 1. Area and power consumption of the proposed MCMC detector.

	DAC	Squarer	ADC	DSP	**1 Gibbs iteration**
Power (mW)	3.87	1.6	0.84	1.0	**7.3**
Area (mm^2)	0.012	0.0026	0.011	0.002	**0.028**
Power Share (%)	53.2	21.8	11.4	13.6	**100**
Area Share (%)	43.4	9.6	39.8	7.2	**100**

Table 2. Performance comparison with synthesized version in [2].

	Proposed MCMC	Synthesized version [2]
Technology (nm)	**45**	130
Power (mW)	**7.3**	N/A
Area (mm^2)	**0.028**	0.37
Clock Freq. (MHz)	**1000**	620
Throughput per Gibbs Sampler (Mbps)	**62.5**	38.75
Area Efficiency $mm^2/(\mu m \times Mbps)$	**0.21**	0.56

5 Comparison and Discussion

This Section provides a high-level comparison between the proposed MCMC detector and the state-of-the-art, especially synthesized version in [2], as well as other MIMO detector implementations [18–21].

Table 1 reports the detailed occupied area and power consumption of the proposed MCMC detector while considering biasing circuits and other auxiliary blocks. Pessimistic parasitic capacitance estimation is also included to account for routing and layout considerations. Based on the results in Table 1, the DAC arrays have the biggest contribution in the total power consumption. In terms of area, DACs and the ADC are the most dominant contributors. The entire system of Fig. 2 consumes 7.3 mW, while occupying a core area of 0.028 mm^2. It is possible to omit DAC_y to reduce area and power consumption. Since we can use the received analog data before digitizing in the RX chain prior to the DSP, DAC_y can be easily removed for the future implementations.

The MCMC presented in [2] is implemented on an FPGA, which makes it hard to have a detailed comparison with our proposed implementation. However, they have synthesized their proposed detector in a 130-nm VLSI IBM process. Therefore, we used their synthesized simulation results to compare it with that of our proposed mixed-mode MCMC detector. Table 2 compares the performance of the proposed MCMC detector with the synthesized version reported in [2]. For a fair comparison, similar system parameters have been selected (e.g., throughput per Gibbs sampler, which is defined as $f_{clk}/(N_{gs}T)$). It is assumed that

Table 3. Performance Comparison with State-of-the-Art MIMO Detectors.

MIMO detectors	[18]	[19]	[20]	[21]	**Proposed detector**
Detection algorithm	LMMSE	MMSE PIC	Relaxed K-best	MMF-LSD	**MCMC**
Technology (nm)	65	90	130	180	**45**
Supply (V)	1.2	1.2	1.3	1.8	**1.0**
Clock Freq. (MHz)	400	568	270	250	**1000**
Throughput (Mbps)	600	757	8.57	31.7	**235.3**
SNR (dB)	N/A	15	17.7	15.5	**6.5**[b]
BER	N/A	1E-2	1E-3	1E-2	**1E-5**[b]
Area (mm^2)	1.4	1.5	2.38	0.31	**0.11**
Power (mW)	266	189.1	94	56.5	**29.44**
Area efficiency (mm^2/(Mbps × μm^2))	0.552	0.245	16.4	0.302	**0.23**
Power efficiency[a] (pJ/b)	255.77	104.08	2920.6	247.5	**125.11**
Simulation (S)/ Measurement (M)	M	M	M	M	**S**

[a]Normalized to 45 nm technology [b]After FEC

$T = N_{gs} = 4$ for both cases. In addition, the area efficiency is defined to be the ratio of the occupied area of each Gibbs sampler to the throughput per Gibbs sampler. Proper scaling factors have been employed to convert power and area between the two technologies. Based on Table 2, our proposed MCMC detector area efficiency outweigh that of [2] by a factor of about 2.66. Unfortunately, there is no power consumption reported for the implementation in [2].

