2021 IEEE 39th International Conference on Computer Design (ICCD) | 978-1-6654-3219-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCD53106.2021.00064

Reconfigurable Array for Analog Applications

Ziyi Chen
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, PA
Email: ziyi.chen@drexel.edu

Ioannis Savidis
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, PA
Email: isavidis@coe.drexel.edu

***Abstract*—In this paper, a novel field-programmable analog array (FPAA) fabric consisting of a 6x6 matrix of configurable analog blocks (CABs) is proposed. The implementation of programmable CABs eliminates the use of fixed analog subcircuits. A unique routing strategy is developed within the CAB units that supports both differential and single-ended mode circuit configurations. The bandwidth limitation due to the routing switches of each individual CAB unit is compensated for through the use of a switch-less routing network between CABs. Algorithms and methodologies are developed to facilitate rapid implementation of analog circuits on the FPAA. The proposed FPAA fabric provides high operating speeds as compared to existing FPAA topologies, while providing greater configuration in the CAB units as compared to switch-less FPAAs. The FPAA core includes 498 programming switches and 14 global switch-less interconnects, while occupying an area of 0.1 mm^2 in a 65 nm CMOS process. The characteristic power consumption is approximately 24.6 mW for a supply voltage of 1.2 V. Circuits implemented on the proposed FPAA fabric include operational amplifiers (op amps), filters, oscillators, and frequency dividers. The reconfigured bandpass filter provides a center frequency of approximately 1.5 GHz, while the synthesized ring-oscillator and frequency divider support operating frequencies of up to 500 MHz.**

I. Introduction

The reconfigurability of an integrated circuit (IC) is an area of research interest. Due to the long design time and the high cost of fabrication for fully customized ICs, the rapid development of an IC at low cost appeals to many IC manufacturers and researchers. One solution that provides rapid and low cost IC development is through implementation on a programmable fabric. The field programmable gate array (FPGA) is well developed and is applied extensively in both commercial products and research. However, minimum effort has been applied to the development of an analog counterpart to the FPGA; the field-programmable analog array (FPAA). While the FPGA utilizes gate arrays and look-up tables (LUT) to implement logical functions, analog circuit blocks are difficult to generalize into a uniform architecture. Existing FPAA fabrics generally target a specific application including a G_m-C filter [1], vector-matrix multiplication (VMM) [2], and a switch-capacitor filter [3], where the implementation of the circuit is based on a fixed analog structure.

The centerpiece of the FPAA is the configurable analog block (CAB). However, the elements within each CAB vary depending on the target application and the given specifications. Generally, each individual CAB contains multiple coarse-grain analog blocks, routing switches, and passive devices in order to provide programmability and tunability. The topology of a CAB that targets filter applications, for example, includes multiple G_m cells connected in parallel to provide control of the transconductance.

In order to provide connectivity between different CABs and increase the configurability of the CAB array, both switch-based [6] and switch-less [1] routing approaches are utilized. While the switch-based routing network provides the highest degree of programmability, the placement of routing switches on the signal path results in massive bandwidth limitations. The switch-less routing strategy, however, reduces the effect on the bandwidth due to the routing switches, but at a cost of less programmability due to the fixed interconnect.

In this paper, a novel FPAA CAB is proposed that is fully programmable with no prior functionality set before programming. In addition, routing switches between separate CABs are eliminated to further reduce the effect on the bandwidth of the overall system. The proposed FPAA is used to implement an op amp, continuous-time filter, ring-oscillator, and frequency divider. The architecture of the FPAA is discussed in detail in Section II. A methodology to rapidly implement an analog circuit on the FPAA is proposed in Section III. The utilization of the reconfigurable multi-CAB system to implement a second-order biquad filter, a differential mode ring-oscillator, and a frequency divider is described in Section IV. A comparison with existing FPAA fabrics is provided in Section V. Some concluding remarks on the proposed FPAA fabric are provided in Section VI.

II. Architecture of FPAA

The proposed FPAA architecture is organized into a 6x6 CAB matrix as shown in Fig. 1, where each column of CABs is directly accessed through seven I/O ports. There are two different topologies of CABs within the array to facilitate both single-ended and differential mode operation. A global routing network allows for both positive or negative feedback across multiple CABs to implement more complex analog topologies. In addition, the bandwidth limitation due to the routing switches is reduced by implementing switch-less interconnects between CABs.

