

Multiple Sampling and Pixel-Wise Accumulation in CMOS Capacitive Sensor Array System for Real-Time Droplet Analysis

Lin-Hung Lai, Wen-Yue Lin, Yu-Chen Hung, Yu-Hsian Wang, Hsi-Hao Huang, Chen-Yi Lee
 Institute of Electronics, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
 E-mail: {hung880417.ee09, cylee}@nycu.edu.tw

Abstract—Capacitive sensor array (CSA) is vital in precise monitoring for lab-on-chip (LOC) systems. However, shrinking electrode sizes to increase spatial resolution bring challenges like noise interference and large data volumes. This paper presents an FPGA-based system that addresses these problems with multiple sampling (MS) and pixel-wise accumulation (PWA). MS reduces Gaussian noise by sampling multiple frames and retaining only representative data points, while PWA compresses data using Block RAM and minimal combinational logic, reducing size from 118 Mb to 0.46 Mb and boosting SNR to 25.30 dB. The system enables real-time monitoring every 5 seconds instead of 17 minutes, with pipeline sensing and transmission further optimizing sensing time. Experiments demonstrate its effectiveness in distinguish between droplets and monitor evaporation in real time. MS and PWA can be easily integrated into future chip designs, offering scalable solutions for fast and precise monitoring in LOC environments.

Index Terms—Lab-on-Chip, Capacitance Sensor Array, Pixel-Wise Accumulation, Multiple Sampling, Droplet Sensing.

I. INTRODUCTION

The demand for real-time, high-precision monitoring in biotechnology has driven advancements in capacitive sensor array (CSA) for lab-on-chip (LOC) systems. These CSAs, based on complementary metal-oxide-semiconductor (CMOS) technology, are widely used for detecting capacitance variations across two-dimensional surfaces, making them essential for applications such as droplet analysis [1]–[3], cell growth monitoring [4]–[7], DNA detection [8], particle identification [9], drug screening [10], and super-resolution imaging [11].

However, as electrode sizes shrink to improve spatial resolution, real-time data acquisition and noise reduction become critical issues. Smaller electrodes with higher throughput decrease the signal-to-noise ratio (SNR) and pose challenges in processing large data volumes in real time. For instance, in [3], scanning 256 electrodes required approximately 7 minutes. Similarly, the CSA in [12] achieved 43 fps at 10 MHz clock speed, but processing and transmitting data from 7200 electrodes with a 4×4 fusion-pixel arrangement took about 17 minutes due to the transmission of 118 Mb of data at a 115200 baud rate. These limitations highlight the need for more efficient data processing and noise reduction at the system level to ensure fast and reliable sensor readings.

This work was supported by National Science and Technology Council (NSTC) under Project 113-2321-B-A49-013 and in part by chip fabrication services from Taiwan Semiconductor Research Institute, Taiwan, R. O. C.

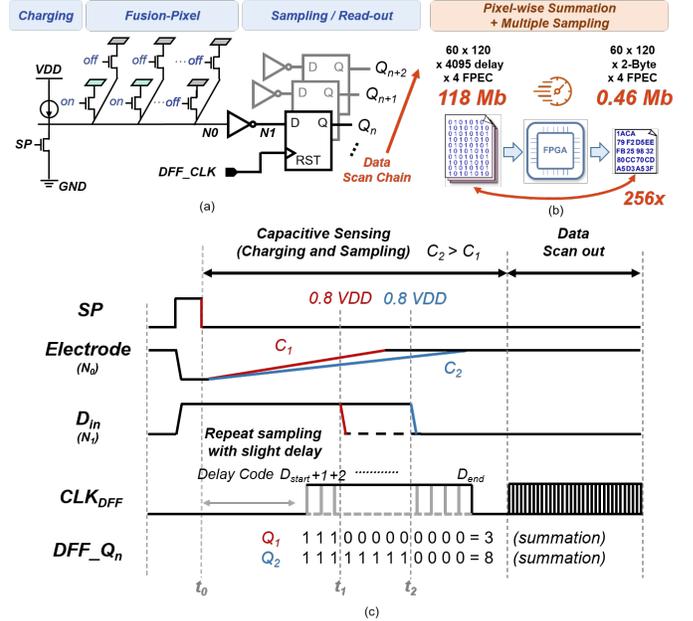


Fig. 1: ts-TDC-based CSA circuit behavior. (a) circuit diagram (b) PWA and MS diagram and (c) timing diagram

