

# **A Multiple Input Floating Gate based Arithmetic Logic Unit with a Feedback Loop for Digital Calibration**

Zhou Zhao<sup>1</sup>, Ashok Srivastava<sup>1\*</sup>, Lu Peng<sup>1</sup>, and Saraju P. Mohanty<sup>2</sup>

<sup>1</sup> Division of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, 70803,  
U.S.A. {zzhao13, eesriv, lpeng}@lsu.edu

<sup>2</sup> Department of Computer Science and Engineering, University of North Texas, Denton, TX, 76207, U.S.A.,  
saraju.mohanty@unt.edu

\* corresponding author: Ashok Srivastava

Address:

Louisiana State University  
Division of Electrical and Computer Engineering  
Patrick F. Taylor Hall  
Baton Rouge, LA, 70803, U.S.A.

Office : Room 3316S  
Fax : 225-578-5200  
Email : eesriv@lsu.edu

Date of Receiving: **to be completed by the Editor**

Date of Acceptance: **to be completed by the Editor**

# A Multiple Input Floating Gate based ALU with a Feedback Loop for Digital Calibration

Zhou Zhao, Ashok Srivastava, Lu Peng, and Saraju P. Mohanty

**Abstract** — We present the design of a 32-bit ALU using multiple input floating gate MOSFETs. Using the reconfigurable surface potential applied on the device, Boolean logic operations such as addition, subtraction and sequence comparison can be performed in the feedforward path. We have built a feedback loop to guide the ALU to implement the error detection. Using TSMC 180nm CMOS technology, the post layout simulation shows that the power dissipation of the proposed ALU varies from 0.0394W to 0.207W when the frequency varies from 0.5GHz to 2GHz. The computation delay in this design is less than 10ns under 10fF load. Compared to the same ALU built in static logic, the proposed one using multiple input floating gate logic has the advantages of energy saving and large tolerance in fan-out. Besides, introduced feedback loop does not bring large overhead to the ALU.

**Keywords** — Multiple input floating gate (MIFG) MOSFET, neuron-like cell, arithmetic logic unit (ALU), error detection

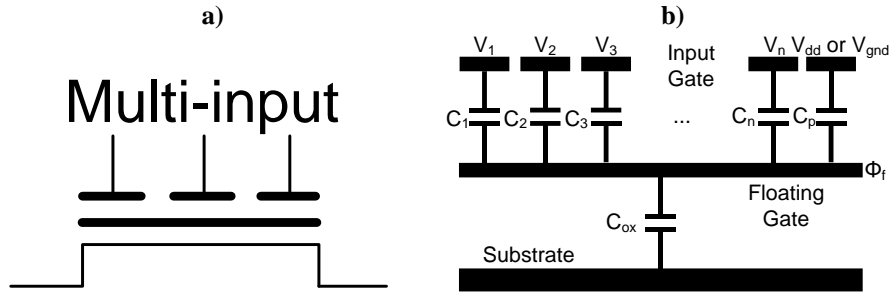
## 1 INTRODUCTION

Arithmetic logic unit (ALU) is one of the important blocks in present computer systems [1, 2]. Especially in graphics processing unit (GPU), products are embedded over 10 ALUs in order to strengthen computing ability [3, 4]. How to address the trade-off between several metrics such as power dissipation, computing latency, maximum fan-out and error ratio needs to be studied. Based on this, ALU is designed also in SOI (silicon on insulator) process to effectively mitigate static power dissipation [5]. Furthermore, the application of dual-voltage has been reported to allow some noncritical blocks or paths to operate under lower power at the cost of computation speed [6]. Back gate forward biasing method has been used in ALU design [7, 8], which increases the speed by reducing transistors' threshold voltage. The method of bit-partition is used for reducing the pipeline stages to speed up the entire computing. To suppress the possible occurrence of error during operation, adaptive clock has been utilized to avoid setup and hold violations [9]. Besides, the ALU design using signal latency in internal computing flow as the detective signal for error calibration has also been proposed [10].

Several factors such as setup and hold violations, imperfect logic transfer and possibly asynchronous signal flow during frequency adjustment, are likely to introduce computing error in a digital block. Thus, the digital calibration techniques are proposed to suppress the possible error. The first mainstream method is proposed in the work of Whatmough et al. [11] in which several checkpoints are set in critical paths to recalculate the result and compare with the main paths. If an error is detected, additional clock cycles are required to wait for both logic reset and newly correct result evaluated. Due to only critical paths which embeds the checkpoints, other error occurred in other logic chains cannot be calibrated. Another method is based on adaptive clocking as described by Chae and Mukhopadhyay [12, 13]. This method uses the information of delay during logic switch to predict the potential error according to the scheduled switching timing. Once unwanted switching is detected, the clock gating will power down the corresponding path and wait for the recycled computing with the scaling of the global frequency. Similar to the method described in [11], the adaptive clock is also generally set in critical paths and so it cannot cover all possible errors in a digital block.

The transistor with multiple input floating gate (MIFG) is proposed to make logic gates much more programmable than the single gate input transistors [14-16]. Using the property of the MIFG transistors, the designs of neuron-like cells have been reported [17-21], in which the state of a transistor is decided by the multiple inputs so that the diversity in design can be improved significantly and the single block with multiple applications can be achieved.

In this work, we propose a novel ALU using MIFG logic. The proposed ALU has both feedforward path and feedback



**Figure 1. MIFG MOSFET: a) Device symbol. b) Equivalent circuit model.**

feedforward path, we use MIFG logic to build neuron-like cells to execute multi-function tasks. In the feedback loop, a low cost block used for the error detection is designed, which assists the calibration block in the feedforward path to avoid any unwanted error.

The main contributions of this work are:

i. Using MIFG MOSFETs, we have designed several neuron-like cells to behave like elementary logic. Some cells can be shared to different operations.

ii. In the design of the proposed ALU, we have introduced how the feedforward path and feedback loop work together to achieve calibration.

iii. The post-layout simulations using TSMC 180nm CMOS show that the power dissipation of the proposed ALU is less than that of the same ALU built by the static logic and our design has a strong driven ability. Introduced feedback loop does not bring large overhead to the normal feedforward path operating all functions.

Section 2 proposes a series of neuron-like cells used as logic gates. Section 3 describes the detailed circuit design of a 32-bit ALU. The post-layout simulation results are shown and discussed in Section 4. Finally, Section 5 is the conclusion of this work.

## 2 NEURAL-LIKE CELLS USING MIFG MOSFETS

Figure 1.a shows the device symbol of a MIFG MOSFET, which is also called as the neuron MOSFET. Figure 1.b shows the equivalent circuit model. The MIFG MOSFET has two gate layers [22]. The bottom one is the floating gate, which is over the transistor channel, and directly controls the conduction of carriers in the channel. The top gate layer is the input gate over the oxide. To obtain multiple inputs, the top layer is deposited as individual gates during fabrication so that each of them can serve as an input port of the device. The oxide layer isolating a floating gate and several input gates can be seen as an array of coupling capacitors to control the surface potential of the channel. For a single MIFG MOSFET, its

surface potential,  $\Phi_f$ , can be expressed as follows [23]:

$$\phi_f = \frac{C_1 V_1 + C_2 V_2 + \dots + C_n V_n}{C_1 + C_1 + \dots + C_n + C_{ox} + C_p} = \frac{C_1}{C_{total}} V_1 + \frac{C_2}{C_{total}} V_2 + \dots + \frac{C_n}{C_{total}} V_n \quad (1)$$

where  $V_1, V_2, \dots, V_n$  are the input voltages through each top gate.  $C_1, C_2, \dots, C_n$  represent input coupling capacitors contributed by the oxide layer, which isolates the floating gate and each input gate.  $C_{ox}$  is the coupling capacitor between the floating gate and the device substrate.  $C_p$  is the parasitic capacitor existing between the floating gate and the substrate. Setting the input coupling capacitors as large value and the dimension of transistors as small value, we can suppress the non-linearity contributed by  $C_p$  and  $C_{ox}$ , respectively.

For a binary decision neuron (BDN), its output can be expressed as follows:

$$Y = f \left( \sum_{i=1}^n W_i X_i - S \right) \quad (2)$$

where  $X_i$  is input signal and is weighted by  $W_i$ .  $S$  is the threshold value of BDN.  $f(x)$  is the activation function, which outputs the final result. If we map  $W_i$  and  $X_i$  in Equation (2) to  $C_i/C_{total}$  and  $V_i$  in Equation (1), respectively, and  $S$  in Equation (2) to the threshold voltage in a MIFG MOSFET or a transfer point in a logic transfer curve, logic gates with floating gate MOSFETs can be built similar to a spike neuron in which the result is evaluated by multiple inputs.