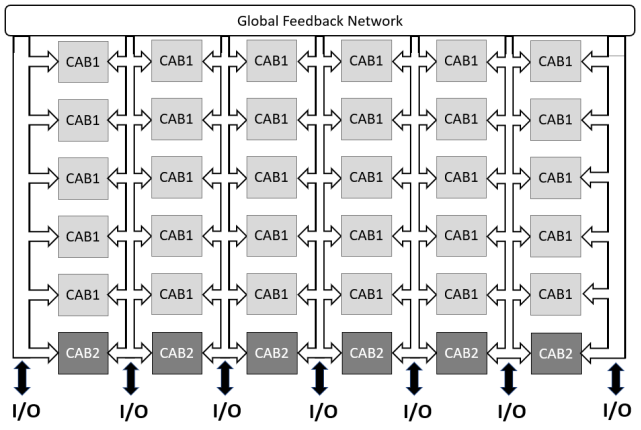


Fig. 1. FPAA Architecture: The FPAA core is organized into a 6x6 CAB matrix. There are two types of CABs, one to support differential mode operation and the other to support single-ended operation.

978-1-6654-3219-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCD53106.2021.00064

A. CAB Architecture

The CAB unit is the mid-level building block of the FPAA fabric, which is reconfigured into variants depending on the specifications of the implemented circuit. In this paper, the programmability of the CABs is explored, with the goal of achieving a maximum number of reconfigurable topologies with a minimum number of transistors and switches. As differential mode analog signal processing usually requires NMOS and PMOS pairs, there are four categories of topologies to consider as shown in Fig. 2. To implement the four distinct configurations of a generic FPAA architecture, two topologies of reconfigurable transistor pairs are proposed as shown in Fig. 3, where routing switches are added as controllable interconnects between the gate and drain terminals.

Based on the generic architecture of the NMOS and PMOS pair, two different topologies of CABs are developed to support both differential mode and single-ended mode operation. As shown in Fig. 4, the first CAB topology utilizes the basic structure of a five transistor operational transconductance amplifier (OTA), which includes one NMOS differential pair, one PMOS active load pair, and one NMOS current source. In addition, 13 routing switches are included to allow for the full programmability of the CAB. The switches S1 to S5 are used to program the NMOS differential pair, where S2 and S4 are utilized for negative feedback, and S1 and S3 are used to provide positive feedback or form the regeneration loop. The switches S6 to S10 program the PMOS differential pair, where S7 and S9 form a diode-connected load, and S6 and S8 implement the regeneration loop for the PMOS pair. The S11 switch controls the common mode feedback, while S12 and S13 completely turn off the CAB by connecting the gate of the PMOS current source to VDD and the gate of the NMOS current source to ground. Note that the CAB topology is undefined before programming, which therefore, increases the flexibility of the proposed FPAA fabric. The first CAB is reconfigured into different types of op amps, including a differential pair with a current source load and a differential pair with a diode-connected load. The proposed CAB topology differs from prior CAB topologies, where the analog sub-circuit is usually fixed and the programmability is achieved by routing between predefined sub-blocks. The limitation on the configurability of the FPAA is addressed by making the analog sub-blocks fully programmable.

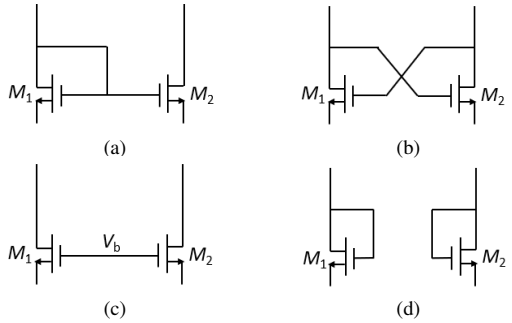


Fig. 2. Possible topologies achieved with a single NMOS transistor pair include a (a) current mirror, (b) regeneration loop, (c) current source pair, and (d) diode-connected pair. The same topologies are implemented with the PMOS transistor pair.