In this work, we propose an integrated noise reduction and data compression solution for time-sharing time-to-digital converter (ts-TDC) based CSAs [12], [13]. By combining multiple sampling (MS) and pixel-wise accumulation (PWA) techniques on a field programmable gate array (FPGA), we compress all the CSA’s binary output into a single 16-bit value for each electrode, reducing the data size from 118 Mb to 0.46 Mb, as shown in Fig 1(b). MS further reduces random noise by averaging data from multiple frames, achieving this with minimal additional scanning time and maximum 256 times of multiple sampling result for each electrode. The following sections will introduce the CMOS capacitive sensor array, present optimized noise reduction algorithms, and demonstrate how this system improves image quality and accelerates data processing in high-throughput CSA applications.

II. CMOS CAPACITIVE SENSOR ARRAY

The ts-TDC based CSA is designed to convert capacitance values into charging time differences, followed by adjusting

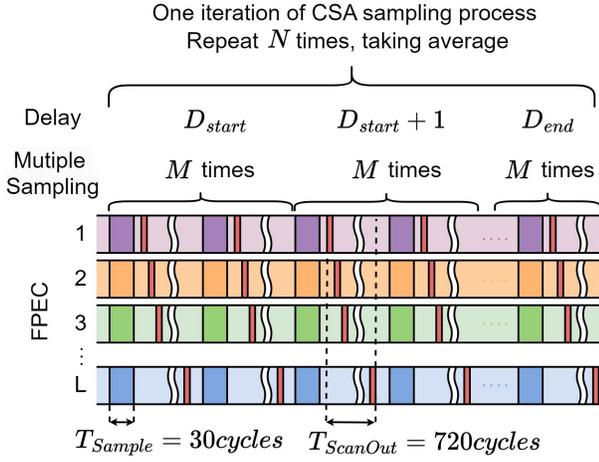


Fig. 2: Multiple sampling timeline for ts-TDC based CSA

the D-flip-flop (DFF)'s sampling times to determine the fine differences between these charging times. This approach allows for the sharing of components on the time axis, reducing the area required to replicate multiple TDCs for whole-array and enabling the system to accommodate a larger number of electrodes. As shown in Fig. 1(a), three key components are involved in each electrode: the charging circuit, fusion-pixel electrode cell (FPEC), and sampling unit. Charging circuit consists of a series of PMOS transistors to provide a small current. The FPEC can be programmed to adjust the electrode size by toggling the connected NMOS transistors, and the sampling unit is a multiplexed DFF either to sample the charging time or to scan out the data with a daisy scan-chain.

The sampling procedure is illustrated in Fig. 1(c), the charging process begins when the sensing pulse SP falls, prompting the current source to charge the FPEC until the voltage at node N_0 reaches approximately 80% of the supply voltage. A high-skew inverter then converts the voltage at N_0 into a sharp digital signal at N_1 . Since C_1 is smaller than C_2 , node N_1 of C_1 experiences an earlier voltage drop than N_1 of C_2 . The DFF sampling process begins with a clock signal CLK_{DFF} at initial delay codes D_{start} , getting both Q_1 and Q_2 to logic 1. The sampling process is repeated with a slight delay increment $D_{start} + 1$ in DFF sampling times. After the delay reaches time t_1 , Q_1 transitions to logic 0 while Q_2 remains at logic 1. Subsequently, after t_2 , Q_2 also transitions to logic 0. By simply summing all the Q values over time, the relative capacitance differences between C_1 and C_2 can be quantified. Note that the sensing window, from negative edge of SP to negative edge of N_1 , would be interference by random noise, resulting in a uncertainty in the measurement.

III. SYSTEM DESIGN AND IMPLEMENTATION

A. Random Noise Reduction

Random noise, often modeled as Gaussian noise, is a common source of interference in CSA systems, leading to inaccuracies in sensor readings. To mitigate this noise, the multiple sampling algorithm 1 is implemented on the FPGA to

Algorithm 1 Multiple Sampling on CSA

Input: $N, M, D_{start}, D_{end}, \tau_{th}, L$.