For each operation, normal ALUs have independent blocks. In this work, we share the same neuron-like cell using MIFG logic to group some specific operations for saving the area.

For both the inverter and buffer, we use the same cell as shown in Figure 2.a. The first stage is an inverter realized from two floating gates and the second one is the normal static inverter. If we ignore both  $C_{ox}$  and  $C_p$ , and extend the analysis of Equation (1), the surface potential in the first stage of Figure 2.a can be expressed as follows:

$$\phi_f = \frac{C_{in} V_{in} + C_{via\_th} V_{via\_th}}{C_{in} + C_{via\_th}} = \frac{C_{in}}{C_{total}} V_{in} + \frac{C_{via\_th}}{C_{total}} V_{via\_th} \quad (3)$$

where  $V_{via\_th}$  is the variable and controlled by the instruction code of the proposed ALU, and  $C_{total}$  is the sum of  $C_{in}$  and  $C_{via\_th}$ . If  $C_{in}$  is equal to  $C_{via\_th}$ , the output after the first stage can be described as follows:

$$V_{out} = f_{INV} \left( \frac{1}{2} V_{in} + \frac{1}{2} V_{via\_th} - V_{th} \right) \quad (4)$$

where  $f_{INV}(x)$  is the inverter transfer function and  $V_{th}$  is the threshold voltage of the inverter. Through the adjustment of

$V_{via\_th}$ , we can vary the threshold voltage of the inverter if  $V_{in}$  is seen as the only input. Since discharging is always faster than the charging, to ensure signal integrity,  $V_{via\_th}$  is set to 0V thus connecting to the ground. Compared to the MIFG inverter with the adjustable threshold voltage reported in [24], the advantage of our design is that it has less capacitors involved to save area and the variable to adjust the threshold voltage is another input signal but not the ratio of two capacitors. Then after the second stage, the buffer's output can be obtained. Thus, the proposed two-stage cell performs the functions of both inverter and buffer.

We map the design principle of both inverter and buffer for the design of a two-input gate performing NAND and NOR using MIFG logic where two data inputs plus a variable input are applied in the first stage. If three coupling capacitors are the same, the output after the first stage can be expressed as:

$$V_{out} = f_{INV} \left( \frac{1}{3}V_a + \frac{1}{3}V_b + \frac{1}{3}V_{via\_th} - V_{th} \right) \quad (5)$$

where  $V_a$  and  $V_b$  are two inputs. If  $V_{via\_th}$  is connected to the ground, Equation (5) can be expressed as follows:

$$V_{out} = f_{INV} \left( \frac{1}{3}V_a + \frac{1}{3}V_b - V_{th} \right) \quad (6)$$

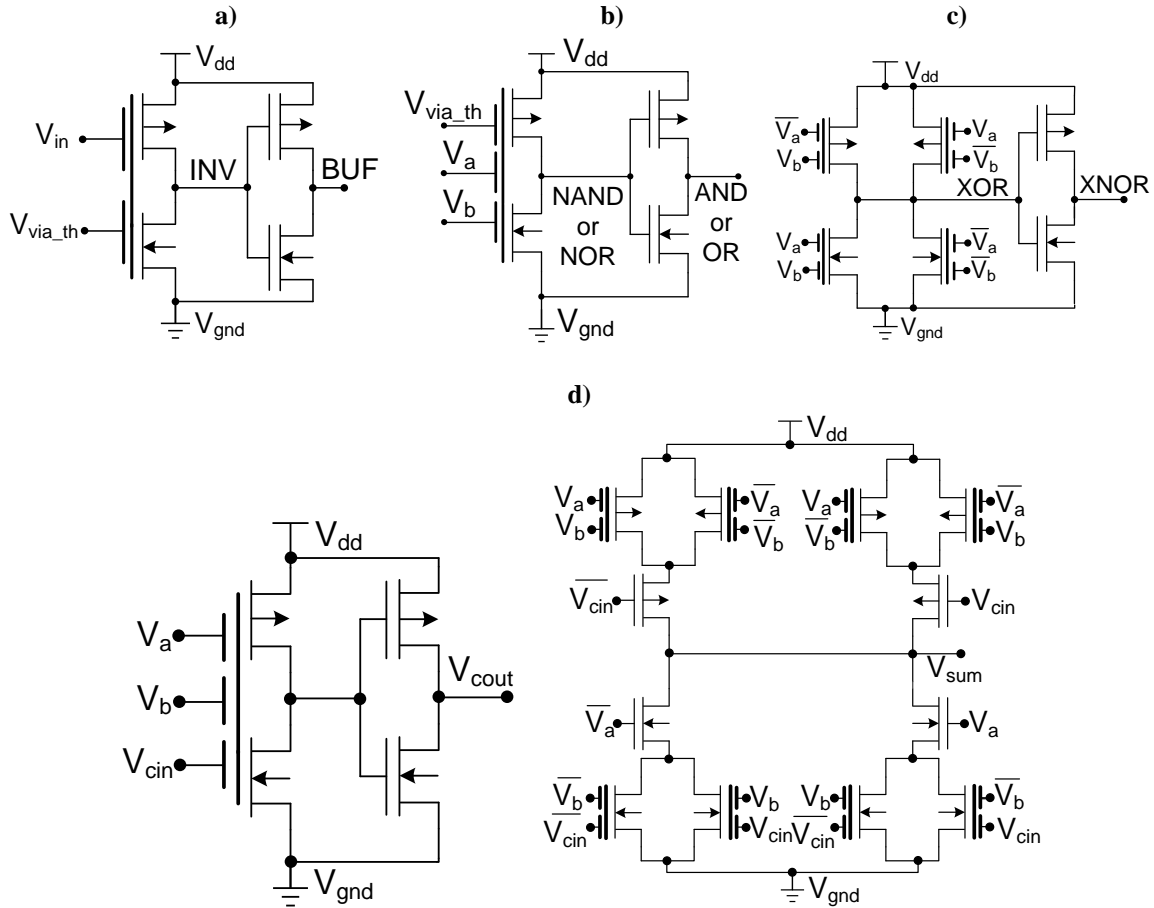
According to above equation, when both inputs are high, the output will be low. Otherwise, the surface potential does not exceed the threshold voltage so that the output stays in high, which matches the NAND operation.

If  $V_{via\_th}$  is connected to supply voltage, Equation (5) can be expressed as follows:

$$V_{out} = f_{INV} \left( \frac{1}{3}V_a + \frac{1}{3}V_b + \frac{1}{3}V_{dd} - V_{th} \right) \quad (7)$$

In this case, only when both inputs are low, the output will be high. Otherwise, the output is always low, which is NOR operation. Thus, as shown in Figure 2.b, we propose a two-stage circuit using MIFG logic to implement the following four Boolean logic functions: NAND, AND, NOR, and OR.

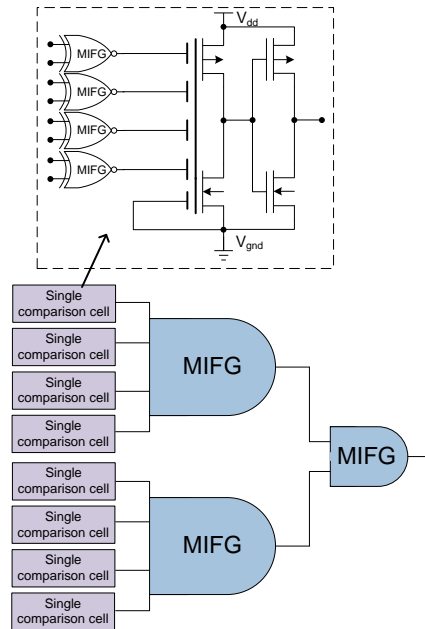
For XNOR and XOR, it is obvious that the floating gate potential diagrams (FPDs) of both XNOR and XOR are not monotonically increasing functions [21]. Therefore, we cannot use an additional input to adjust the surface potential to program the desired function. One solution is to use a cascade structure with NAND, NOR and inverter gates to obtain both XNOR and XOR at the cost of delay and area. We propose a pass transistor based multiple input floating gate (PT-MIFG) logic to achieve XOR and XNOR as shown in Figure 2.c. The first stage is in complementary topology, which uses



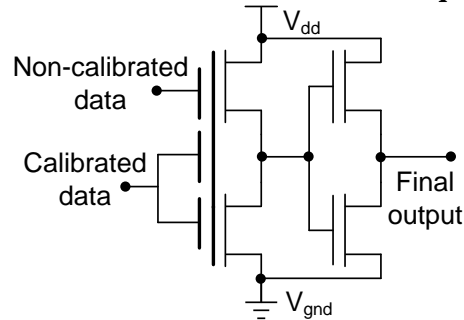
**Figure 2. Circuit diagram of the proposed MIFG logic gates: a) INV and BUF. b) NAND, AND, NOR and OR. c) XOR and XNOR. d) Full adder including carry-out and sum blocks.**

four transistors with eight same coupling capacitors to execute XOR. The main principle of this circuit is that four pass transistors with MIFGs reflect four input cases to obtain the desired output of an XOR gate. For an n-type pass transistor with MIFGs, only if two inputs are high, the transistor will be switched-on. Likewise, only if two inputs of a p-type pass transistor with MIFGs are low, the transistor will be switched-on. XNOR is also obtained by connecting to a static inverter.