B. Routing Wire

In order to eliminate routing switches between CABs and increase the peak bandwidth, a global switch-less routing technique is implemented on the FPAA. The input and output terminals of the CABs in each column are tied together using single global interconnects, one for the inputs and one for the outputs. To explicitly define the polarity of the differential signals applied to the CABs, the switch-less networks are labeled to differentiate between the positive path, p-path (c#p), and the negative path, n-path (c#n), as shown in Fig. 5. The positive terminal of each CAB is directly connected to the p-path only, and the negative terminal of each CAB is routed to the n-path only. With precise polarity notation, the propagation of a differential signal between CABs is easier to track.

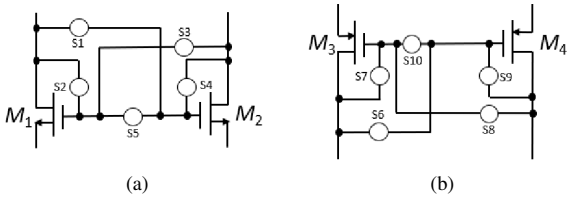


Fig. 3. Schematic of reconfigurable NMOS and PMOS pairs. Each pair utilizes five routing switches that are reconfigurable to form a current mirror, regeneration loop, current source pair, and diode-connected pair.

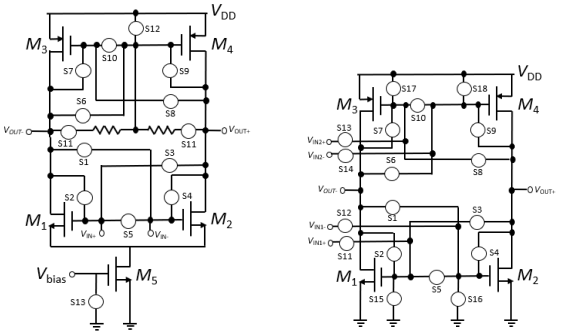


Fig. 4. Schematic of (a) a differential mode CAB (CAB1), and (b) a single-ended mode CAB (CAB2). CAB1 includes 13 routing switches in addition to four possible configurations of the transistor pair. S11 is added to provide common-mode feedback, while S12 and S13 are used to completely turn off CAB1. CAB2 includes 18 routing switches. S11 to S14 isolate the global switch-less interconnects and the gates of transistors M_1 to M_4. S15 to S18 turn off the CAB.

C. Feedback Network

Feedback, positive and/or negative, is widely used in analog and RF circuits. While negative feedback is utilized to implement amplifiers with greater performance and less sensitivity to parameter variations, positive feedback is utilized with oscillators and dynamic latches. In addition to the routing switches included in each CAB to configure either positive or negative feedback, programmable feedback paths across multiple CABs are also provided to increase the flexibility of the multi-CAB topology. To provide feedback between any two columns of CABs, a global feedback network structure as shown in Fig. 5 is developed. The global feedback networks include two parallel paths to support differential mode operation. Since the switch-less interconnects include an explicitly annotated polarity, negative feedback is provided by connecting a positively labeled output with a negatively labeled input and a negatively labeled output with a positively labeled input. Similarly, positive feedback is achieved by connecting inputs and outputs with the same polarity. Since

each column of CABs is connected to the global feedback network, the positive or negative feedback between any two columns of CABs is easily implemented.

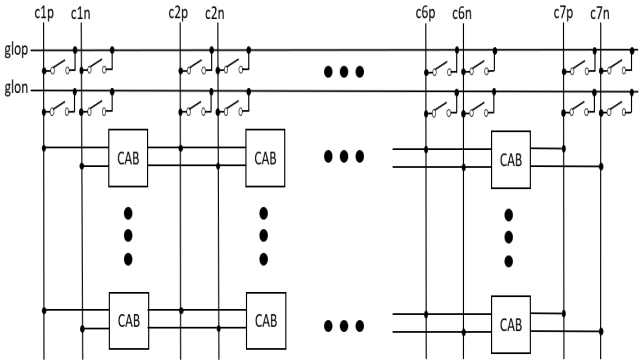


Fig. 5. Schematic of global feedback network. The feedback network consists of 2x28 switches that support multi-CAB level feedback throughout all six columns of the CAB matrix.