Output: Compressed array data stored in FIFO.

```

1: Reset BRAM1 and BRAM2 to zeros.
2: for  $n = 1$  to  $N$  do
3:   for  $delay = D_{start}$  to  $D_{end}$  do
4:     for  $m = 1$  to  $M$  do
5:       Charge & Sample with delay
6:       for  $addr = 1$  to  $L$  do
7:          $a, b \leftarrow BRAM1(addr), BRAM2(addr)$ 
8:          $CSA_{data} \leftarrow ScanChain.pop$ 
9:         if  $m \neq M$  and  $n \neq N$  then  $\triangleright$  Case 1
10:           $BRAM1(addr) \leftarrow CSA_{data} + a$ 
11:         else if  $m = M$  and  $n \neq N$  then  $\triangleright$  Case 2
12:           if  $CSA_{data} + D_1 > \tau_{th}$  then
13:              $BRAM2(addr) \leftarrow b + 1$ 
14:           else
15:              $BRAM2(addr) \leftarrow b$ 
16:           end if
17:            $BRAM1(addr) \leftarrow 0$ 
18:         else  $\triangleright$  Case 3
19:           if  $CSA_{data} + a > \tau_{th}$  then
20:              $FIFO.push((b + 1)/N)$ 
21:           else
22:              $FIFO.push(b/N)$ 
23:           end if
24:            $BRAM1(addr), BRAM2(addr) \leftarrow 0$ 
25:         end if
26:       end for
27:     end for
28:   end for
29: end for

```

average data from multiple frames. The algorithm operates in two main stages, as depicted in Fig. 2: (1) multiple sampling, where each delay code is sampled M times before moving to the next one, and (2) repeating this process N times across a given delay code range, from D_{start} to D_{end} . Both stages are aimed at reducing random noise, however, a large M may introduce motion blur effect due to long sampling time. Hence, the optimal M and N are determined by the trade-off between noise reduction and motion blur effect. To ensure efficient data storage with block random access memory (BRAM) and minimum combinational logic, BRAM1 and BRAM2 are used to store the accumulated data for each electrode in stage (1) and (2), respectively, while a FIFO buffer is used to transmit the averaged data to the personal computer (PC).

In algorithm 1, each iteration starts with charging the electrode and sampling by capturing a 1 or 0 after waiting for $delay$, taking about 30 cycles denoted as T_{Sample} . The captured data is then retrieved from the scan chain and stored in either BRAM1 or BRAM2. Scan-out data takes 720 cycles, denoted as $T_{ScanOut}$, to complete the entire scan chain with length L . The algorithm handles three specific cases: (1) when $m \neq M$ and $n \neq N$, (2) when $m = M$ and $n \neq N$, and (3)

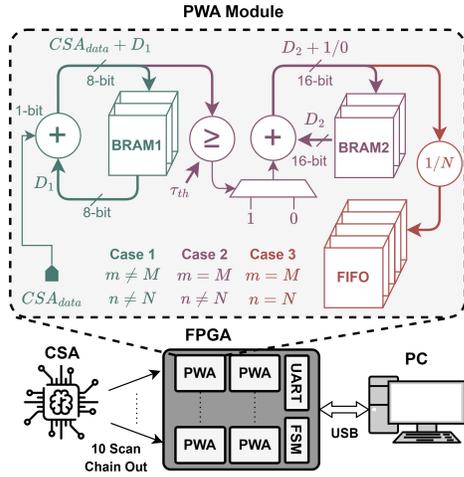


Fig. 3: FPGA implementation for PWA and MS

when $m = M$ and $n = N$. In case (1), data is accumulated in BRAM1. In case (2), the data in BRAM1 is compared to the threshold value τ_{th} , and the results are accumulated in BRAM2. τ_{th} is set to half of M because exceeding this threshold indicates higher confidence in declaring the value as 1. Finally, in case (3), the accumulated data in BRAM2 is averaged by dividing by N and pushed into the FIFO buffer for transmission. Here, dividing by N is equivalent to shifting right by \sqrt{N} . Note that all cases perform along the scan chain at the same time, i.e. the data is pop out from the scan chain and stored in BRAM1, BRAM2, or FIFO in parallel.

B. FPGA Implementation

Due to the large amount of data generated by MS, directly transmitting all raw data to the PC is inefficient. Therefore, an FPGA-based architecture, shown in Fig. 3, was implemented to optimize throughput and reduce computational overhead. The design incorporates "PWA modules," each consisting of 8-bit BRAM1, 16-bit BRAM2, a 16-bit FIFO, and combinational logic, responsible for data storage and processing. The number of PWAs was selected to match the number of scan-chains in the CSA, with memory depth proportional to the scan-chain length. In our design, we implement 10 scan-chains with 720 electrodes each, requiring 10 PWAs.

