Efficient Routing in Coarse-Grained Reconfigurable Arrays Using Multi-Pole NEM Relays

Akash Levy*
Dept. of Electrical Engineering
Stanford University
Stanford, CA
akashl@stanford.edu

Michael Oduoza*
Dept. of Electrical Engineering
Stanford University
Stanford, CA
mcoduoza@stanford.edu

Akhilesh Balasingam
Dept. of Electrical Engineering
Stanford University
Stanford, CA
avb03@stanford.edu

Roger T. Howe
Dept. of Electrical Engineering
Stanford University
Stanford, CA
rthowe@stanford.edu

Priyanka Raina
Dept. of Electrical Engineering
Stanford University
Stanford, CA
praina@stanford.edu

***Abstract*—In this paper, we propose the use of multi-pole nanoelectromechanical (NEM) relays for routing multi-bit signals within a coarse-grained reconfigurable array (CGRA). We describe a CMOS-compatible multi-pole relay design that can be integrated in 3-D and improves area utilization by 40% over a prior design. Additionally, we demonstrate a method for placing multiple contacts on a relay that can reduce contact resistance variation by 40× over a circular placement strategy. We then show a methodology for integrating these relays into an industry-standard digital design flow. Using our multi-pole relay design, we perform post-layout simulation of a processing element (PE) tile within a hybrid CMOS-NEMS CGRA in 40 nm technology. We achieve up to 19% lower area and 10% lower power at iso-delay, compared to a CMOS-only PE tile. The results show a way to bridge the performance gap between programmable logic devices (such as CGRAs) and application-specific integrated circuits using NEMS technology.**

***Index Terms*—NEM relay, nanoelectromechanical relay, NEMS, CGRA, coarse-grained reconfigurable array**

I. Introduction

Coarse-grained reconfigurable arrays (CGRAs) have been gaining popularity as specialized programmable logic device (PLD) architectures for applications such as image processing and machine learning [1] [2] [3]. CGRAs are similar to field-programmable gate arrays (FPGAs), but improve power, performance, and area (PPA) metrics for applications through the use of coarse-grained datapaths which: (1) route multiple bits together using multi-bit routing switches, and (2) replace fine-grained look-up tables (LUTs, which perform arbitrary Boolean operations) with processing elements (PEs, which perform common arithmetic operations, e.g. addition, multiplication). It was estimated in [3] that a CGRA can provide 1.4× better energy-efficiency and 3.1× better compute density than an FPGA. However, providing general programmability is still costly: compared to an application-specific integrated circuit (ASIC) solution, a CGRA was estimated to have 6-10× worse energy and area efficiency. This wide gap between the performance of PLDs and ASICs motivates the need for techniques to reduce the reconfigurability overhead in PLDs.

Akash Levy is partially funded by the NSF Graduate Research Fellowship Program. *These authors contributed equally to this work.

Nanoelectromechanical (NEM) relays are nano-scale mechanical switches that can be electrostatically actuated with a gate. Their properties have been widely studied, and they have been proposed as alternatives/complements to CMOS logic for improving PPA in digital circuits [4]. NEM relays have zero static power dissipation and low ON-state resistance (∼1-10 kΩ experimentally, comparable to that of modern NMOS transistors), but they switch much more slowly than NMOS transistors (nanoseconds to microseconds, rather than picoseconds) [4]. However, when used as statically configured routing switches, the long switching delay usually does not impact PPA negatively, since they only need to be toggled once, during initialization (programming) of the PLD. NEM relays may also have multiple poles (we refer to such relays as *multi-pole NEM relays*), which allow multiple signals to be switched by the same gate. Previous work has analyzed the potential benefits of integrating *single-pole* NEM relays into FPGAs as static routing switches and configuration memory and found significant opportunities for improving PPA across several applications [5] [6]. However, these studies assumed that multiple layers of relays can be monolithically integrated with CMOS, which has not been demonstrated experimentally and would not be cost-effective.

In this paper, we show for the first time that integration of *multi-pole* NEM relays into a CGRA design improves PPA, thereby reducing reconfigurability overhead. The major contributions of this paper are:

1) A multi-pole NEM relay design integrated on top of CMOS back-end-of-line (BEOL) circuitry, featuring *a single layer* of relays which incorporate anchor sharing, folded beams, and tessellation to improve area utilization by 40% over a prior design reported in [7].
2) Iterative finite element modeling (FEM) to optimize multi-pole contact placement, reducing expected contact resistance variation by $> 40\times$ over a circular placement.
3) Methodology to integrate multi-pole NEM relays into a standard digital design flow.
4) Design of a processing element (PE) tile in a hybrid CMOS-NEMS CGRA that integrates multi-pole NEM

978-1-6654-2135-5/22/$31.00 ©2022 IEEE

relays, achieving 19% lower area and 10% lower power at iso-delay.