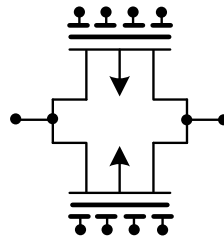
For addition and subtraction, the full adder using MIFG MOSFETs includes two circuit blocks to output sum bit and carry-out bit as shown in Figure 2.d. The full adder has three inputs, which are two adding bits and a carry-in bit. For the block outputting the carry-out bit, it can be concluded that when more than one inputs are high, the output will be high. Thus, we can use three-input MIFG inverter to implement this function. For the design of sum block, it is the XOR function with all three inputs. We combine PT-MIFG logic and traditional pass transistor logic together to achieve the function. The circuit has four branches representing eight input cases. For a single branch, it includes a pair of paralleled MIFG MOSFETs and a CMOS transistor corresponding to two input cases, each input case can be enabled to output the



**Figure 3. Proposed circuit block used for 32-bit sequence comparison.**



**Figure 4. Proposed circuit block of digital calibration.**



**Figure 5. PT-MIFG logic based TG with four coupling capacitors per transistor to decode 4-bit instruction set.**

sum bit corresponding to the unique input case.

Above designs are regarding eight standard Boolean operations and two arithmetic functions. We will introduce 32-bit sequence comparison, single-stage multiplexer (MUX) and digital calibration to the following ALU design. For digital comparison, its function is to judge if two 32-bit sequence is the same. We use sequence partitioning to first compare each 4-bit sequence. The 4-bit digital comparator is shown in Figure 3. An XNOR gate compares two 4-bit signals from two vectors. The outputs of XNOR array are connected to a 4-input AND gate using MIFG logic. To obtain a correct output,



**Table I**  
**OPERATION SUMMARY**

<i>Operation</i>	<i>Instruction code</i>	<i>Operation</i>	<i>Instruction code</i>
<i>INV</i>	0000	<i>Left rotation</i>	1000
<i>BUF</i>	0001	<i>Right rotation</i>	1001
<i>NAND</i>	0010	<i>Left shift</i>	1010
<i>AND</i>	0011	<i>Right shift</i>	1011
<i>NOR</i>	0100	<i>Addition</i>	1100
<i>OR</i>	0101	<i>Subtraction</i>	1101
<i>XOR</i>	0110	<i>Sequence comparison</i>	1110
<i>XNOR</i>	0111	<i>Reset</i>	1111

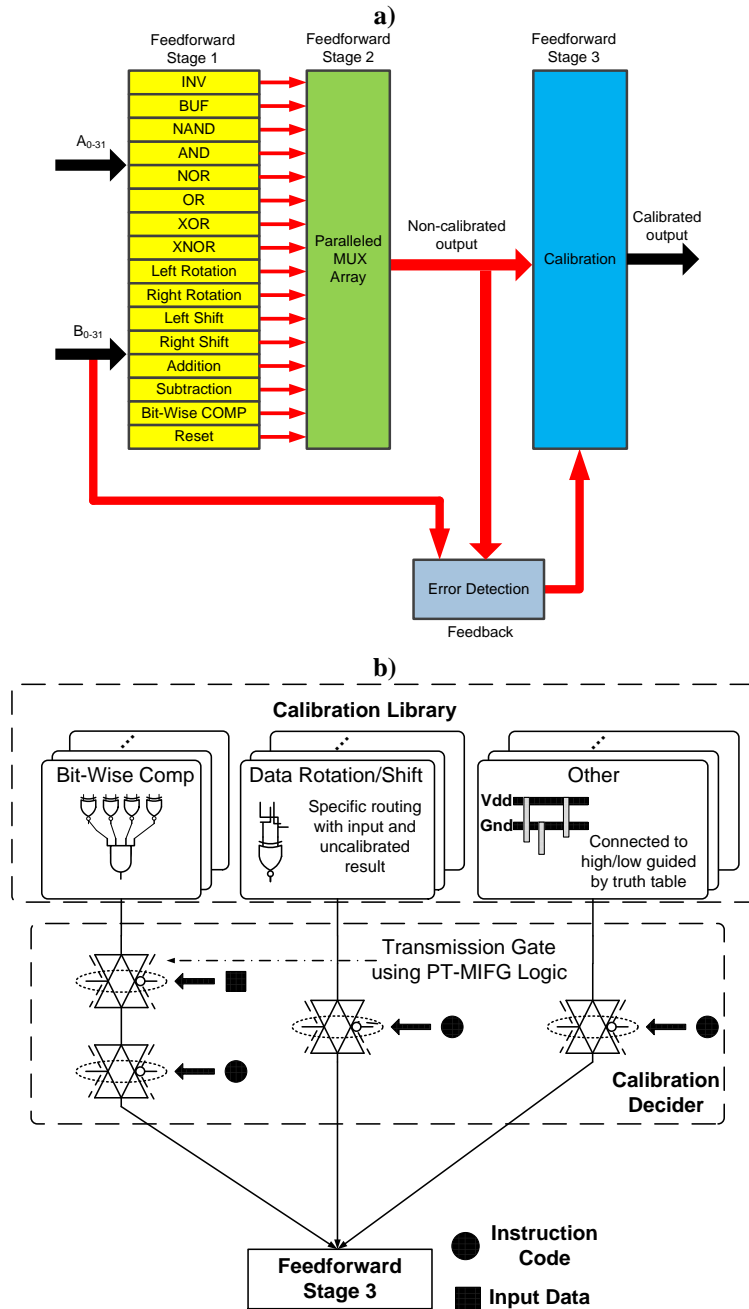
the ratio of five coupling capacitance in the 4-input AND gate of Figure 3 is set to 1:1:1:1:3. After comparing each 4-bit sequence, the result of 32-bit sequence is determined by MIFG AND gate chain including 2-input MIFG AND gate and 4-input MIFG AND gate as also shown in Figure 3. The final output is decided by both the non-calibrated data and the correct data. Figure 4 shows the implementation of a single calibration block. It is similar to the circuit shown in Figure 2.b. Data without calibration connects a single coupling capacitor. Data with calibration connects the rest of two coupling capacitors. Thus, the weight of data with calibration is larger than that of data without calibration. In this configuration of coupling capacitors, the data with calibration will replace the non-calibrated data. This stage not only can output correct data but can improve also the signal integrity of the proposed ALU's output.

MUX is necessary in ALU and can be designed by the transmission gate (TG) array or logic gates. In this work, we use PT-MIFG logic to design TG array. The proposed TG is shown in Figure 5. Each node on floating gates can be connected with an inverter to change the enable logic for ALU decoding. Since ALU design has 16 operations so that normally there are four CMOS TGs required for decoding each path. Using the proposed design, the only one TG implements decoding in each path to reduce process latency.

### 3 ALU DESIGN

#### 3.1 System overview

Figure 6.a shows the block diagram of the proposed ALU. In feedforward path, 32-bit input is processed in parallel operation blocks in feedforward stage 1. After this stage, the MUX array selects the operation in feedforward stage 2. The output from the feedforward stage 2 is then calibrated by the feedforward stage 3 to obtain final result. The unwanted error is detected through the feedback loop with the aid of both input data and instruction code. Table 1 summarizes operations



**Figure 6. Proposed ALU: a) Block diagram. b) Feedback loop for error detection.**

with the corresponding instruction codes.

In feedforward stage 1, neuron-like cells shown in Figures 2 and 3 are used. Note that each cell for operating Boolean logic in this work performs at least one logic. MUX is implemented by PT-MIFG logic based TG as shown in Figure 5. As stated earlier, the proposed TG can decode faster than the conventional one with four cascading TGs. The calibration cell shown in Figure 4 builds the feedforward stage 3 to achieve error recovery with the support of the feedback loop.