Fig. 4 illustrates the memory hierarchy and data flow. BRAM1 can store up to 256 times MS results per electrode, denoted as $\sum_{i=1}^M s_i$, where s_i is the sampling value for the i -th times. After M times, 8-bit data is compressed to 1-bit representative value, and stored in BRAM2. BRAM2 stores 16-bit accumulated results for each electrode, denoted as $\sum_{k=1}^N \sum_{j=D_{start}}^{D_{end}} S_{j,k}$, where $S_{j,k}$ is the compressed value for the j -th delay code in the k -th iteration. Maximum number of N depends on the delay range $D_{range} = D_{end} - D_{start} + 1$. After averaging by N , data is pushed into FIFO and transmitted to the PC via UART at a 115200 baud rate. The overall sampling time is defined by $N \times D_{range} \times M \times (T_{Sample} + T_{ScanOut})$, where $T_{Sample} + T_{ScanOut}$ is 750 cycles (75

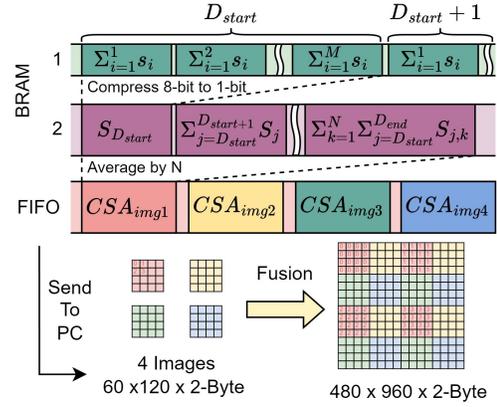


Fig. 4: Memory hierarchy and data flow in our system design

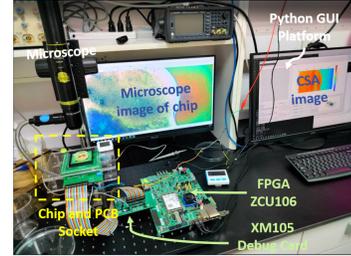


Fig. 5: Experimental setup for CSA

$\mu s @ 10 MHz$). Given $D_{range} = 500$, $M = 32$, and $N = 1$, the total sampling time is 1.2 seconds per frame. Sampling and transmission are pipelined to run concurrently; while the UART transmission of 0.46 Mb takes 4 seconds, the system initiates the next sampling during this time, overlapping operations to minimize idle time and optimize throughput despite the UART occupying most of the frame time. Hence, this system can achieve a frame rate of about 5 seconds per frame for a 4×4 FPEC arrangement.

C. Experimental Setup

As shown in Fig. 5, MS and PWA are implemented on an FPGA (Xilinx ZCU106) to collect and compress data from the CSA. The compressed data is transmitted to a PC via USB for further processing such as calibration, fusion 4 images (60×120) into 1 whole frame (480×960), and 2D capacitance image display through Python. A microscope is used to verify the consistency between the capacitance and optical images. The delay range is set as $D_{start} = 1000$, $D_{end} = 1500$ to cover sample value. The N is fixed to 1 and only M is varied to evaluate the impact of MS in the following experiments, and FPEC is set as 4×4 to balance between spatial resolution and sensitivity.

IV. EXPERIMENTAL RESULT

A. Impact of Multiple Sampling Iterations

To evaluate the impact of number of M on the CSA's performance, we measured the capacitance of a deionized (DI) water on the CSA five times. The average output values across

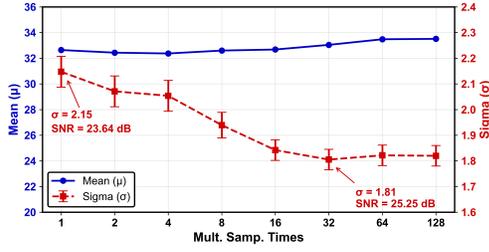


Fig. 6: Impact of multiple sampling iterations on mean and standard deviation of CSA data

480×960 pixels were analyzed for the mean and standard deviation as M increased from 1 to 128. The results, as shown in Fig. 6, demonstrate that while the mean remained stable, the standard deviation decreased from 2.15 at $M = 1$ to 1.81 at $M = 32$, resulting in an improved SNR, calculated as $10 \times \log_{10}(\mu^2/\sigma^2)$, from 23.64 dB to 25.25 dB, respectively. Note that improvements were most up to $M = 32$, beyond which further iterations showed diminishing returns. Therefore, selecting $M = 32$ strikes a balance by minimizing noise while reducing acquisition time.

