# Using Multiple-Input NEMS for Parallel A/D Conversion and Image Processing

Kaisheng Ma[1], Nandhini Chandramoorthy[1], Xueqing Li[1], Sumeet Kumar Gupta [2], John Sampson[1], Yuan Xie[3], Vijaykrishnan Narayanan[1]

[1]Dept. of Computer Science and Engineering, [2]Dept. of Electrical Engineering, The Pennsylvania State University
[3]Dept. of Electrical and Computer Engineering, University of California at Santa Barbara
Email: {kxm505,nic5090,lixueq,sampson,vijay}@cse.psu.edu, skgupta@psu.edu, yuanxie@ece.ucsb.edu

## ABSTRACT

The technology advancements in semiconductor process have led to rapid progress in design and fabrication of multiple-input Nano-electro-mechanical relays/switches (NEM relays/NEMS). This work explores the design space of implementing image processing algorithms using a hybrid multiple-input NEMS–CMOS architecture. Different from the existing the approaches of building logic gates (e.g. INV, AND, OR, NAND, NOR, etc.) as conventional CMOS circuits, using NEMS, this work takes advantages of the electrical and mechanical physical features of the NEM relays to implement image processing algorithms. Simulation results show that the multiple-input NEM relays can be applied for decision-making and compare-select image processing algorithms. Moreover, we show that NEM relays can operate as parallel analog-to-digital converters (ADCs) with advantages of leakage reduction and power efficiency.

## Keywords

ADC, computation in physics, image processing, NEMS.

## 1. INTRODUCTION

For nano-scaled device technology, NEMS relays demonstrate prominent advantages due to their non-leakage feature, aggressive supply voltage(VDD) scaling [1-5]. In traditional works, various digital integrated circuit building blocks including logic, memory, and clocking structures implemented purely with NEMS has been demonstrated [1-5]. But in this work, we open another path for using the NEMS as physical computing devices. By taking advantages of the mechanical and electrical features of the NEMS, the physical force can be used to make the computation. And this process can been done momentarily. All the other supporting circuits are used to read and transform the inputs and the results to digital signals.

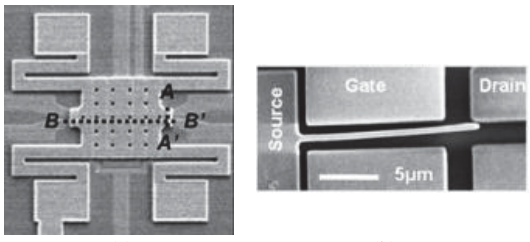
In spite of the zero leakage advantage, the NEMS technology still faces several challenges [1-2]. One is the relatively lower speed (tens of ns to ms) compared to CMOS (less than 1 ns). This is because the switching between the off- and on-states is based on the mechanical movement of the electrode. Recently, a higher speed has been predicted with the device size scaling [3]. The other challenge is the higher on-state source/drain resistance which may further limit the applications. According to the Scaled Model in [2], the NEMS on-state resistance is in the range of 540~900 Ohm, while the most advanced NEMS has shown an on-state resistance of approximately 500 Ohm [3]. The third challenge is the comparatively larger cell area (about 50x50 $um^2$) [1][2]. As for the duration issue in the previous works, in the most advanced NEMS it is generally considered not a disadvantage with the duration up to 60 billion times [2].

Many works have focused on designing NEMS-based logic circuits to achieve similar functions to the conventional CMOS [1, 2, 5, 6]. Some basic logic cells have been designed including INV, AND, OR, NAND, NOR, XOR, latch, 32-bit full adder, etc. Multi-input NEMS design has been proposed to improve the original NEMS device [1]. In this case, more complex logic functions could be implemented with less delay [1].

In traditional digital image processing platforms, the pixel digitization and other higher-level digital signal processing are carried out separately by sensors and digital processors, which results in high transmission bandwidth with redundant information [8-14]. In the contrast, in this work, we explore the design space of implementing image processing algorithms using 3D integration of image sensors, CMOS and multiple-input NEMS. Instead of building logic gates (e.g. INV, AND, OR, NAND, NOR, etc.) in conventional CMOS circuits, this work takes advantages of the electrical, mechanical, and physical force features of the NEMS to implement certain image processing algorithms. With cooperative small-scale CMOS circuits, the multiple-input NEMS array demonstrates the potential to work as a parallel power-efficient analog-to-digital converter (ADC) array, and is capable of image processing such as Gaussian filtering, edge detection, saliency, as well as decision-making and compare-select image processing algorithms.

In the rest of this paper, Section 2 introduces the NEM relay physical layout and models. Section 3 describes how to use NEMS array to implement different image processing algorithms. Section 4 provides the simulation results and discussions. Section 5 benchmarks the speed and power performance, and integration with CMOS architecture. Section 6 concludes this paper.