Table 3 provides a performance comparison between the proposed mixed-mode MCMC detector and some of the state-of-the-art MIMO detectors with VLSI implementations. The throughput of the proposed MCMC detector is calculated based on:

$$Throughput = \frac{N_{ant} \times f_{clk}}{N_{Cycles}} \tag{11}$$

where, N_{ant} is the number of antennas, f_{clk} is the clock frequency, and N_{Cycles} is the total number of cycles which is equal to:

$$N_{Cycles} = (N_{gs} \times N_{ant}) + 1 \tag{12}$$

Also, power efficiency (PE) for each reference is normalized to 45 nm technology as calculated by:

$$PE = Power \times \frac{1(V)}{Supp.(V)} \times \frac{45(nm)}{Tech.(nm)} \times \frac{1}{Throughput} \tag{13}$$

Although a crude comparison is provided in Table 3, the following points should be considered for a more reasonable comparison:

1) SNR: The reported detectors in Table 3 operate at higher SNR regime which increases the throughput and reduces the achieved BER. Hence, working at the same SNR as the MCMC detector (i.e., 6.5 dB [2]) may adversely affect the throughput, power and area efficiency of such detectors.

2) BER: Higher BER means lower reliability of the detector and thus the system. Therefore, working at different BER affects the power and area efficiency of the system. Since the proposed MCMC detector provides soft decisions, its BER can be improved using a FEC unit. Although FEC is not implemented in this work, its power and area overhead is considered in the DSP unit. Based on [2], MCMC detector with FEC can achieve a BER of 10^{-5} while SNR is only 6.5 dB. However, the reported detectors in Table 2 are working at BER less than 10^{-3}.

3) Adopted algorithm: Last but not least, the intention of this paper is to introduce an alternative way for implementing MIMO detectors which reduces the cost by moving the DSP design complexity to analog/mixed mode domain circuits. In other words, this approach can be applied to other detection algorithms that are more efficient than MCMC algorithm. For instance, multiplying is considered as a complex arithmetic operation in DSP. However, it can be simply implemented in analog domain using DACs. Almost every detection algorithm includes multiplication operation. Hence, the proposed approach can be utilized to reduce their DSP complexity and thus, reduce the total system cost.

6 Future Works

The possible future works can be divided into two categories of algorithm-level improvements and circuit-level improvements.

6.1 Algorithm-Level Improvements

In this paper, the proposed approach is applied to the conventional MCMC algorithm that is discussed in [2]. However, this approach is also suitable for any other algorithm that includes multiplication operations.

For instance, stochastic iterative MIMO (SIM) detector introduced in [22] works based on the bit-flop MCMC method too. Comparing the hardware efficiency of [22] with that of [2] shows a superior performance for [22] by a factor of more than 5. While in conventional MCMC detector the Gibbs sample is updated by the conditional probabilities calculated in the DSP of the detector, SIM in [22] updates the Gibbs sampler directly using decoded bits from channel decoder. Since the SIM works based on the bit-flop method as in the conventional MCMC, the analog/mixed mode circuits working after DSP unit could be more or less the same. In other words, by applying the same approach to the DSP unit in [22] to move some of the complicated processes to the analog domain, throughput of the system significantly improves as compared with this paper.

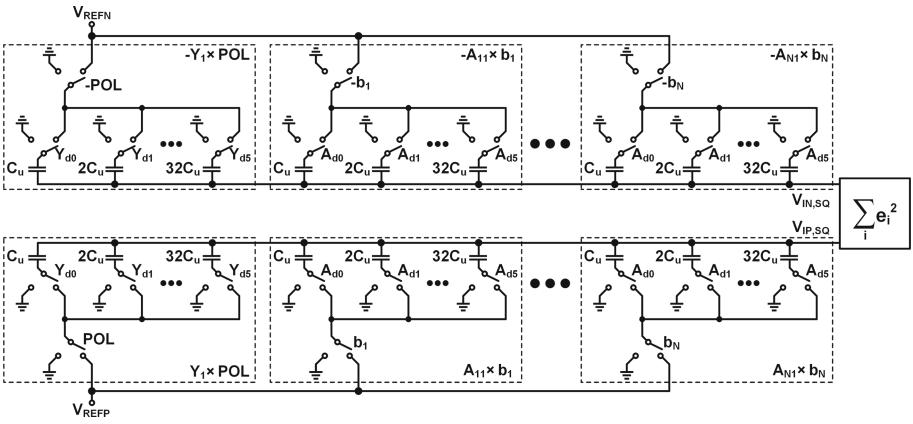

Fig. 11. Proposed charge-based architecture of the merged DAC-multiplier-summer for future implementations.