B. Dual Droplets Sensing

The dual-droplet experiment aimed to extend the analysis from last experiment by comparing the impact of $M = 1$ and $M = 32$ on CSA image quality, focusing on the shape and the values of two distinct droplets: DMEM (culture medium, CM) and DI water. Calibration data (C_{calib}) was collected first, followed by raw capacitive responses (C_{sample}), with final measurements obtained by subtracting the baseline ($C_{\text{sample}} - C_{\text{calib}}$). The results in Fig. 7(a) and (b) show differentiation between the two droplets. In (b) $M = 32$, the random noise decreased due to MS, but this also made the circuit-induced pattern noise more apparent compared to (a) $M = 1$, suggesting that future efforts could focus on this.

In the blue-highlighted region of interest (ROI), the mean values for CM and DI water remained stable, while the standard deviation improved, with $\sigma_{CM=1} = 1.5$ reducing to $\sigma_{CM=32} = 1.31$, and $\sigma_{DI=1} = 1.52$ reducing to $\sigma_{DI=32} = 1.22$, reflecting better measurement precision with increased sampling. To demonstrate the CSA’s ability to reconstruct droplet shapes in Fig. 7, OpenCV’s morphologyEx function with a (5,5) kernel was applied to (b), clearly outlining the droplet boundaries in (d), which closely matched the optical image in (c). Thus, the reduction in random noise through MS allowed for more precise measurement of both droplet shapes and values, as the decrease in standard deviation highlights the improved data accuracy.

C. Real-time Evaporation Monitoring

To demonstrate the system’s real-time monitoring capabilities, a DI water droplet was placed on the CSA, and the evaporation process was captured over 70 frames, with each frame taken every 5 seconds, and $M = 32$. The results, shown in Fig. 8, illustrate the gradual reduction in the droplet

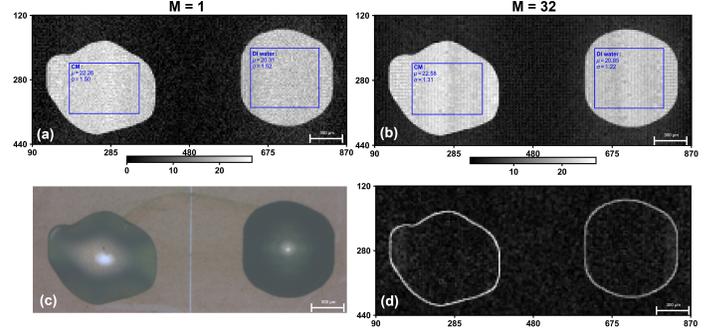


Fig. 7: Dual droplets sensing of CM and DI water (a) $M = 1$ (b) $M = 32$ (c) optical image (d) edge detection

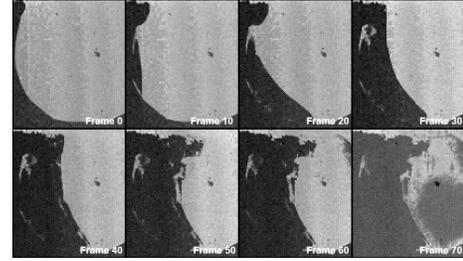


Fig. 8: Real-time DI water evaporation monitoring from Frame 0 to Frame 70, with 5 secs per frame

size over time, with the capacitive response decreasing as the water evaporates. The full video of the experiment in Fig 8 is available at: <https://dx.doi.org/10.21227/h1m6-9c69> [14]. Comparison table is shown in Table I.

V. CONCLUSION

This paper presents an FPGA implementation for CMOS CSA that employs PWA and MS to achieve efficient data compression and noise reduction. PWA reduced data size from 118 Mb to 0.46 Mb and cut processing time from 17 minutes to 5 seconds per frame. MS improved the SNR to 25.30 dB, enabling differentiation between culture medium and DI water and allowing real-time water evaporation monitoring. This FPGA implementation enhances real-time monitoring and lays the groundwork for integrating MS and PWA into chip-level designs—advancements crucial for future applications requiring rapid and precise monitoring, such as super-resolution imaging and high-frame-rate biological experiments including droplet analysis and cell movement observation.