## 2. NEMS MODEL INTRODUCTION




(a) (b)

**Figure 1. Two types of NEMS structure from [1] and [5].**

The NEMS physical layout and electrical model have been described in detail in [2, 5, 6]. Figure 1 illustrates the physical layout of the NEMS [1][5]. Figure 1(a) is electrically a four terminal devices, including gate, drain, source and body, which is similar to a conventional bulk MOSFET. As in Figure 1, the gate

978-1-4799-8719-1/15 $31.00 © 2015 IEEE
DOI 10.1109/ISVLSI.2015.114

is movable and suspended by spring-like flexures above the body, drain and source electrodes [1]. When a sufficiently large voltage is applied to the gate, the electrostatic force pulls the gate towards the bottom plate. As a result, the drain and source electrodes are connected through the channel, making the NEMS work in the "turned-on" state in Figure 2(a). Without the sufficiently large gate voltage, the source and drain electrodes are disconnected from the channel and the NEMS is working in the "turned-off" state. Figure 1(b) is another structure with two gates. In this paper, we propose a NEMS structure based on it, as shown in Figure 2(b). By splitting the two gates into smaller ones, this structure supports multiple inputs. Because the structure's gates are in two opposite directions, the force is equivalent to subtract operation.

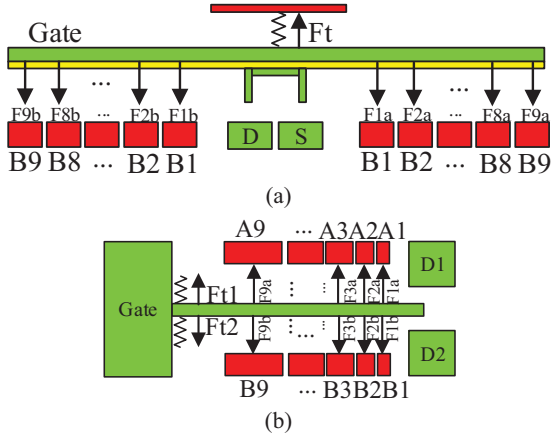



**Figure 2. Illustration of the two different NEMS devices: (a) Structure based on Figure 1(a) for add operation. (b) Structure based on Figure 1(b) for add and subtract operation.**

In the following part of this section, the physical features of the NEMS based on Figure 1(a) will be introduced as an example. Because the NEMS operation states are determined by the physical contact between the channel and the drain/source terminals, a high on-to-off current ratio is achievable. Different from the electrical switching mechanism in CMOS transistors, the switching delay of an equivalent NEMS is comparatively larger. Despite of such a larger delay, the NEMS has significantly lower leakage, and higher energy efficiency.

The behavior of the NEMS could be accurately modeled through a nonlinear second-order differential equation to describe and predict the mechanical and electrical actions.

$$m \cdot x'' = F(x) - b \cdot x' - k \cdot x, \tag{1}$$

where $m$ is the gate mass, $x$ is the displacement of the gate, $b$ is the damping coefficient, $k$ is the effective spring constant of the gate structure, and $Felec(x)$ is the nonlinear electrical force between the gate and the body:

$$F(x) = \frac{\varepsilon_0 A_{ov} V_{gb}^2}{2(g_0 - x)^2}. \tag{2}$$

In (2), $\varepsilon_0$ is the permittivity of free space, $A_{ov}$ is the overlap area between the gate and body electrodes, $g_0$ is the normal gap between the electrodes without the electrical force, and $V_{gb}$ is the gate-body voltage. The required voltage to snap the gate-body structure shut is called the pull-in voltage, $V_{pi}$:

$$V_{pi} = \sqrt{\frac{8}{27} \cdot \frac{k g_0^3}{\varepsilon_0 A_{ov}}}. \tag{3}$$

The turn-on delay is determined by (1). The gate mass $m$ and the spring constant $k$ affect the mechanical turn-on delay $t_{mech}$ in a way that could be modeled as:

$$t_{mech} = \sqrt{\frac{m}{k}} \cdot \left( \frac{V_{pi}}{|V_{gb}|} \right). \tag{4}$$

To accurately model the switching behavior of the NEMS, electrical parameters are also extracted. The contact resistance between the channel and the source/drain $R_{con}$ is modeled as

$$R_{con} = \frac{4\rho\lambda}{3A_r}, \tag{5}$$

where $\rho$ is the contacting material resistivity, $\lambda$ is the mean free path of electrons in the contact material, and $A_r$ is the effective contact area given by

$$A_r \approx \frac{F_{elec}(g_d)}{\xi H}. \tag{6}$$

Some key parameter values are summarized in Table I, with the others the same as [2]. The gate-body capacitance ($C_{gb}$ and $C_{gc}$) is modeled as

$$C_{gb}(x) = \frac{\varepsilon_0 A_{ov}}{g_0 - x}, C_{gc} = \frac{\kappa_{gox} \varepsilon_0 A_{ch}}{t_{gox}}, \tag{7}$$

where $\kappa_{gox}$ is the relative permittivity of the gate oxide, $A_{ch}$ is the gate-channel overlap area, and $t_{gox}$ is the gate oxide thickness.