### 3.2 *Feedback loop*

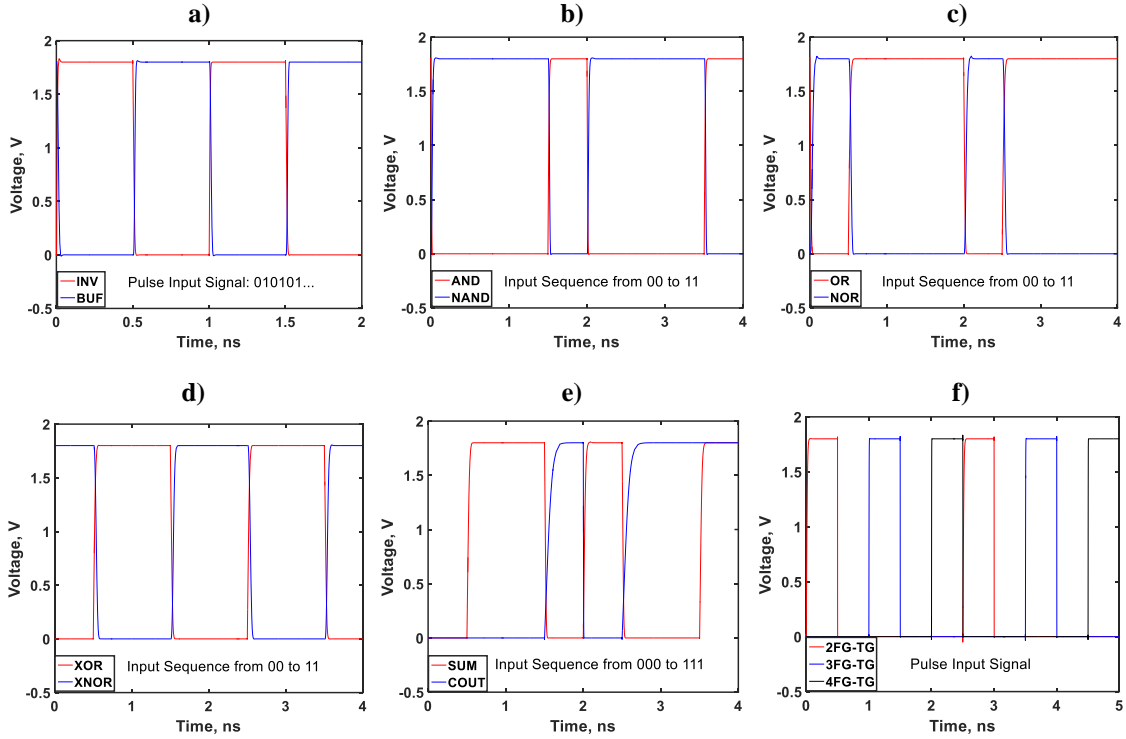
The feedback loop in the proposed design works for error detection. The current error detection mainly uses lookup table (LUT), delay-aware counter and digital comparison in checkpoint [25-27]. The described methods can reliably calibrate error at the overhead of the chip area, computation latency and power dissipation.

Figure 6.b shows the block diagram of the feedback loop, which implements the error detection. This stage has two parts including calibration library and decider as follows. For eight Boolean logic operations, two arithmetic operations and reset operation, we use truth table to guide the calibration. The correct data is obtained through the connection to either the power supply or ground. The calibration decider used for these operations is built by two TGs using PT-MIFG logic. The first TG is controlled by the input data to enable desired result going to the next TG. The second TG is controlled by the instruction code directly to judge which operation should be activated. Taking AND operation with the input of 01 as an example, the input of the first PT-MIFG logic based TG in the calibration decider connects to ground as the correct result, and its controlled nodes are enabled by 01 to let the data go through. The controlled nodes in the second TG in the calibration decider are connected to the instruction code, 0011. Thus, under this principle, the correct output, low logic, can be uniquely transferred to the feedforward stage 3 for calibration. For the operations of shift and rotation, the calibration library can be implemented by the XNOR array. Each XNOR gate in this array is an input by the non-calibrated data and an original input data used as the correct output after shift or rotation with the specific direction. If the output of XNOR is logic high, it means that the result without calibration is correct. Otherwise, it means that the error happens during shift or rotation. The output of the XNOR will control a normal TG with the input of the correct result routed from the original input. Once this normal TG is switched-on, the data will go to the calibration decider, which is a TG with floating gates controlled by the instruction code. As an example, considering the left rotation operation with the input of all logic low, if a logic high obtained before calibration, the XNOR will be logic low in the calibration library and thus the correct result will finally go to the feedforward stage 3 through the calibration decider enabled by the instruction code, 1000. For sequence comparison, introduced XOR and AND chain used as the calibration library is to achieve bit-wise comparison to obtain the correct result. It is connected to the calibration decider, a PT-MIFG logic based TG controlled by the instruction code to judge if the calibrated data go through or not. Essentially, the calibration of this operation is to recalculate the result, which may go to the feedforward stage 3 if this operation is enabled by the instruction code.

## 4 RESULT AND DISCUSSION

### 4.1 Logic cell

Figure 7 shows the transient simulations of all proposed cells. We observe that all basic functions can be correctly implemented using MIFG logic. Note that for the proposed MIFG logic gates, all transistors in them require proper aspect ratio,  $W/L$  in order to obtain correct surface potential evaluating the desired output.  $W$  and  $L$  are channel width and length of transistors. Channel lengths of all transistors in the proposed MIFG logic are uniformly set as  $3\lambda$ , where  $\lambda$  is 90nm which is the unit length of layout design restricted by TSMC 180nm CMOS.  $W_p$  and  $W_n$  are the widths of p-type transistors and n-type transistors in any logic gate, respectively. For INV/BUF,  $W_p/W_n$  of the first stage with floating gates is set to  $9\lambda/3\lambda$ . For NAND/AND/NOR/OR,  $W_p/W_n$  of the first stage with floating gates is set to  $12\lambda/3\lambda$ . For XOR/XNOR,  $W_p/W_n$  of the first stage with floating gates is set to  $13.5\lambda/3\lambda$ . For the carry-out block in the full adder,  $W_p/W_n$  of the first stage with floating gates is set to  $6\lambda/3\lambda$ . For the sum block in the full adder,  $W_p/W_n$  of the first stage with floating gates is set to  $16.5\lambda/3\lambda$ . For above logic gates,  $W_p/W_n$  of the second stage (a static inverter) is set to  $9\lambda/3\lambda$ . For a TG with coupling capacitors,  $W_p/W_n$  of the two transistors is set to  $12\lambda/6\lambda$ . With described transistor dimension, the noise margins of all cells have been set to avoid logic errors due to multiple states on the channels of transistors. For static logic provided by TSMC 180nm standard library, channel lengths of all transistors are  $2\lambda$ .  $W_p/W_n$  of all logic gates are  $10\lambda/6\lambda$ . Table 2 lists the comparison of static logic and the proposed logic cells using MIFGs. We can see that for simple Boolean logic, the delay of MIFG logic is slightly larger than that of static logic. However, MIFG logic consumes less power than the static logic since the multiple coupling can mitigate the power consumed by the dynamic current during logic switching, and coupling capacitors does not consume power. For complex logic including XOR/XNOR, and unit full adder, PT-MIFG logic are used without cascade topology. Thus, these blocks work faster than the gates built by static logic. For the transmission gate designed in PT-MIFG logic, the performance of the proposed one is slightly improved from the view of both power dissipation and delay. The static power dissipation due to leakage current in transistors is also listed as the data within brackets in Table 2. Simulation is done in TSMC 180nm technology with 1.8V supply and 0.1fF load. All results of static logic gates use TSMC 180nm standard library. The power dissipation contributed by the leakage current in static logic ranges from 0.64% to 2.13% of the total power dissipation. The power dissipation contributed by the leakage current in MIFG logic varies from 3.7% to 13.3% of the total power dissipation. Thus, the static power dissipation in MIFG logic is more dominant than that in static logic due to large leakage current in MIFG logic with various surface potentials. From the point of view of area, the ALU in MIFG logic consumes relatively large area than in static logic due



**Figure 7. Transient simulations of the MIFG logic based cells: a) INV and BUF. b) NAND and AND. c) NOR and OR. d) XOR and XNOR. e) Full adder. f) Transmission gate with 2, 3 and 4 coupling capacitors per transistor.**

to coupling capacitors. In essence, the proposed cells using MIFGs are more energy-efficient and behave better in complex logic considering both the speed and power dissipation at the cost of static power dissipation and area than the traditional static logic gates.