III. Synthesis Methodology

In this section, an automated flow for FPAA synthesis is described. In addition, a CAB-oriented programming algorithm is formulated to rapidly implement analog circuits on the FPAA.

A. Software Interface

While computer-aid design (CAD) tools for FPGAs have been extensively developed, there are no general methodologies or CAD tools developed for FPAA fabrics. The tools utilized for FPAAs are typically customized to a specific CAB topology, array architecture, and/or target application. A design flow is proposed to rapidly generate programming keys for all the CABs within the array. The proposed flow is shown in Fig. 6. The synthesis and configuration of the FPAA is divided into three steps: 1) programming the CABs, 2) programming the CAB biases, and 3) programming the global feedback network. For the configuration of all CABs within the matrix array, each CAB is labeled with row-major order. The configuration of the FPAA is stored in a python script. The bias condition for each CAB is then determined. Finally, the configuration of the global feedback network is performed, and a configuration list for all the elements within the FPAA is generated. The configuration list is compiled into an ocean script that is directly used for SPICE simulation.

B. FPAA Programming Procedure

The mapping of an analog circuit onto the FPAA fabric is divided into two steps: 1) placement and 2) routing. The placement step maps the lowest-level analog sub-blocks into different CABs, while the routing procedure activates switches within and between CABs. In the proposed FPAA fabric, swith-less interconnects are used as the forward signal path, while the routing switches between CABs are components of the feedback network. The matrix architecture allows the user to access CABs in each column directly. The location of the output port is also flexible and is set according to the location of the inputs. The general synthesis strategy is presented as Algorithm 1. Starting with the placement procedure, the analog sub-circuits are first mapped into different CABs. The general activation and selection of each CAB unit is dependent on the mapping of the devices of the target analog circuit being implemented. With the utilization of several predefined configurations of the CABs, users are able to rapidly implement desired analog circuits with single or multiple CABs.

Algorithm 1: Synthesis algorithm for FPAA

Input: queue Q containing all sub-blocks(C_1, C_2,.. C_N)
Output: switch list S
Function `Placement()`**:**
    get the length of Q
    select equal number of CABs
Function `Routing()`**:**
    initialize empty switch list S
    for *each CAB in array* **do**
        if *CABi is selected* **then**
            CabMode = pop(Q)
            S.pushBack(getKeys(CabMode))
        else
            CabMode = OFF
            S.pushBack(getKeys(CabMode))
return S

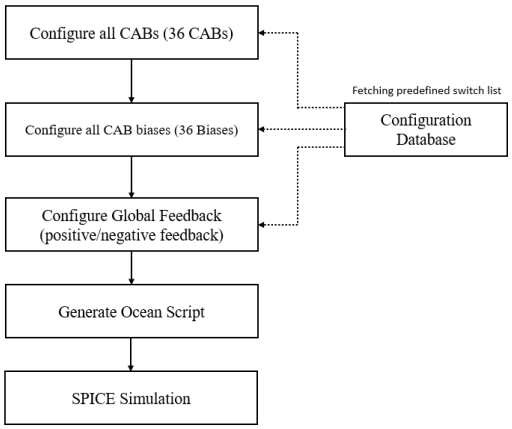


Fig. 6. Design flow of implementing analog circuits on the FPAA. After determining the configuration of the CABs, the CAB biases, and the global feedback network, a list containing all the configuration data is generated. The configuration list is then compiled into an ocean script for simulation.

IV. Utilization of the Multi-CAB Array

In this section, a multi-CAB system is utilized to implement a second order biquad filter as described in Section IV-A, a differential mode ring oscillator as described in Section IV-B, and a frequency divider (quadrature phase generator) as described in Section IV-C.