**Table 1. Parameter of the NEMS in this paper [2]**

| Parameter | Device before scaling | Scaled Device |
|---|---|---|
| $A_{ov}$ [μm$^2$] | 450 | 0.77 |
| $R_{on}$ [Ω/contact] | ~0.1 | 40-400 |
| $R_{pox}$ [Ω/contact] | 500 | 500 |
| $t_{mech\_turn\text{-}on}$ [μS] | 34 | 0.02-0.08 |
| $t_{mech\_turn\text{-}off}$ [μS] | 3 | 0.002-0.008 |
| $t_{elec}$ [pS] | 304.4 | 2.5-3.5 |
| Duration [times] | 60e10 | unknown |
| Stability | 2000g | unknown |

# 3. HYBRID CMOS & NEMS IMAGE PROCESSING (NEMSIP)

In this section, NEMS IMAGE PROCESSING (NEMSIP) is proposed for various image processing algorithms. The key idea of NEMSIP is computing in physics. All the CMOS circuits are designed as supporting peripherals.

## 3.1 NEMSIP for Figure 1(a) Structure

In NEMS, the electrical force in (2) could be rewritten as:

$$F(x) = \frac{\varepsilon_0}{2(g_0 - x)^2} * \left[ \sum_{i=1}^{n} A_{ovi} * (V_g - V_{bi})^2 \right], \tag{8}$$

where $V_{bi}$ is output from the CCD or CMOS sensor array, $V_g$ is the gate voltage, $A_{ovi}$ is the area of inputs for each pixel in the back layer, and $x$ is the distance between the gate layer and back layer. Once the voltage is applied to NEMS, the computation is carried out by physics. All the CMOS peripherals are functioning for data reading and A/D conversion.

Based on such NEMS computation mechanism, Figure 3 illustrates the readout circuits and ADCs. There is an 8-bit self-adder, which adds by 1 on each rising clock edge. This adder output is then converted to an analog voltage by the DAC. It is apparent that all NEMS gates are connected together and driven

by the same DAC output. Therefore, only one adder and one DAC are required as the CMOS circuits. The adder output is then fed to an 8-bit flip-flop chain. With the voltage between the NEMS back and gate layers increasing, the force will enhance until the two layers are connected through the channel to turn on the NEMS and trigger the flip-flop to maintain the digital data. To ensure that the DAC and the NEMS stabilize within each clock cycle, the clock cycle should be carefully designed to meet the NEMS turn-on and turn-off timing requirement.

The speed of this NEMS-based parallel ADC array could be improved by replacing the self-adder by a self-subtracter. When the dimples pump up, the digital outputs would be ready. The improved speed originates from the fact that the mechanical turn-off delay is much smaller than the turn-on delay [14], in Table 1.

We propose an efficient DAC implementation for the NEMS-based ADC, as shown in Figure 4. Compared with Figure 3, the DAC is replaced with a constant current to charge a capacitor. Thus, the voltage of the capacitor increases linearly with time. A buffer is added between the charger and the NEMS gate for the purpose of isolation. The constant charging is designed in a way that the charging time matches the time required for the self-adder to add up to 255 from 0. Every time when the self-adder becomes 8'b0, the capacitor will be completely discharged to prepare for the next computation readout. Similarly, if a self-subtracter is used instead of the self-adder, the constant-current charger should be replaced by a constant-current discharger accordingly.