II. NEM Relay Background

In this work, we focus on planar vertically-actuated NEM relays [8]. These relays can be fabricated at relatively low temperatures ($< 450°C$) and use "clean" materials commonly available in foundries today (e.g. poly-SiGe, Al_2O_3, W), making them compatible with current CMOS back-end-of-line (BEOL) technology. Additionally, these relays have low operating voltage and low contact resistance—both desirable properties for integration with silicon CMOS circuits. We examine the case where relays are laid out in a single layer on top of a CMOS circuit. Having a single layer of relays enables cost-effective fabrication, reducing the number of masks required and precluding the need to encapsulate the relays for further processing. More details on fabrication steps for this type of relay can be found in [8] and [9].

Fig. 1 shows a 4-terminal (4T) NEM relay (single-pole). Its four terminals (*source*, *drain*, *body*, *gate*) have similar functions as the four terminals of a MOSFET but make use of different conduction/switching mechanisms. The electric potential between the gate (G) and the body (B) (V_{GB}) results in electrostatic attraction between the relay and the gate. When V_{GB} exceeds a critical value, called the pull-in voltage (V_{pi}), the elastic force of the relay can no longer balance the electrostatic force, and the relay body collapses toward the gate. Once this happens, the *conductive channel* touches down on the source (S) and drain (D). This state is referred to as the *ON state* (Fig. 1b), and the source and drain are electrically connected, enabling current flow (I_{DS}) between them. As V_{GB} is decreased, the channel disconnects from the drain at another critical voltage, called the pull-out voltage (V_{po}), which is smaller than V_{pi} due to electromechanical instability and stiction. When the channel is disconnected, the relay is in the *OFF state* (Fig. 1a), and no current can flow between source and drain. Since $V_{po} < V_{pi}$, a value of $V_{GB} = V_{hold}$ where $V_{po} < V_{hold} < V_{pi}$ allows the relay to retain its state (hysteresis) (Fig. 1d). Fig. 1c shows the top view of a NEM relay and Fig. 1d shows its typical I_{DS}-V_{GB} characteristic.

NEM relays may also have multiple poles that connect independent source-drain pairs (Fig. 1e). Working demonstrations of multi-pole relays have been reported [10], but strategies to make use of multiple poles at a system level have not been studied in detail, to our knowledge.

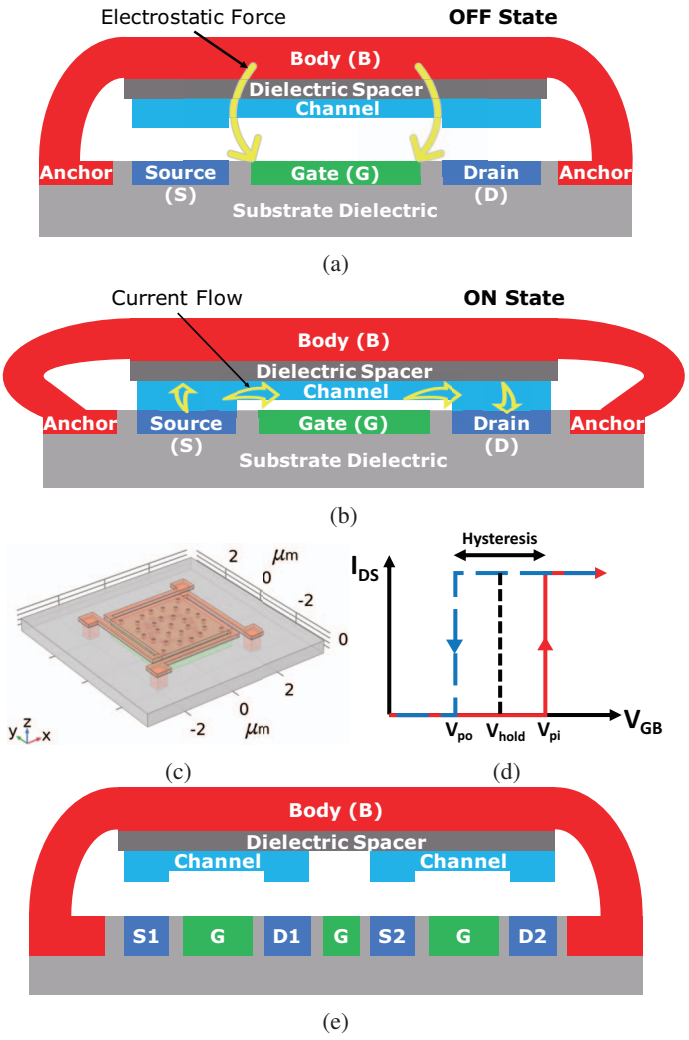

Fig. 1: Illustration of NEM relay side view in (a) OFF state, and (b) ON state. The electric potential between body and gate controls current flow from source to drain. (c) Top view of a NEM relay. The relay moves up and down in the z-direction when a voltage is applied to its gate. (d) I_{DS} vs. V_{GB} for a NEM relay. (e) Multi-pole NEM relay side view. The D, S, G, B terminals correspond to drain, source, gate, and body, respectively. Note that there are multiple source-drain pairs, but only one gate.