## 4.2 Entire system

Figure 8 shows layout of the proposed 32-bit ALU in TSMC 180nm CMOS. The chip area is  $7.5\text{mm}^2$  with 69 input ports, 33 output ports, a voltage supply port, and a ground port. Feedforward path and feedback loop occupy  $3.3\text{mm}^2$  and  $2.2\text{mm}^2$ , respectively. Four metal layers are used for interconnections. The most part of the chip is occupied by the coupling capacitors to implement MIFGs.

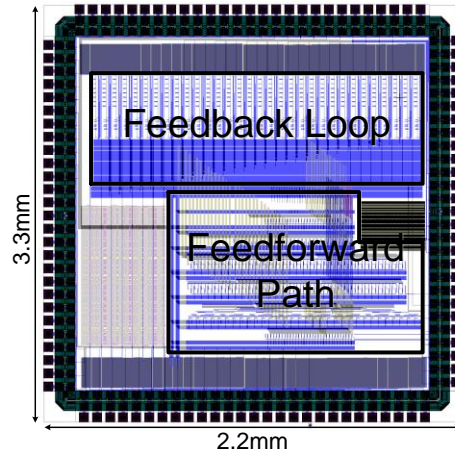
We first focus on the proposed calibration method. Figure 9 shows the result in which logic low should be the correct result and logic high should be the incorrect pulse. We used three operations, NAND, left rotation and digital sequence comparison, corresponding to three methods in calibration library of Figure 6.b in this simulation. We observe that unwanted spike can be well detected and calibrated by the proposed method. The calibrated times are 0.072ns, 0.067ns and 0.108ns corresponding to NAND, left rotation and digital sequence comparison, respectively.

**Table II**  
**COMPARISON OF STATIC LOGIC AND PROPOSED NEURON-LIKE CELLS USING MIFG LOGIC**

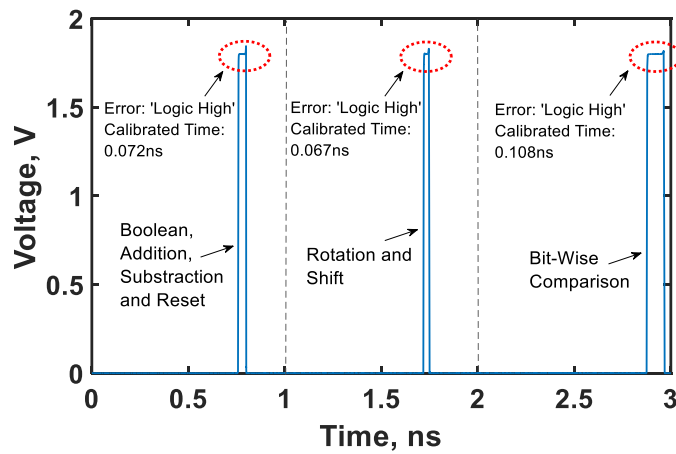
<i>Performance</i>	<i>Delay (ps)</i>		<i>Power Dissipation@1GHz (nW)*</i>		<i>Area (<math>\mu\text{m}^2</math>)</i>	
	<i>Static Logic</i>	<i>MIFG Logic</i>	<i>Static Logic</i>	<i>MIFG Logic</i>	<i>Static Logic</i>	<i>MIFG Logic</i>
<i>INV</i>	15	21	2.002 (0.0161)	0.944 (0.0634)	7.9	105.5
<i>BUFFER</i>	63	64	3.896 (0.0335)	1.899 (0.0702)	11.8	105.5
<i>NAND</i>	23	28	3.881 (0.0283)	1.313 (0.1359)	11.8	167.8
<i>AND</i>	65	70	4.825 (0.0452)	1.845 (0.0813)	19.7	167.8
<i>NOR</i>	47	68	4.619 (0.0297)	1.148 (0.1523)	11.8	167.8
<i>OR</i>	88	92	5.808 (0.0462)	2.044 (0.0919)	15.7	167.8
<i>XOR</i>	94	46	6.833 (0.0738)	2.092 (0.2402)	27.6	1339.5
<i>XNOR</i>	124	99	7.487 (0.0681)	2.977 (0.1821)	23.6	1339.5
<i>FA</i>	658	216	11.393 (0.2425)	4.809 (0.4253)	339.1	2919.9
<i>TG</i>	37	36	0.723 (0.0092)	0.698 (0.0792)	13.6	537.4

\*Data without brackets is the total power dissipation.

The second part of simulations focuses on power issue. Figure 10.a exhibits the power dissipation of each operation and its average value of the proposed ALU built by both MIFG logic and CMOS static logic for operation at 1GHz with random input data flow. For the static logic based simulation, we use the same schematic design, in which all MIFG logic are replaced by static logic. The proposed ALU built by MIFG logic dissipates less energy than that designed in static logic, especially for addition, subtraction and sequence comparison. This reduction in power dissipation is due to the short circuit current suppressed by the multiple coupling capacitances of MIFGs in the design. Figure 10.a also shows the penalty in power dissipation when the error occurs. It can be seen that when complex operations including addition, subtraction and sequence comparison have the error, the power dissipation increases the most since the blocks of error detection used for these three operations are more complex than that used for the rest of operations. Averaging all operations, the power growth rate due to calibration in static logic and MIFG logic is 23.2% and 19.6%, respectively. Figure 10.b exhibits the power dissipation with the variation of frequency of four types of ALUs. It shows the overhead of power dissipation due to the proposed feedback loop and the power comparison of the proposed ALU designed in MIFG

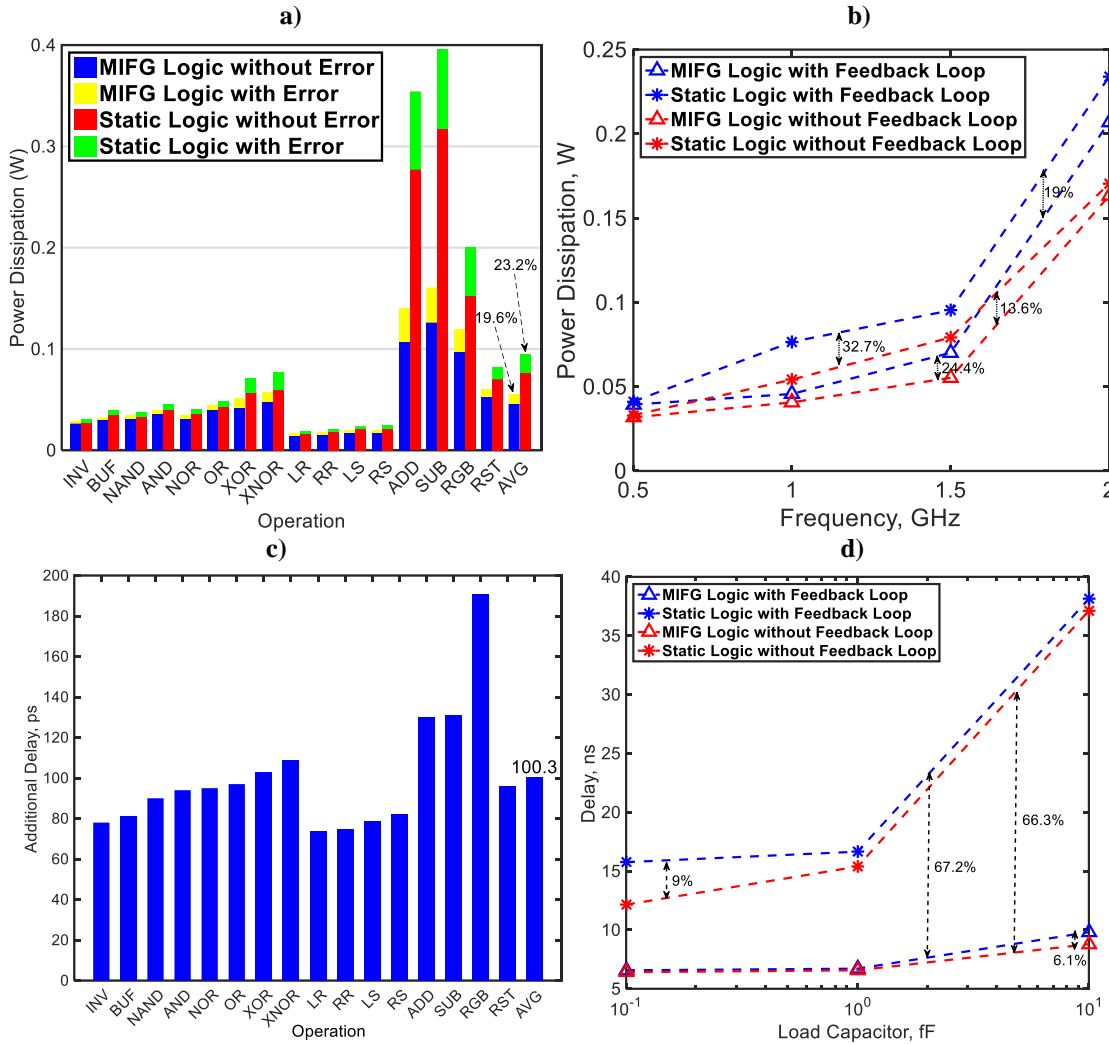


**Figure 8. Full layout view of the proposed ALU.**



**Figure 9. Transient simulation of error (logic high) corresponding to three calibration methods shown in the calibration library of Figure 6.b.**