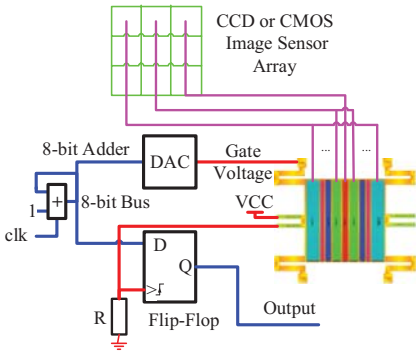



**Figure 3. NEMSIP with a DAC.**

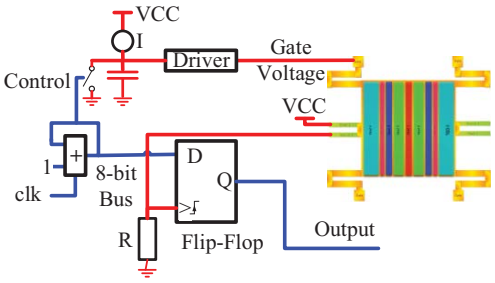



**Figure 4. NEMS-based ADC structure with a constant current charger.**

Another structure is based on the successive-approximation-register (SAR) structure, in which the NEMS comparison result is used to control the DAC so that the output of the DAC could approximate to the final state by at most $N$ steps for an $N$-bit ADC. The advantage of this structure is high speed, as compared with the structure in Figure 3 or Figure 4 which needs $2^N$ steps of DAC output changes in the worst case to finish the analog-to-digital conversion. The disadvantage is more power consumption due to the feed-back operations which results in a dynamic change in the output of the DAC and consequently the loss of the simple DAC implementation shown in Figure 4. For the image processing tasks in this paper, we'll use the structure in Figure 3 and Figure 4 which provides sufficient performance while enables a quick evaluation of the parallel ADC application using NEMS.

In such a CMOS-NEMS hybrid system, the NEMS are working as comparators, which turn the system into parallel ADCs. Straightforwardly, these comparators could be implemented in CMOS circuits as well. However, using CMOS circuits to implement such comparators will result in significant power consumption in terms of energy per comparison. Moreover, such NEMS-based comparators have zero standby leakage current, which outperforms the CMOS comparators.

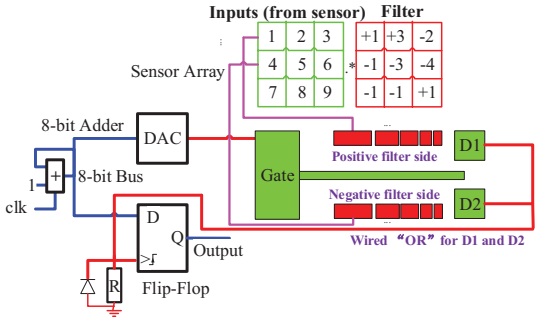



**Figure 5. NEMSIP using the NEMS in Figure 1(b).**

## 3.2 NEMSIP for Figure 1(b) Structure

Using the NEMS structure in Figure 1(b) instead of Figure 1(a), the subtraction operation could be efficiently implemented as shown in Figure 5. The NEMS in Figure 5 has two opposite force directions, achieving a subtraction operation. The temporal differential computation can also be implemented by this Figure 5. structure within 2 phrases. In phrase 1, the pixels are connected to positive filter side, while in phrase 2, connected to negative filter side. In this way, temporal differential can be computed. The computation function becomes

$$F(x) = \frac{\varepsilon_0}{2(g_0 - x)^2}\left[\sum_{pi=1}^{n} A_{ov,pi}(V_g - V_{b,pi})^2 - \sum_{nj=1}^{n} A_{ov,nj}(V_g - V_{b,nj})^2\right], \quad (9)$$

where *pi* represents the positive gate controls in the NEMS and *nj* represents the negative gate controls.

The voltage of the components *D1* and *D2* in Figure 5 should be high enough to be capable of triggering the flip-flops. With such a voltage range requirement, the structure in Figure 5 is more suitable for compare-and-select algorithms, such as the motion estimation in JPEG and H.264.

## 3.3 PIPELINED NEMSIP

The challenge of the structure in Figure 5 is the routing and wiring for NEMS, especially in the scaled devices. To mitigate challenge, we propose the pipelined NEMSIP in Figure 6 to implement a 5*5 filter with three 3*3 NEMS. In the Pipeline Phase 1 in Figure 6, the 25-pixel input window is divided into three groups with 9, 9 and 7 pixels. The outputs are analog gate voltage instead of digital bits. Between Pipeline 1 and Pipeline2, a capacitor is used to store the analog charge. SW2 is