6.2 Circuit-Level Improvements

As previously discussed, a crude and very primitive circuit solution is offered here and can be improved by employing different circuit techniques and architectures. Since DAC arrays have a big contribution to both total power consumption and occupied area, more efficient DAC architectures are highly desirable. Hence, moving to a charge redistribution-based DAC which is inherently high speed, can significantly improve the efficiency of the system as compared with the current CML-based DAC implementation.

Figure 11 shows the suggested charge-based implementation of the DAC arrays to generate the **e** function for each channel. There are **N** copies of 6-bit binary weighted cap array which convert the digital [**A**] for each channel back to the analog domain. An extra switch for each capacitive bank is provided to simply take care of the multiplication function. Another capacitive bank is added to implement DAC_y in Fig. 2. Moreover, POL signal controls the summation polarity in (3) similar to the CML DAC_y in Fig. 6. Since the unit cap, Cu, can be as small as 1 or 2 fF in a 6-bit DAC, total occupied area and power consumption will be remarkably reduced as compared with the CML-based DACs. Also, as discussed earlier in Sect. 5, DAC_y array can be neglected if the received data is directly utilized before digitization prior to the DSP unit.

The currently employed CML-based architecture in Fig. 6 and the suggested charge-redistribution-based DAC in Fig. 11 are calculating the e function based on (3). As shown in both of these figures, $N + 1$ DAC arrays in each channel are operating for $N + 1$ clock cycles to perform 1 Gibbs iteration. Hence, we can define the switching activity (SA) based on the number of DACs that are operating in each Gibbs iteration:

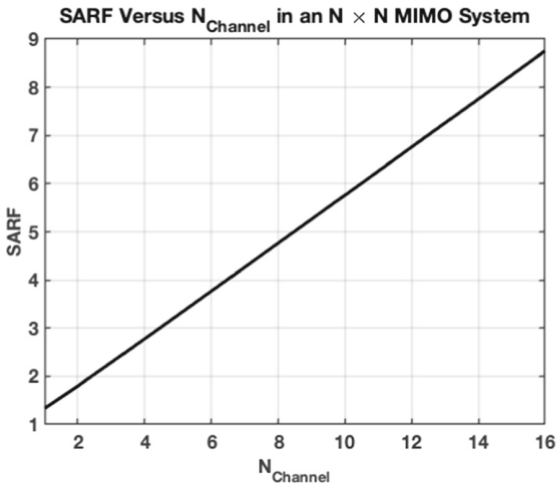

Fig. 12. Switching activity reduction factor (SARF) versus number of channels in an $N \times N$ MIMO system.

$$\begin{aligned} SA_{conv} &= (N+1)_{(DAC)} \times N_{(channel)} \times (N+1)_{cycles} \\ &= N \times (N+1)^2 \end{aligned} \tag{14}$$

where, SA_{conv} shows the switching activity of the charge-redistribution-based DAC in Fig. 11. However, based on (4), finding $[\mathbf{e}]_{(\mathbf{k})}$with k > 1 is simply the addition of $[\mathbf{e}]_{(\mathbf{k-1})}$ and the second term in (4). Given that generating the second term in (4) requires only one DAC per channel, it is feasible to turn off all other N DACs in each channel for K > 1 by utilizing a sample and hold (S/H) circuit. Hence, a S/H circuit which is easily compatible with charge-redistribution-based DAC, can be employed to hold the value of $[\mathbf{e}]_{(\mathbf{k-1})}$ for $[\mathbf{e}]_{(\mathbf{k})}$ calculation and significantly improve the system power efficiency. In other words, except for the first clock in which $N+1$ DACs are operating in each channel, the next N cycles only need one DAC per channel to calculate the error function based on the stored value in S/H circuit. Therefore:

$$SA_{S/H}(i) = \begin{cases} (N+1)_{(DAC)} \times N_{(channel)} & \text{if } i = 1 \\ 1_{(DAC)} \times N_{(channel)} & \text{if } i = 2, 3, ..., N+1 \end{cases} \tag{15}$$

where, $SA_{S/H}(i)$ shows the switching activity of the charge-redistribution-based DAC with S/H for i_{th} clock cycle. Considering all of the $N+1$ cycles, $SA_{S/H}$ for 1 Gibbs iteration is equal to:

$$SA_{S/H} = N \times (2N+1) \tag{16}$$

In order to compare SA_{conv} and $SA_{S/H}$, we can define the switching activity reduction factor (SARF) as follows:

$$SARF = \frac{SA_{conv}}{SA_{S/H}} = \frac{(N+1)^2}{2N+1} \tag{17}$$

As shown in Fig. 12, employing a S/H with the charge-redistribution-based DAC of Fig. 11 significantly reduces the switching activity in an $N \times N$ MIMO system by increasing the number of channels. It should be noted that the number of channels does not necessarily represent the number of antennas in a MIMO system. In fact, the effective number of channels can be increased using different types of modulations such as 16/64/256-QAM.

7 Conclusion

An analog/mixed-mode approach for designing MCMC MIMO detectors is presented. The proposed system relaxes some of the complexities in the design of conventional digital detectors, especially by moving some high-speed operations to analog domain. While the proposed system consumes 29.3 mW, the proposed detector operates at 1 GHz clock frequency. Achieving a throughput of 235.3 Mbps, the circuit occupies 0.11 mm^2 Silicon area (estimated). Moreover, a charge-redistribution-based implementation is presented for future works that can significantly enhance the power and area efficiency of the current implementation. The proposed approach can be applied for implementing similar processing systems in which speed and energy efficiency are the concerns.

References

1. Foschini, G.J. : Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas. Bell Labs Tech. J. **1**, 41–59 (1996). https://doi.org/10.1002/bltj.2015
2. Laraway, S.A., Farhang-Boroujeny, B.: Implementation of a Markov chain Monte Carlo based multiuser/MIMO detector. IEEE Trans. Circuits Syst. I Regular Papers **56**(1), 246–255 (2008). https://doi.org/10.1109/TCSI.2008.925891
3. Verdu, S.: Minimum probability of error for asynchronous Gaussian multiple-access channels. IEEE Trans. Inf. Theory **32**(1), 85–96 (1986). https://doi.org/10.1109/TIT.1986.1057121
4. Zhu, H., Shi, Z. and Farhang-Boroujeny, B.: MIMO detection using Markov chain Monte Carlo techniques for near-capacity performance. In: Proceedings. (ICASSP 2005). IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, p. 1017 (2005). https://doi.org/10.1109/ICASSP.2005.1415885
5. Farhang-Boroujeny, B., Zhu, H., Shi, Z.: Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems. IEEE Trans. Sig. Process. **54**(5), 1896–1909 (2006). https://doi.org/10.1109/TSP.2006.872539
6. Hedstrom, J.C., Yuen, C.H., Chen, R., Farhang-Boroujeny, B.: Achieving near MAP performance with an excited Markov chain Monte Carlo MIMO detector. IEEE Trans. Wirel. Commun. **16**(12), 7718–7732 (2017). https://doi.org/10.1109/TWC.2017.2750667