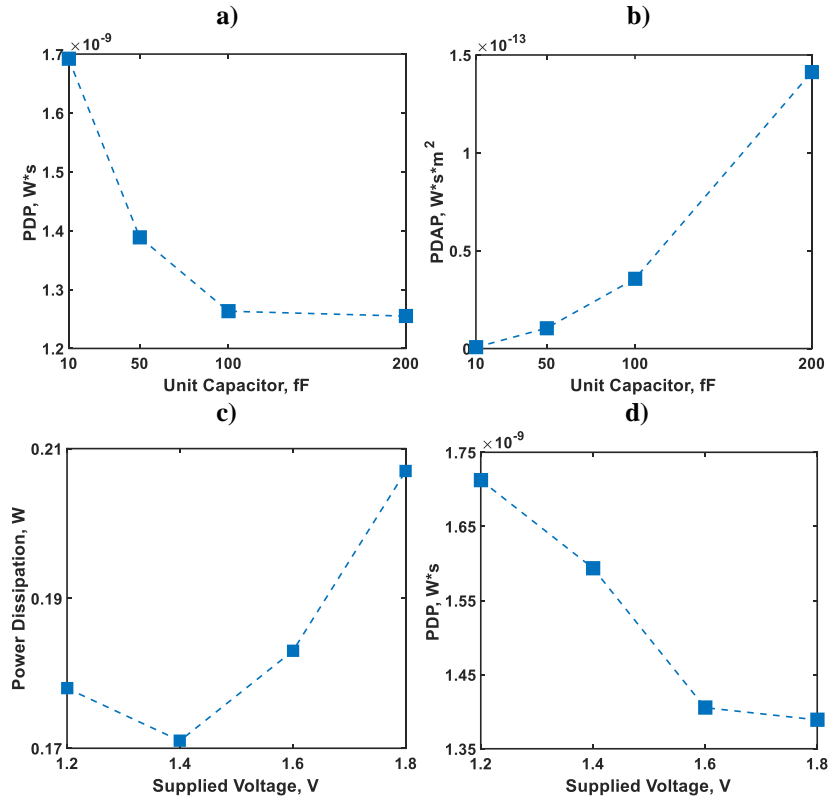
logic and static logic. The power dissipations in the proposed ALU built in MIFG logic are 0.0394W and 0.207W at 0.5GHz and 2GHz frequency, respectively. Compared to the proposed ALU designed in traditional static logic, the average reduced rate of power dissipation corresponding to MIFG logic is 19%. If the proposed feedback loop is cancelled, the average reduced rate of power dissipation due to different logic types is 13.6%. Considering the same logic type, the average reduced rate of power dissipation due to the cancellation of the feedback loop is 24.4% and 32.7% corresponding to MIFG logic and static logic, respectively. Thus, the ALU designed by MIFG logic is much more energy efficient than that by static logic and the proposed feedback loop does not bring large overhead to power dissipation. Figure 10.c shows the overhead of computation delay due to the feedback loop. For most operations, additional delay due to error detection and calibration is below 100ps. The error contributes the largest delay for the operations of addition, subtraction and sequence comparison. Especially in sequence comparison, the additional delay almost reaches to 200ps. Since the detection block used for this operation requires logic chain as described in the last section, the average delay is 100.3ps. We also looked into the relationship between load capacitor and computation delay. In most cases, an ALU is not



**Figure 10. Simulation results: a) Power dissipation of each operation and average value. b) Power dissipation dependence on frequency. c) Delay due to feedback loop. d) Delay dependence on load capacitor.**

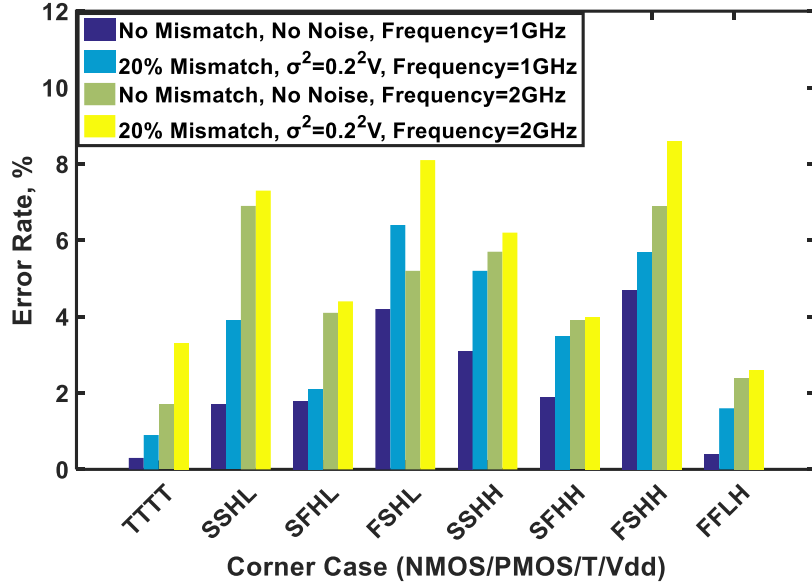
fabricated into a single chip but connects to other large blocks to form a bigger chip, such as the CPU and GPU. Therefore, the drive-in capability of the output port in the ALU is important. We have simulated the computation delay with the variation of load capacitor of four types of ALUs as shown in Figure 10.d. The most impressive feature from Figure 10.d is that the MIFG logic based ALU is insensitive to an external load. The computation delay in the proposed ALU built in MIFG logic is less than 10ns when the output is loaded by the 10fF capacitor. Compared to the proposed ALU designed by traditional static logic, the average reduced rate of delay corresponding to MIFG logic is 67.2%. If the proposed feedback loop is cancelled, this average reduced rate of delay due to different logic types is 66.3%. Considering the same logic type, the average reduced rate of latency due to the cancellation of the feedback loop is 6.1% and 9% corresponding to MIFG logic and static logic, respectively. In essence, the proposed design could work well if the ALU is a sub-module embedded in a VLSI chip with a large fan-out. The computation delay is not sensitive to the introduced feedback loop,





**Figure 11. Scalability study: a) PDP depending on applied unit capacitor as floating gates. b) PDAP depending on applied unit capacitor as floating gates. c) Power dissipation depending on the reduction of supplied power. d) PDP depending on the reduction of supplied power. All simulations are performed at 1GHz.**

We also implement the scalability study considering area, power and delay for our design. Two metrics are used in this study. The first metric is power-delay product (PDP) and the second one is power-delay-area product (PDAP). The unit capacitor used in our ALU design is 50fF. Figure 11.a and Figure 11.b show the performance with the variation of unit capacitor from 10fF to 200fF. We see that larger unit capacitor used in our design will decrease PDP. PDPs corresponding to 10fF and 200fF used as the unit capacitor are  $1.69 \times 10^{-9} W*s$  and  $1.26 \times 10^{-9} W*s$ , respectively. The increased in PDP can be explained by the fact that strong coupling due to floating gates can reduce the glitch current during logic transfer. Thus, both delay and power dissipation due to the short current existed in logic transfer are reduced. However, the PDP variation from 100fF to 200fF in Figure 11.a shows that the benefit from larger floating gate is not as large as the variation from 10fF to 100fF. Besides, from Figure 11.b, it is concluded that larger floating gates used in our design does increase the overhead of area. For the case of 200fF used as the unit capacitor, PDAP can reach to  $1.45 \times 10^{-13} W*s*m^2$ . Since the ALU design in our work is a digital circuit, it can be operated under lower supplied voltage at the cost of larger delay. Figure 11.c and Figure 11.d show the simulation results of the ALU performance with voltage scaling. We notice that



**Figure 12. Sensitivity Study: error rate with PVT variation and introduced noise.**

power supply is 1.4V, power dissipation obtains the minimum value, 0.172W. If the proposed ALU under lower power supplied, more imperfect charge and discharge cannot meet the noise margins of logic gates. Thus, both short current and power dissipation will increase corresponding to 1.2V used as the supplied voltage. For PDP, the overhead of computing delay due to the reduced power supply is always dominant compared to the overhead of power dissipation due to the increased power supply voltage. Thus, considering both power and delay, 1.8V power supply is the best choice for the proposed design. The PDPs are  $1.72 \times 10^{-9}W*s$  and  $1.38 \times 10^{-9}W*s$  corresponding to the ALU powered by 1.2V and 1.8V, respectively. Besides, reduced supplied power leads to more errors occurring at high frequency.