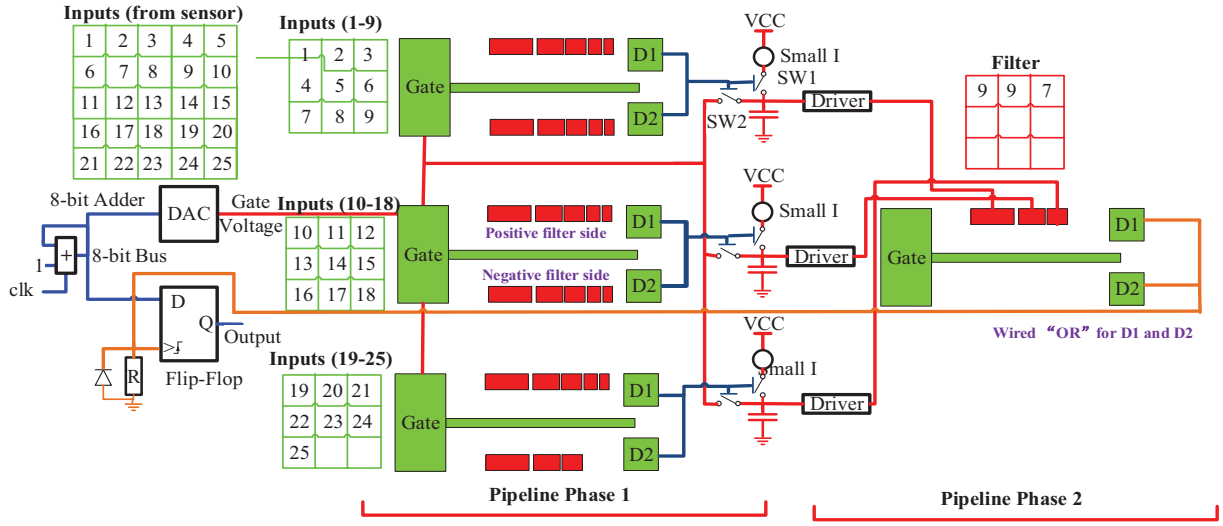



**Figure 6. The pipelined NEMSIP structure to implement a 5*5 filter with 3*3 NEMS.**

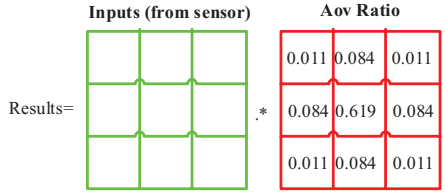



**Figure 7. Gaussian filter implementation.**


**Figure 8. Gaussian filter results comparison.** (a), Lenna grey scale 128*128, as inputs to NEMSIP. (b), Matlab results. (c), NEMSIP result. (d), Larger input scale. (e), Smaller input scale affects darkness. (f),Larger scale of gate voltage affects output to be darker. (g), Smaller scale of gate affects output quality. (h), high-speed results, with other setups same as (c). (i), high-speed mode with adjusted voltage scales.

turned-off, and SW1 is turned-on during Phase 1. In Pipeline Phase 2, the analog results from Phase 1 are inputs. During Phase

2, SW2 is turned-on, while SW1 is turned-off for the current source to compensate the capacitor leakage. The filter from Phase two is 9, 9, 7 to represent the weight.

## 4. MAPPING AND RESULTS

In this section, several image processing and video compression algorithms are mapped to NEMSIP. In this work, the mechanical and electrical parameters in [2][5][6], as well as the delay by the DAC and wires, are employed in the simulation platform. Although the model employed is a behavioral model and does not include second-order effects, the simulations based on this model proves the potential of the proposed NEMSIP in image applications.

### 4.1 POSITIVE FILTERS - GAUSSIAN FILTER

Gaussian filter is a commonly used basic method to reduce noise. When implemented with NEMSIP, the sensor output voltages are connected to the back layer of NEMS, and the $A_{ov}$ ratio is another multiplier, as shown in Figure 7. The simulation results are shown in Figure 8. As shown in Figure 8(c), it is comparable to the Matlab result in Figure 8(b). Figure 8(i) is the result of the high-speed (on-to-off) mode. The quality in the high speed mode is similar to that of the low-speed (off-to-on) mode. The high-speed mode is very fast because the pump switching distance to break is less than 1nm distance. Just a little gap can turn off the connection.

### 4.2 POSITIVE AND NEGATIVE FILTERS - EDGE DETECTION

By changing the Ratio $A_{ov}$, an edge detection function could be implemented with NEMSIP. Various types of input are implemented and shown in Figure 9, including "Text" in Figure 9(a), "Cartoon" in Figure 9(d), "Dice" with different levels of distance and details in Figure 9(g), "Lenna" in Figure 9(j), "Car license" in Figure 9(m). Results show that the NEMSIP implementation could detect the edges in Figure 9(c, f, i, l, o), but the effect is weaker than the Matlab canny algorithm in Figure 9(b, e, h, k, n). These differences originate from the algorithm differences, not the implementation.

## 4.3 REDUCING MULTIPLE FILTERS - EDGE DETECTION

In edge detection, Gaussian filtering is useful to suppress the noise. The $A_{vo}$ could also be tuned to include Gaussian filter into edge detection. Figure 10 shows the edge detection results using NEMSIP. Compared with Figure 9(o), Figure 10 gives preferred results. It is also observed that the NEMSIP could realize more complicated edge detection kernels through combining more than one path into one single NEMSIP processing step.