A. Second Order Biquad Filter

A biquad filter is a type of G_m-C filter that is found in a large number of RF and mixed-signal systems due to the tunability and robustness provided by the topology. The FPAA is reconfigured to implement a second order biquad filter [4] to test the performance of the multi-CAB system. The schematic of the second order biquad filter is shown in Fig. 7, which consists of four G_m cells. To provide a tunable G_m, the number

of CABs activated in a given column is varied to achieve a variable transconductance. In order to maintain the stability of the DC operating points, the CAB1s are reconfigured into a single stage op amp with resistive common-mode feedback. Since each G_{m} cell consumes one column of the CAB matrix, a second order biquad filter requires four columns of CAB1s in total. The connection between the CABs is achieved by utilizing switch-less routing interconnects to minimize the effects on the bandwidth while the negative feedback path for G_{m2} is established through routing switches (S_2 and S_4 in Fig. 4a) that are embedded within the CAB. The positive feedback path from G_{m4} to G_{m2} is established through the global feedback network. The band pass transfer function of a second-order biquad filter is given by

$$\frac{V_{out}}{V_{in}} = \frac{g_{m1}\frac{C_2}{C_1}S}{S^2 + \frac{g_{m2}}{C_1}S + \frac{g_{m3}g_{m4}}{C_1C_2}} \tag{1}$$

$$= K\frac{\frac{\omega_0}{Q}S}{S^2 + \frac{\omega_0}{Q}S + \omega_0^2} \text{ for} \tag{2}$$

$$K = \frac{g_{m1}}{g_{m2}},\ \frac{\omega_0}{Q} = \frac{g_{m2}}{C_1} \text{ and } \omega_0^2 = \frac{g_{m3}g_{m4}}{C_1C_2}, \tag{3}$$

where K represents the peak gain, ω_0 the center frequency, and ω_0/Q the bandwidth. For the implementation of the biquad filter, the peak gain is tuned to the target center frequency by adjusting the value of G_{m2}. The results of the simulation of the second-order biquad filter are shown in Fig. 8, with a minimum peak gain of -9 dB and a maximum peak gain of 8 dB.

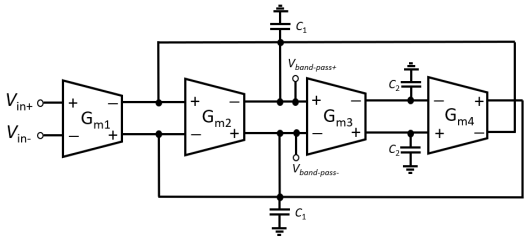


Fig. 7. Schematic of a second-order biquad filter. Each G_{m} cell consists of one or multiple CABs from one column of the CAB matrix. The negative feedback for G_{m2} is set by the feedback switches within the CAB. The positive feedback between G_{m3} and G_{m4} is achieved through the global feedback network.

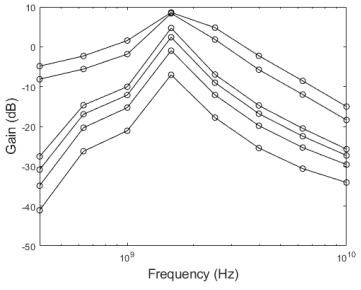


Fig. 8. Simulation of the second-order biquad filter, where the bandwidth and peak gain are tuned by adjusting G_{m1} and G_{m2}.

B. Ring Oscillator

A ring oscillator is a primary component of phase-locked loops (PLLs) and data recovery circuits of serial data communication systems. While single-ended inverter-based ring oscillators are easily generated in the digital domain, the differential mode CABs allow for the configuration of more complex differential mode ring oscillators. First, the desired number of CABs from a specific row of the CAB matrix is selected based on the target frequency specification, while unused CABs of the other rows are deactivated. The selected individual CABs are programmed into single stage op amps with diode connected PMOS loads. Diode-connected loads are chosen due to the superior linearity provided as compared to resistive loads. As a switch-less routing strategy between CABs is utilized, no routing switches are needed for the forward propagation of the signal. Therefore, the global routing network is utilized to form the feedback path across multiple CABs. An advantage of using a differential mode ring oscillator is that the use of an even number of stages is permitted, while an inverter-based ring oscillator only allows an odd number of stages. Therefore, the proposed FPAA topology allows for the implementation of 3 to 6 stages in a six column CAB floorplan. The transient simulation of the FPAA configured as a six-stage differential ring oscillator is shown in Fig. 10a. The oscillating frequency as a function of the number of stages is plotted in Fig. 10b, where a maximum frequency of 478 MHz is generated with three stages and a minimum frequency of 207 MHz is generated with six stages.