Even through the proposed design uses all digital blocks, we have used floating gates also. Therefore, the sensitivity study is necessary. Our sensitivity study will consider the variations of process, voltage and temperature (PVT) and input noise. For process issue, we focus on both the corner of transistor and device mismatch. To evaluate the performance of these variations, we use error rate as the metric. For device mismatch, we use Monte Carlo method. The mismatch values of all devices including transistors and capacitors used in floating gates are set within  $\pm 20\%$  distribution. The corner cases include TTTT, SSHL, SFHL, FSHL, SSHH, SFHH, FSHH and FFLH. Four letters from left to right in these cases represent the variations of n-type MOSFET, p-type MOSFET, temperature and voltage supply, respectively. S and F located in the first two letters are slow and fast MOSFETs, respectively. T located in the first two letter represents typical MOSFETs provided by the foundry. We use  $3\sigma$  method to calculate the variable threshold voltage to define S and F for MOSFETs [28, 29]. H and L appeared in the third letter represent 340K and 270K, respectively. H and L appeared in the last letter represent 1.9V and 1.7V, respectively. T located in the last two letter represents default temperature in SPICE

simulation, 300K, or standard supply power, 1.8V. For input noise, it follows Gaussian distribution which is represented by  $V_{in\_noise}=V_{in\_ideal}\times(1+N(0,\sigma^2))$ . We set  $\sigma^2$  as  $0.2^2V$  and  $V_{in\_ideal}$  as 0V or 1.8V corresponding to perfect logic low or high. Based on above configuration, Figure 12 shows the simulation results. We see that error rates in the corners of FFLH, SFHL and SFHH are lower than other corners. This fits the theory that the pair of the fast-PMOS and slow-NMOS or the pair of fast-PMOS and fast-NMOS is the best corner for digital operation. Under 1GHz operation, for the performance influence due to both input noise and device mismatch, we observe that error rates in the corners of SSSL, FSHL, SSHH are increased mostly than other corners. Under 2GHz operation, the error rates in the corners of FSHL and FSHH are increased mostly than other corners considering input noise and device mismatch. With the increased in frequency, the error rate in the corner of SSSL is increased mostly compared to other corners. The largest error rate is 8.6% corresponding to the corner of FSHH with both input noise and device mismatch.

Comparing to other work of MIFG MOSFET based ALU [30-38], in our work, both the number of functions and bit length of our work are more than that in previous work. From the point of view of structure, the proposed design has introduced a feedback loop which can calibrate an unwanted error. Besides, the design of logic cells in our work is simple in comparison to previous work. In [30], the design of full adder still used the cascade logic chain which is penalty to both latency and area. In [31], the grouped cell using MIFGs can achieve multi-function and so cancel the stage of MUX array. However, this design strategy is not suitable when the number of required functions is increasing since FPDs of all functions cannot be always the same. Work in [32, 33] proposed two analog-neuron-cells using floating gates working for signal amplification and neural network, respectively. These two floating gate based designs have very small signal swing connecting to floating gate so that the surface potential is more sensitive to external variations of coupling capacitors and input noise than our digital design. In [34], floating gates are applied to digital I/O block to increase the work efficiency of data transmission at the overhead of area which is the same as our design. Designs in [35-37] use floating gates to implement programmable array. The essence of these three work is to use the array of floating gate based logic gates to accelerate computing. Thus, compared to our design, the reported work requires additional decoder to select which operation should be enabled like the work flow in normal FPGAs. The design in [38] is novel since it used transistors with floating gates operating in weak inversion to build an integrator. Thus, this design of the integrator saves much power than that other normal integrators in which all transistors operate in saturation. However, due to weak inversion is applied to all transistors with floating gates, both input and output swings are smaller than normal designs.

## 5 CONCLUSION

We have proposed a 32-bit ALU. The essence of this design is to use MIFG logic to build neuron-like cells as the logic gates and use a feedback loop to detect error. The post-layout simulation shows that the proposed design is energy-efficient, fast-computing and insensitive to load capacitor compared to the one built in traditional static logic. Implementation of floating gates with high capacitance per unit area is required to save the block area though the overhead of area due to applied floating gates is large. Besides, the overhead of power dissipation due to our calibration feedback loop is large. However, the circuit optimization of this block to reduce the power dissipation is necessary. Design can be improved to compensate for PVT variation effects for implementation in advanced CMOS processes and is suggested for future work.

## ACKNOWLEDGMENT

Part of the work is supported under NSF grant No. 1422408.

## REFERENCES

- [1] Y. Kawakami, H. Tanaka, T. Nukiyama, M. Yoshida, T. Nishitani, I. Kuroda, M. Araki, T. Hoshi, and A. Nakajima, "A 32b floating point CMOS digital signal processor" Proceedings of the IEEE International Solid-State Circuits Conference, (1986), pp.86-87.
- [2] F. Cesaroni, S. Marco, E. Gennari, and S. Gentile, "A general purpose arithmetic logic unit" Nuclear Instruments and Methods in Physics Research (1987), Vol. 260, N° 2, pp.425-429.
- [3] N. Gong, J. Wang, and R. Sridhar, "Application-driven power efficient ALU design methodology for modern microprocessors" Proceedings of International Symposium on Quality Electronic Design, (2013), pp.184-188.
- [4] W. Jia, K. A. Shaw, and M. Martonosi, "Stargazer: Automated regression-based GPU design space exploration" Proceedings of the IEEE International Symposium on Performance Analysis of Systems & Software, (2012), pp.2-13.
- [5] S. Mathew, R. Krishnamurthy, M. Anders, R. Rios, K. Mistry, and K. Soumyanath, "Sub-500-ps 64-b ALUs in 0.18- $\mu\text{m}$  SOI/bulk CMOS: design and scaling trends" Proceedings of the IEEE International Solid-State Circuits Conference, (2001), pp.318-319.

- [6] Y. Shimazaki, R. Zlatanovici, and B. Nikolic, "A shared-well dual-supply-voltage 64-bit ALU" IEEE Journal of Solid-State Circuits (2004), Vol. 39, N° 3, pp.494-500.
- [7] A. Srivastava, and D. Govindarajan, "A fast ALU design in CMOS for low voltage operation" VLSI Design (2002), Vol. 14, N° 4, pp.315-327.
- [8] J. Kao, and A. Chandrakasan, "Dual-threshold voltage techniques for low-power digital circuits" IEEE Journal of Solid-State Circuits (2000), Vol. 35, N° 7, pp.1009-1018.
- [9] S. Ghosh, P. Ndai, and K. Roy, "A novel low overhead fault tolerant Kogge-Stone adder using adaptive clocking" Proceedings of Design, Automation and Test in Europe, (2008), pp.366-371.
- [10] B. Chatterjee, and M. Sachdev, "Design of a 1.7-GHz low-power delay-fault-testable 32-b ALU in 180-nm CMOS technology" IEEE Transactions on Very Large Scale Integration Systems (2005), Vol. 13, N° 11, pp.1296-1304.
- [11] P. N. Whatmough, S. Das, and D. M. Bull, "A low-power 1GHz razor FIR accelerator with time-borrow tracking pipeline and approximate error correction in 65nm CMOS" Proceedings of IEEE International Solid-State Circuits Conference, (2013), pp.428-429.
- [12] K. Chae, and S. Mukhopadhyay, "A dynamic timing error prevention technique in pipelines with time borrowing and clock stretching" IEEE Transactions on Circuits and Systems I: Regular Papers (2014), Vol. 61, N° 1, pp.74-83.
- [13] K. Chae, C. Lee, and S. Mukhopadhyay, "Timing error prevention using elastic clocking" Proceedings of IEEE International Conference on IC Design & Technology, (2011), pp.1-4.
- [14] D. Kahng, and S. Sze, "A floating gate and its application to memory devices" The Bell System Technical Journal (1967), Vol. 46, N° 6, pp.1288-1295.
- [15] A. Soennecken, U. Hilleringmann, and K. Goser "Floating gate structures as nonvolatile analog memory cells in 1.0µm-LOCOS-CMOS technology with PZT dielectrics" Proceedings of European Solid State Device Research Conference, (1991), pp.633-636.
- [16] Y. Nissan-Cohen, "A novel floating-gate method for measurement of ultra-low hole and electron gate currents in MOS transistors" IEEE Electron Device Letters (1986), Vol. 7, N° 10, pp.561-563.