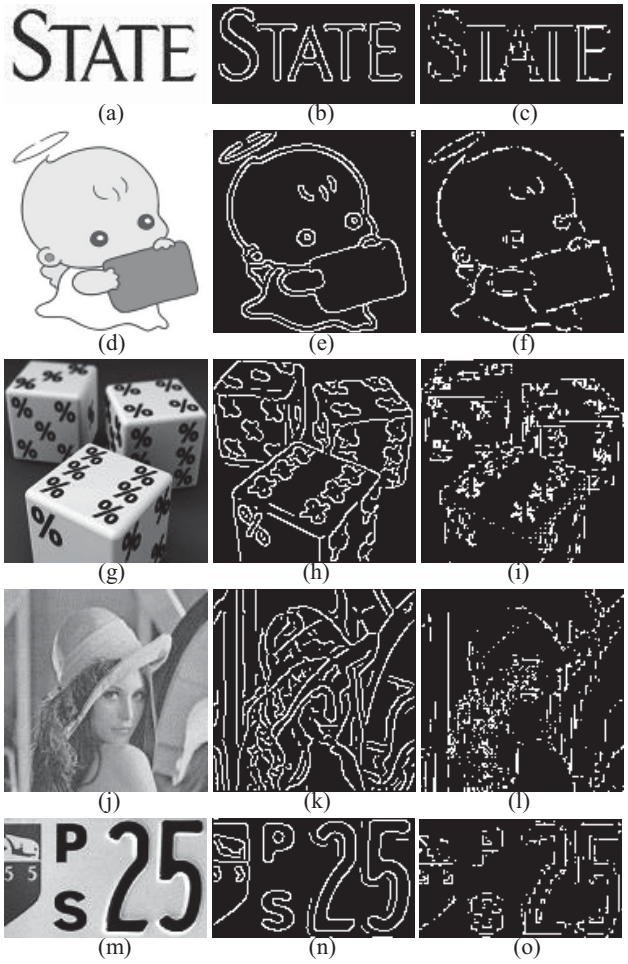



**Figure 9. Edge detection results comparison.** (a) Text image from sensor, as inputs to NEMSIP. (b) Matlab canny edge detection result. (c) NEMSIP edge detection result. (d) Cartoon inputs. (e) Matlab canny edge detection for cartoon in (d). (f) NEMSIP outputs for cartoon. (g) dice inputs. (h) Matlab canny edge detection for dice. (i) NEMSIP outputs for dice. (j) Lenna 128*128 inputs. (k) Matlab canny edge detection for Lenna. (l) NEMSIP outputs for Lenna. (m) Car license inputs. (n) Matlab canny edge detection result. (o) NEMSIP outputs for Car license.

## 4.4 MAPPING LARGE WINDOW IN NEMSIP FOR MOTION ESTIMATION

The widely-used motion estimation method calculates the sum of the absolute difference (SAD) block by block to search for the best matched block to reduce the difference between the source and the target [16]. NEMSIP is able to achieve the same quality with the full search algorithm with a delay in the order of nanoseconds. Figure 11(a, b, c) shows the JPEG inner frame motion estimation with the Pipelined NESMIP. Figure 11(a) is the input to NEMSIP, and the results in Figure 11(b, c) show that NEMSIP is able to find the best matched block. Figure 11(d, e) shows two frames from the testbench "football", in which the human and ball are moving fast, and the NEMSIP can still find the best matched block as shown in Figure 11(f).

**Figure 10. Edge detection with Gaussian filter.**

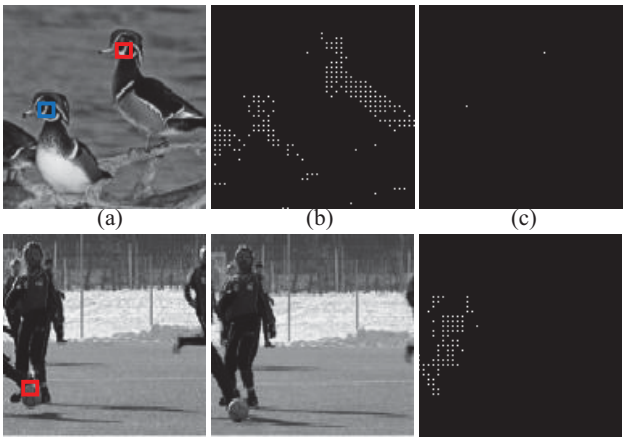



**Figure 11. Motion estimation results for Figure 6.** (a) Motion estimation input. (b) Results to identify the red block at a certain low gate voltage threshold. (c) By scanning the gate voltage from high to low, the best matched point is obtained (as well as the motion vector). (d, e) Inter frame motion estimation for H.264 video compression algorithm with the red block in (d) as the source block. (f) NEMSIP motion estimation results.