7. El Gamal, H., Hammons, A.R.: A new approach to layered space-time coding and signal processing. IEEE Trans. Inf. Theory **47**(6), 2321–2334 (2001). https://doi.org/10.1109/18.945250
8. Aghighi, A., Farhang-Boroujeny, B. and Tajalli, A.: Energy and area efficient mixed-mode MCMC MIMO detector. In: 2020 IFIP/IEEE 28th International Conference on Very Large Scale Integration (VLSI-SOC), Salt Lake City, UT, USA, pp. 105–110 (2020). https://doi.org/10.1109/VLSI-SOC46417.2020.9344098
9. Auras, D., Deidersen, U., Leupers, R., Ascheid, G.: A parallel MCMC-Based MIMO detector: VLSI design and algorithm. In: Claesen, L., Sanz-Pascual, M.-T., Reis, R., Sarmiento-Reyes, A. (eds.) VLSI-SoC 2014. IAICT, vol. 464, pp. 149–169. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25279-7_9
10. Deidersen, U., Auras, D., Ascheid, G.: A parallel VLSI architecture for Markov chain Monte Carlo based MIMO detection. In: Proceedings of the 23rd ACM International Conference on Great Lakes Symposium on VLSI, pp. 167–172 (2013). https://doi.org/10.1145/2483028.2483084
11. Tribble, S.D. : Markov chain Monte Carlo algorithms using completely uniformly distributed driving sequences. Ph.D. thesis, Stanford University (2007)
12. Aghighi, A., et al.: CMOS amplifier design based on extended GM/ID methodology. In: 2019 17th IEEE Int. New Circuits and Systems Conference (NEWCAS), pp. 1–4 (2019). https://doi.org/10.1109/NEWCAS44328.2019.8961308
13. Atkinson, J., Aghighi, A., Anderson, S., Bailey, A., Crane, M., Tajalli, A.: Multi-stage current-steering amplifier design based on extended GM/ID methodology. In: 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 129–132 (2019). https://doi.org/10.1109/MWSCAS.2019.8885313
14. Aghighi, A., Tabib-Azar M., Tajalli, A.: An ULP self-supplied brain interface circuit. In: 2020 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SOC), Salt Lake City, UT, USA, pp. 100–104 (2020). https://doi.org/10.1109/VLSI-SOC46417.2020.9344092
15. Aghighi, A., Alameh, A.H., Taherzadeh-Sani, M., Nabki, F.: A 10-Gb/s low-power low-voltage CTLE using gate and bulk driven transistors. In: IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp. 217–220 (2016). https://doi.org/10.1109/ICECS.2016.7841171
16. Aghighi, A., Tajalli, A., Taherzadeh-Sani, M.: A low-power 10 to 15 Gb/s common-gate CTLE based on optimized active inductors. In: 2020 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SOC), Salt Lake City, UT, USA, pp. 100–104 (2020). https://doi.org/10.1109/VLSI-SOC46417.2020.9344076
17. Gerosa, A., Soldà, S., Bevilacqua, A., Vogrig, D., Neviani, A.B.: An energy-detector for noncoherent impulse-radio UWB receivers. IEEE Trans. Circuits Syst. I Regular Papers **56**(5), 1030–1040 (2009). https://doi.org/10.1109/TCSI.2009.2016125
18. Chen, X., et al.: Flexible, efficient multimode MIMO detection by using reconfigurable ASIP. IEEE Trans. Very Large Scale Integration (VLSI) Syst. **23**(10), 2173–2186 (2014). https://doi.org/10.1109/TVLSI.2014.2361206
19. Studer, C., Fateh, S., Seethaler, D.: ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation. IEEE J. Solid-State Circuits **46**(7), 1754–1765 (2011). https://doi.org/10.1109/JSSC.2011.2144470
20. Chen, S., Zhang, T., Xin, Y.: Relaxed K-best MIMO signal detector design and VLSI implementation. IEEE Trans. Very Large Scale Integration (VLSI) Syst. Regular Papers **15**(3), 328–337 (2007). https://doi.org/10.1109/TVLSI.2007.893621

21. Myllyla, M., Cavallaro, J.R., Juntti, M.: Architecture design and implementation of the metric first list sphere detector algorithm. IEEE Trans. Very Large Scale Integration (VLSI) Syst. **19**(5), 895–899 (2010). https://doi.org/10.1109/TVLSI.2010.2041800
22. Chen, J., Hu, J., Sobelman, G.E.: Stochastic iterative MIMO detection system: algorithm and hardware design. IEEE Trans. Circuits Syst. I Regular Papers **62**(4), 1205–1214 (2015). https://doi.org/10.1109/TCSI.2015.2390558