- [17] T. Morie, T. Matsuura, M. Nagata, and A. Iwata, "A multinanodot floating-gate MOSFET circuit for spiking neuron models" *IEEE Transactions on Nanotechnology* (**2003**), Vol. 2, N° 3, pp.158-164.
- [18] A. Basu, S. Ramakrishnan, C. Petre, S. Koziol, S. Brink, and P. Hasler, "Neural dynamics in reconfigurable silicon" *IEEE Transactions on Biomedical Circuits and Systems* (**2010**), Vol. 4, N° 5, pp.311-319.
- [19] S. Liu, and B. Minch, "Silicon synaptic adaptation mechanisms for homeostasis and contrast gain control" *IEEE Transactions on Neural Networks* (**2002**), Vol. 13, N° 6, pp.1497-1503.
- [20] T. Shibata, and T. Ohmi, "Neuron MOS binary-logic integrated circuits. I. Design fundamentals and soft-hardware-logic circuit implementation" *IEEE Transactions on Electron Devices* (**1993**), Vol. 40, N° 3, pp.570-576.
- [21] T. Shibata, and T. Ohmi, "Neuron MOS binary-logic integrated circuits. II. Simplifying techniques of circuit configuration and their practical applications" *IEEE Transactions on Electron Devices* (**1993**), Vol. 40, N° 5, pp.974-979.
- [22] A. Srivastava, and H. Venkata, "Quaternary to binary bit conversion CMOS integrated circuit design using multiple-input floating gate MOSFETs" *Integration, the VLSI journal* (**2003**), Vol. 36, N° 3, pp.87-101.
- [23] T. Shibata, and T. Ohmi, "A functional MOS transistor featuring gate-level weighted sum and threshold operations" *IEEE Transactions on Electron Devices* (**1992**), Vol. 39, N° 6, pp.1444-1455.
- [24] H. Venkata, "Ternary and quaternary logic to binary bit conversion CMOS integrated circuit design using multiple input floating gate MOSFETs" M.S.E.E. Thesis, Baton Rouge (LA), Louisiana State University (**2002**).
- [25] P. Nigh, and W. Maly, "A self-testing ALU using built-in current sensing" *Proceedings of the IEEE Custom Integrated Circuits Conference*, (**1989**), pp.22.1.1-22.1.4.
- [26] M. Anders, S. Mathew, B. Bloechel, S. Thompson, R. Krishnamurthy, K. Soumyanath, and S. Borkar, "A 6.5GHz 130nm single-ended dynamic ALU and instruction scheduler loop" *Proceedings of the IEEE International Solid-State Circuits Conference*, (**2002**), pp.332-534.
- [27] M. Fojtik, "Bubble razor: eliminating timing margins in an ARM Cortex-M3 processor in 45 nm CMOS using architecturally independent error detection and correction" *IEEE Journal of Solid-State Circuits* (**2013**), Vol. 48, N° 1, pp.66-81.

- [28] T. McConaghy, K. Breen, J. Dyck, and A. Gupta, Variation-Aware Design of Custom Integrated Circuits: a Hands-on Field Guide, Springer Science & Business Media (2012).
- [29] A. Asenov, "Random dopant induced threshold voltage lowering and fluctuations in sub-0.1  $\mu\text{m}$  MOSFET's: A 3-D "atomistic" simulation study" IEEE Transactions on Electron Devices (1998), Vol. 45, N° 12, pp.2505-2513.
- [30] A. Srivastava, and A. Srinivasan, "ALU design using reconfigurable CMOS logic" Proceedings of the IEEE Midwest Symposium on Circuits and Systems, (2002), pp.663-666.
- [31] E. Cortés-Barrón, M. Reyes-Barranca, L. Flores-Nava, and A. Medina-Santiago, "4-bit arithmetic logic unit (ALU) based on neuron MOS transistors" Proceedings of the International Conference on Electrical Engineering, Computing Science and Automatic Control, (2012), pp.1-6.
- [32] Y. Berg, T. S. Lande, O. Naess, and H. Gundersen, "Ultra-low-voltage floating-gate transconductance amplifiers" IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing (2001), Vol. 48, N° 1, pp.37-44.
- [33] J. Lu, S. Young, I. Arel, and J. Holleman, "A 1 TOPS/W analog deep machine-learning engine with floating-gate storage in 0.13  $\mu\text{m}$  CMOS" IEEE Journal of Solid-State Circuits (2015), Vol. 50, N° 1, pp.270-281.
- [34] C. Wang, C. Hsu, S. Liao, and Y. Liu, "A wide voltage range digital I/O design using novel floating N-well circuit" IEEE Transactions on Very Large Scale Integration (VLSI) Systems (2011), Vol. 19, N° 8, pp.1481-1485.
- [35] S. Kim, J. Hasler, and S. George, "Integrated floating-gate programming environment for system-level ICs" IEEE Transactions on Very Large Scale Integration (VLSI) Systems (2016), Vol. 24, N° 6, pp.2244-2252.
- [36] V. Srinivasan, G. Serrano, C. M. Twigg, and P. Hasler, "A Floating-Gate-Based Programmable CMOS Reference" IEEE Transactions on Circuits and Systems I: Regular Papers (2008), Vol. 55, N° 11, pp.3448-3456.
- [37] J. Ramirez-Angulo, and G. Gonzalez-Altamirano, "A new programmable logic family using multiple-input floating-gate transistors" Proceedings of Midwest Symposium on Circuits and Systems, (1997), pp.354-357.
- [38] E. Rodriguez-Villegas, A. Yufera, and A. Rueda, "A 1-V micropower log-domain integrator based on FGMOS transistors operating in weak inversion" IEEE Journal of Solid-State Circuits (2004), Vol. 39, N° 1, pp.256-259.

## BIOGRAPHIES

**Zhou Zhao** received the B.S. in automation and M.S. in Circuit and System from the University of Electronic Science and Technology of China in 2011 and 2014, respectively. He is currently working toward the Ph.D. degree in Electrical Engineering at Louisiana State University, Baton Rouge, LA. His research interests include low power design of general-purpose computing core, neural accelerator, digital calibration and data converter.

**Ashok Srivastava** obtained M. Tech. and Ph.D. degrees in Solid State Physics and Semiconductor Electronics area from Indian Institute of Technology, Delhi in 1970 and 1975, respectively. He joined the Department of Electrical & Computer Engineering of Louisiana State University, Baton Rouge in 1990, and is Wilbur D. and Camille V. Fugler, Jr., Professor of Engineering in the School of Electrical Engineering & Computer Science. He serves on the Editorial Review Board of the several international peer reviewed journals and serves as an Associate Editor on the Editorial Board of IEEE Transactions on Nanotechnology. He is a Life Senior Member of IEEE, Electron Devices, Circuits and Systems, Solid-State Circuits Societies, and Computer Society, Member of IEEE Nanotechnology Council, and Sr. Member of SPIE and Member ASEE. His research interests are: Low Power VLSI Circuit Design and Testability (Digital, Analog and Mixed-Signal); Noise in Devices and VLSI Circuits; Nanoelectronics (Non-classical Device Electronics with focus on Carbon Nanotube, Graphene and other 2D materials for post-CMOS VLSI and Emerging Integrated Electronics; RF Integrated Circuits; Semiconductor Devices Modeling; Radiation-hard Integrated Circuits; and Low-Temperature Electronics.

**Lu Peng** is the Gerard L. “Jerry” Rispono Professor with the Division of Electrical and Computer Engineering at Louisiana State University. He received the bachelor’s and master’s degrees in computer science and engineering from Shanghai Jiao Tong University, China, and the PhD degree in computer engineering from University of Florida. His research focus on computer architecture, memory hierarchy system, reliability, power efficiency, and other issues in processor design. He received an ORAU Ralph E. Power Junior Faculty Enhancement Awards in 2007 and the Best Paper Award (Processor Architecture track) from IEEE International Conference on Computer Design in 2001.

**Saraju P. Mohanty** is a Professor at the Department of Computer Science and Engineering (CSE), University of North Texas (UNT), where he directs the NanoSystem Design Laboratory (NSDL). Prof. Mohanty is an author of 220 peer reviewed publications and three books, and inventor of four patents. His research has been funded by NSF, SRC, and Air Force. He serves on the editorial board of six peer-reviewed international journals, including IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), ACM Journal on Emerging Technologies in Computing Systems (JETC). He is the Editor-in-Chief (EiC) of the IEEE Consumer Electronics Magazine.