## 5. SPEED, POWER, and 3D INTEGRATION

In this section, NEMSIP is compared with traditional CMOS solutions in terms of speed and power. With the near-sensor processing by NEMSIP, the transmission bandwidth between the sensor chip and the digital signal processor could be reduced significantly after pre-processing by NEMSIP.

## 5.1 SPEED EVALUATION

The speed of NEMSIP is determined by the NEMS turn-on/off time $t_{mesh}$ and $t_{elec}$. The time required to process one frame, T, could be obtained based on Figure 12(a):

$$T=(2^8+1)*(t_{adder} + t_{DAC} + t_{mesh} + t_{elec} + t_{flip\text{-}flop}), \quad (10)$$

where $t_{adder}$ $t_{DAC}$, and $t_{flip\text{-}flop}$ are the delay of the adder, the DAC, and the flip-flop, respectively. By applying the high-speed (on-to-off) mode, the $t_{mesh}$ could be reduced significantly. The reset time could be optimized through the Ping-Pong loop shown in Figure 12(b). In the Ping loop, the adder increases from 0 to 255; while in the Pong loop the substractor decreases from 255 to 0.

Table 2 shows the speed comparison between NEMSIP and CMOS solutions. The NEMSIP solution is 7.6 times faster than the pipelined CMOS structure. For the scaled model, NEMSIP is 16 times faster than CMOS.

Furthermore, for NEMSIP, the speed is independent on the image scale, while the pipelined CMOS solution highly depends on the image scale. For NEMSIP, the image processing does not require

additional CMOS-based ADC which, in contrast, is inevitable for the CMOS solution. In this test, the entire image is stored in the memory, and the read delay from the memory is not included.

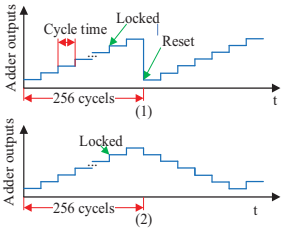



**Figure 12. Timing strategies. (1) Reset solution. (2) Ping-Pong.**

**Table 2. Speed Comparison between NEMSIP and CMOS**

| Type | Speed (frame/second) |
|---|---|
| Current NEMSIP | 233k |
| NEMSIP scaled to 90nm | 488k |
| 32nm CMOS [a] | 30.5k |

[a]: The CMOS is an optimized pipelined structure with one clock cycle per pixel output, 32nm process in tt corner, standard $V_t$, 1.05 V power supply, and 500MHz clock.

## 5.2 POWER EVALUATION

The power in NEMSIP is mainly consumed by the following parts: the adder, flip-flops, and charging and discharging of the NEMS capacitor. The NEMS could be treated as a panel capacitor. Each computation charges or discharges the capacitor once. For the current NEMS device, the $V_{pi}$ is measured to be 8~10V, which is lowered to 40mV after scaling to 90nm. It is observed that the NEMSIP power estimation in this paper is significantly different from [2], in which NEMS was used to function similarly to traditional transistors switches.

Table 3 shows the power consumption comparison between NEMSIP and CMOS solutions in Table 2. The NEMSIP results are calculated, and 32nm CMOS results are from RTL level with Design Compiler. For the same frame rate, the 8-bit adder and the 128*128*8 flip-flops operates at 7.8MHz. The NEMSIP solution using the existing NEMS consumes only 5.9% of the power by the CMOS solution for the same frame rate.

**Table 3. Power Comparison between NEMSIP and CMOS [a]**

| Type | Power |
|---|---|
| Current NEMSIP | 0.2626mW for 128*128 NEMS + 0.286mW for 128*128*8 flip-flops + 2.9uW for adder |
| 32nm CMOS | 9.325mW total power |

[a]: The NEMSIP is operating at the same frame frequency of 30.5k frame/second. The 32nm CMOS solution area is 0.45mm$^2$.

## 5.3 3D INTEGRATION

When integrated with the CCD or CMOS sensor layer on the top and the CMOS digital processor on the bottom, the sandwiched NEMSIP layer achieves near-sensor image pre-processing. Between every two adjacent layers, an insulator layer is used for isolation. Similar to the integration in [3], through-silicon via (TSV) is not necessary, and the top metal layer interconnection is sufficient for such integration with high interconnection density.