TABLE I: Comparison Table

	TCASII'23 [12]	This work
Method	Single sampling	Multiple sampling
FPGA function	Read-out only	Pixel-wise accumulation
Data size	~118 Mb	0.46 Mb
Time for whole frame	~17 minutes	~5 seconds
Noise level (SNR)	-	25.3 dB

REFERENCES

- [1] H. O. Tabrizi, S. Forouhi, and E. Ghafar-Zadeh, "A high dynamic range dual 8×16 capacitive sensor array for life science applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, no. 6, pp. 1191–1203, 2022.
- [2] S. Forouhi, H. O. Tabrizi, A. Panahi, S. Magierowski, and E. Ghafar-Zadeh, "Novel CMOS thermo-capacitive sensing method for lab-on-chip applications," in *2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2023, pp. 1–5.
- [3] H. Osouli Tabrizi, S. Forouhi, T. Azadmousavi, and E. Ghafar-Zadeh, "A multidisciplinary approach toward CMOS capacitive sensor array for droplet analysis," *Micromachines*, vol. 15, no. 2, p. 232, 2024.
- [4] N. Couniot, L. A. Francis, and D. Flandre, "A 16×16 CMOS capacitive biosensor array towards detection of single bacterial cell," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 2, pp. 364–374, 2016.
- [5] G. Nabovati, E. Ghafar-Zadeh, A. Letourneau, and M. Sawan, "Towards high throughput cell growth screening: A new CMOS 8×8 biosensor array for life science applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 2, pp. 380–391, 2017.
- [6] B. P. Senevirathna, S. Lu, M. P. Dandin, J. Basile, E. Smela, and P. A. Abshire, "Real-time measurements of cell proliferation using a lab-on-CMOS capacitance sensor array," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 3, pp. 510–520, 2018.
- [7] R. Abdelbaset, Y. El-Sehrawy, O. E. Morsy, Y. H. Ghallab, and Y. Ismail, "CMOS based capacitive sensor matrix for characterizing and tracking of biological cells," *Scientific Reports*, vol. 12, no. 1, pp. 1–10, 2022.
- [8] C. Stagni, C. Guiducci, L. Benini, B. Ricco, S. Carrara, B. Samori, C. Paulus, M. Schienle, M. Augustyniak, and R. Thewes, "CMOS DNA sensor array with integrated A/D conversion based on label-free capacitance measurement," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2956–2964, 2006.
- [9] F. Widdershoven, A. Cossettini, C. Laborde, A. Bandiziol, P. P. van Swinderen, S. G. Lemay, and L. Selmi, "A CMOS pixelated nanocapacitor biosensor platform for high-frequency impedance spectroscopy and imaging," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 6, pp. 1369–1382, 2018.
- [10] G. Nabovati, E. Ghafar-Zadeh, A. Letourneau, and M. Sawan, "Smart cell culture monitoring and drug test platform using CMOS capacitive sensor array," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 4, pp. 1094–1104, 2019.
- [11] K. Hu, J. Ho, and J. K. Rosenstein, "Super-resolution electrochemical impedance imaging with a 512×256 CMOS sensor array," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, no. 4, pp. 502–510, 2022.
- [12] L.-H. Lai, W.-Y. Lin, Y.-W. Lu, H.-Y. Lui, S. Yoshida, S.-H. Chiou, and C.-Y. Lee, "A 460800 pixels CMOS capacitive sensor array with programmable fusion pixels and noise canceling for life science applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 5, pp. 1734–1738, 2023.
- [13] H.-Y. Liu, L.-H. Lai, W.-Y. Lin, Y.-W. Lu, Y.-W. Lin, and C.-Y. Lee, "A $2.56\text{-}\mu\text{s}$ dynamic range, 31.25-ps resolution 2-d vernier digital-to-time converter (DTC) for cell-monitoring," in *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2024, pp. 1–5.
- [14] L.-H. Lai, "Demonstration of real-time data acquisition in CMOS capacitive sensor arrays: Evaporation monitoring," 2024. [Online]. Available: <https://dx.doi.org/10.21227/h1m6-9c69>