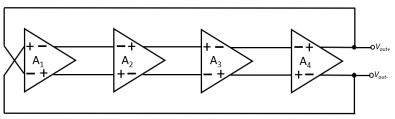
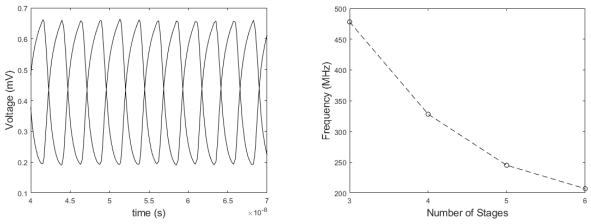


Fig. 9. Schematic of a four-stage ring oscillator. An even number of stages is possible due to the utilization of differential mode signals.



(a) (b)

Fig. 10. Simulation of a differential ring oscillator to characterize (a) the transient response of six-stages, which oscillates at a frequency of 207 MHz, and (b) the oscillating frequency as a function of the number of stages, which are programmed from three to six.

C. Frequency Divider

The differential mode divide-by-2 frequency divider is widely used in mixed-signal and RF circuits including for analog to digital conversion (ADC), clock generation, and quadrature amplitude modulation (QAM). A typical divider is implemented from two differential dynamic latches as shown in Fig. 11. The proposed FPAA supports the implementation of a frequency divider by first configuring two differential mode CABs (CAB1) from two adjacent columns. Each CAB is programmed to generate a differential dynamic latch, where the PMOS transistor pair forms a regeneration loop while the input NMOS pair maintains separate input ports. In addition, a positive feedback path across two CABs is established through the global feedback network. Note that each dynamic latch requires a clock signal with a 180 degree phase shift as input to the gate of transistor M_5. An additional signal path is added to

TABLE I
COMPARISON WITH PRIOR WORK

	Technology	Architecture	Operating Freq.	Area	Num. of CABs	Application	CAD
[3]	130 nm	Switch-less	82 MHz	1 mm^2	7	Filter	-
[6]	350 nm	FG-based	57 MHZ	9 mm^2	32	Receiver / Speech Processor	Simulink
[7]	90 nm	Switch-less	500 Hz	-	27	Filter	-
[5]	350 nm	FG-based	18 MHz	84 mm^2	108	Analog-to-digital Conversion	Verilog / Matlab
[3]	2 µm	Switch-based	10 MHz	626 µm x 750 µm	40	Analog Arithmetic Operations / Filter	-
[8]	350 nm	Switch-based	20 MHz	0.9 mm^2	25	Translinear Computing	-
[9]	350 nm	FG-based	200 KHz	9 mm^2	18	Translinear Computing	Verilog / Matlab
This Work	65 nm	hybrid	1.5 GHz	0.1 mm^2	36	Filter / Oscillator / Divider	Python / Ocean

support such a requirement. Therefore, the gate of M_5 is either connected to a DC bias or to an input signal terminal. The results from transient simulation of the divide-by-2 frequency divider are shown in Fig. 12, where a 200 MHz signal is used to produce four 100 MHz signals phase shifted by 90 degrees.

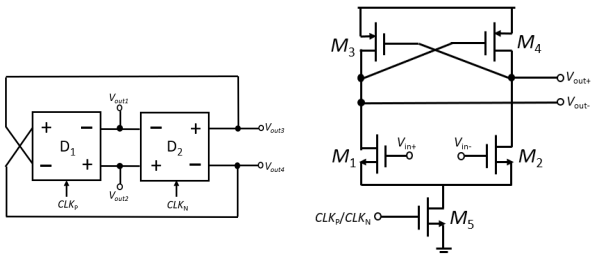


Fig. 11. Schematic representation of (a) the frequency divider configured with two CABs, where both CABs are programmed to implement differential dynamic latches, and (b) the implemented differential dynamic latch.

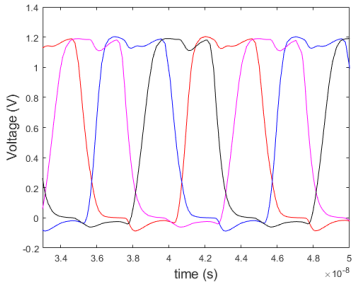


Fig. 12. Transient simulation of a divide-by-2 frequency divider with a 200 MHz input frequency. The four outputted signals are 90 degree phase shifted, with all four signals providing full swing (1.2 V).