# 6. CONCLUSION

This paper uses NEMS as weighted multiple-input energy-efficient comparators to implement parallel ADCs for near-sensor image processing. The proposed NEMSIP solution has almost no leakage by the NEMS, and no conventional ADCs are required. The current NEMS and scaled NEMS could operate at a speed of 16 times faster than the pipelined CMOS solution, while consuming only 5.9% of the CMOS power even if the ADC power of the CMOS solution is not included. Furthermore, the bandwidth of the near-sensor NEMSIP processing could also be significantly reduced with less transmitted data. In general, the NEMSIP is promising in bringing energy and performance benefits in future image processing.

# 7. ACKNOWLEDGEMENTS

This work was supported in part by the LEAST, one of the six SRC STARnet Centers, sponsored by MARCO and DARPA. This work was also supported in part by NSF Expeditions in Computing Award 1317560.

# 8. REFERENCES

[1] Jaeseok Jeon et al. Multiple-Input Relay Design for More Compact Implementation of Digital Logic Circuits. Electron Device Letters. 2012.

[2] Spencer, M. et al. Demonstration of Integrated Micro-Electro-Mechanical Relay Circuits for VLSI Applications. JSSC, vol.46, no.1, pp.308,320. 2011

[3] V. Pott et al. Mechanical Computing Redux: Relays for Integrated Circuit Applications. Proceedings IEEE, vol.98, no.12, pp.2076,2094.2010

[4] J. Daesung et al. Combinational Logic Design Using Six-Terminal NEM Relays. CADICS, vol.32, no.5, pp.653,666. 2013

[5] C. Soogine et al. Nanoelectromechanical (NEM) relays integrated with CMOS SRAM for improved stability and low leakage. ICCAD. 2009

[6] S. Yong-Ha et al. High-performance MEMS relay using a stacked-electrode structure and a levering and torsional spring for power applications. Micro Electro Mechanical Systems (MEMS). 2012

[7] B. Pruvost et al. 3-D Design and Analysis of Functional NEMS-gate MOSFETs and SETs. IEEE Transactions on Nanotechnology. 2007

[8] Alaghi, A. et al. Stochastic circuits for real-time image-processing applications. Design Automation Conference. pp.1,6. 2013

[9] Elouardi, A. et al. A Smart Sensor for Image Processing: Towards a System on Chip. ISIE, vol.4, no., pp.2857,2862.

[10] X. Zhang et al. A second generation 3D integrated feature-extracting image sensor. IEEE Sensors, pp.1933,1936.

[11] M. Sarkar, Biologically Inspired CMOS Image Sensor for Fast Motion and Polarization Detection. IEEESJ, vol.13, no.3, pp.1065,1073.

[12] Y. Chin et al. A 0.5V 34.4uW 14.28kfps 105dB smart image sensor with array-level analog signal processing," SSCC, pp.97,100, 11-13 Nov. 2013

[13] K. Kiyoyama et al 2013. A block-parallel ADC with digital noise cancelling for 3-D stacked CMOS image sensor. 3D Systems Integration Conference (3DIC), 2013 IEEE International, pp.1,4.

[14] Lie, D. et al. Analysis of the Performance, Power, and Noise Characteristics of a CMOS Image Sensor With 3-D Integrated Image Compression Unit. CPMT, vol.4, no.2, pp.198,208. 2014

[15] F.D. Oliveira et al. CMOS Imager With Focal-Plane Analog Image Compression Combining DPCM and VQ. Circuits and Systems I: Regular Papers, IEEE Transactions on, vol.60, no.5, pp.1331,1344. 2013

[16] K. Ma et al. A Novel Multi-direction Fast Parallel Search (MFPS) Method for Motion Estimation in Video Compression and Its Hardware Implementation. The IEEE 10th International Conference on ASIC, 2013.

[17] L. Areekath et al 2013. Sensor assisted Motion Estimation. Engineering and Systems (SCES), pp.1,6, 12-14 .

[18] Chen, X. et al 2011. Sensor-Assisted Video Encoding for Mobile Devices in Real-World Environments. IEEE Transactions on Circuits and Systems for Video Technology, vol.21, no.3, pp.335,349.

[19] L. Kang-Wook Lee et al 2013. Die-Level 3-D Integration Technology for Rapid Prototyping of High-Performance Multifunctionality Hetero-Integrated Systems. Electron Devices,vol.60, no.11, pp.3842-8.

[20] X. Cui et al, Research on power model of multi-mode FinFET standard cell, ICSICT, pp.1,3, Oct. 29 2012-Nov. 1 2012

[21] K. Ma et al, Key characterization factors of accurate power modeling for FinFET circuits. Science China Information Sciences, 2015, 58(2): 1-13.

[22] K. Ma et al, Independently-Controlled-Gate FinFET 6T SRAM Cell Design for Leakage Current Reduction and Enhanced Read Access Speed, ISVLSI, pp.296,301, 9-11 July 2014