V. COMPARISON TO PRIOR FPAA

The performance of the proposed FPAA is compared with prior work, with primary charateristics summarized in Table I. The most important metrics to evaluate an FPAA include the bandwidth, the variability of the circuits to implement, and the area. Greater circuit variability usually requires additional routing switches, which results in a smaller bandwidth due to the parasitic impedance of the routing network. The cost in bandwidth due to increased circuit variability is improved by implementing a hybrid routing technique, where both switch-based and switch-less connections are used. Routing switches between CABs are avoided, while providing greater variability within individual CABs by considering each differential NMOS or PMOS pair as a programmable unit. The proposed fully programmable CABs provide significant reconfiguration in addition to positive or negative feedback paths. The area of each CAB is also more compact as compared to a floating gate based CAB, where large interconnect switches are needed [5].

VI. CONCLUSION

In this paper, a 6x6 CAB matrix is developed in a 65 nm CMOS technology. The multi-CAB topology supports analog signal filtering, frequency division, and clock signal generation. The proposed architecture integrates both switch-less and switch-based routing to increase the reconfigurability of the FPAA and to minimize the effect on the bandwidth. The CABs are designed to be fully programmable while occupying a minimum area. Feedback paths are formed by routing switches within individual CABs or between multiple CABs utilizing the global feedback network. A synthesis methodology is also described to aid in the implementation of CAB-based analog and RF circuits.

The FPAA core consists of 36 CABs, 14 switch-less interconnects, and 498 programming switches while occupying 0.1 mm^2 of area. For a reconfigured bandpass filter, the tunable range of the peak gain is from -9 dB to 8 dB. For the implemented ring-oscillator, the operating frequency is between 207 MHz and 478 MHz, with a maximum peak-to-peak voltage swing of 0.45 V. The divider supports frequency division of a signal of up to 200 MHz bandwidth.

REFERENCES

[1] J. Becker, F. Henrici, S. Trendelenburg, M. Ortmanns and Y. Manoli, "A Field-Programmable Analog Array of 55 Digitally Tunable OTAs in a Hexagonal Lattice," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2759-2768, Dec. 2008.

[2] C. R. Schlottmann and P. E. Hasler, "A Highly Dense, Low Power, Programmable Analog Vector-Matrix Multiplier: The FPAA Implementation," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 403-411, Sept. 2011.

[3] B. Pankiewicz, M. Wojcikowski, S. Szczepanski and Yichuang Sun, "A Field Programmable Analog Array for CMOS Continuous-time OTA-C Filter Applications," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 2, pp. 125-136, Feb. 2002.

[4] P. Crombez, J. Craninckx, P. Wambacq and M. Steyaert, "A 100-kHz to 20-MHz Reconfigurable Power-Linearity Optimized G_m–C Biquad in 0.13-µm CMOS," *IEEE Transactions on Circuits and Systems II*: Express Briefs, vol. 55, no. 3, pp. 224-228, March 2008.

[5] R. B. Wunderlich, F. Adil and P. Hasler, "Floating Gate-based Field Programmable Mixed-signal Array," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 8, pp. 1496-1505, Aug. 2013.

[6] A. Basu et al., "A Floating-Gate-Based Field-Programmable Analog Array," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 9, pp. 1781-1794, Sept. 2010.

[7] M. S. Diab and S. A. Mahmoud, "Field Programmable Analog Arrays for Implementation of Generalized nth-order Operational Transconductance Amplifier-C Elliptic Filters," *ETRI Journal.*, vol. 42, no. 4, pp. 534–548, Aug. 2020.

[8] D. Fernandez, L. Martinez-Alvarado and J. Madrenas, "A Translinear, Log-domain FPAA on Standard CMOS Technology," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 2, pp. 490-503, Feb. 2012.

[9] C. R. Schlottmann, D. Abramson and P. E. Hasler, "A MITE-based Translinear FPAA," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 1-9, Jan. 2012.