III. Multi-Pole NEM Relay Design and Modeling

A. Relay Layout

We first develop a parametric NEM relay layout, shown in Fig. 2a, that maximizes the relative size of the relay body (for larger electrostatic force) while maintaining a low enough spring constant to enable CMOS-compatible pull-in/pull-out voltages ($< 5\,V$). Parameters include the various lateral and thickness dimensions of the relay, given in Table I.

The relay consists of four cantilever *beams* that connect to the square-shaped *electrostatic plate*. The electrostatic plate generates most of the force needed to snap the relay down into the ON state (shown in Fig. 2d). The purpose of the beams is to provide a sufficient spring force to restore the relay back to the OFF-state (Fig. 2c) when $V_{GB} < V_{po}$. There are also release holes, which allow the sacrificial material under the relay to be removed during fabrication. These holes also allow gas under the relay to escape and hence affect the relay damping behavior—in our design, these holes are placed in concentric circles. The relay layout is designed to be invariant under $90°$ rotations in the xy-plane.

To maximize area efficiency of our relays while still using CMOS-compatible voltages ($<5\,V$), we use the following techniques: (1) We share anchors and tessellate the relays in a mosaic structure, as shown in Fig. 2b. This reduces the relay pitch and minimizes unused area between relays. (2) We use folded beams to reduce the relay footprint. We define *area utilization* as the ratio of the electrostatic plate

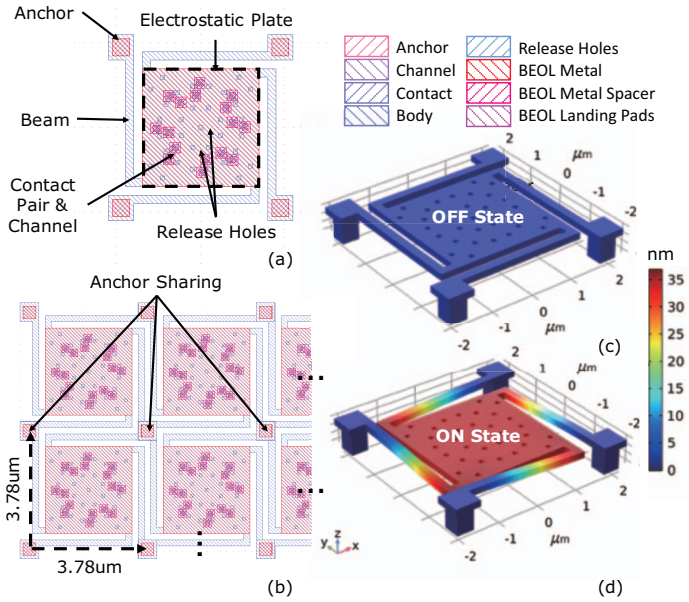

Fig. 2: Relay layout. (a) Top view layout of a single relay, showing the different components and layers. (b) Mosaic of relays, illustrating how anchors may be shared across relays. (c) Top view of a NEM relay in the OFF state, and (d) the ON state. The color indicates the downwards z-displacement of the relay.

TABLE I: Important NEM relay dimensions.

Lateral Dimensions (x-y plane)	
Length of electrostatic plate (square)	2780 nm
Side length of contact (square)	150 nm
Side length of release hole (square)	100 nm
Side length of anchor (square)	400 nm
Width of cantilever beam	200 nm
Gap between landing pad and body	200 nm
Gap between cantilever beam and body	200 nm
Thickness Dimensions (z-direction)	
Thickness of electrostatic plate	120 nm
Thickness of contact	25 nm
Thickness of dielectric spacer	30 nm
Thickness of conductive channel	10 nm
Gap between relay plate and substrate	60 nm

TABLE II: CAD-extracted NEM relay properties.

Param	Value	Description
C_{GB}	1.4 fF	Gate-to-body cap (OFF)
C_{CG}	7.4 aF	Gate-to-channel cap (OFF)
C_{DG}	0.07 fF	Gate-to-drain/source cap (OFF)
C_{CB}	0.15 fF	Body-to-channel cap (OFF)
C_{DB}	1.1 aF	Body-to-drain/source cap (OFF)
C_{DC}	< 1 aF	Chan-to-drain/source cap (OFF)
C_{GB}	2.5 fF	Gate-to-body cap (ON)
C_{CG}	17.3 aF	Gate-to-channel cap (ON)
C_{DG}	0.07 fF	Gate-to-drain/source cap (ON)
C_{CB}	0.15 fF	Body-to-channel cap (ON)
C_{DB}	1.6 aF	Body-to-drain/source cap (ON)
R_{DS}	80 Ω	Total source-drain resistance (ON)

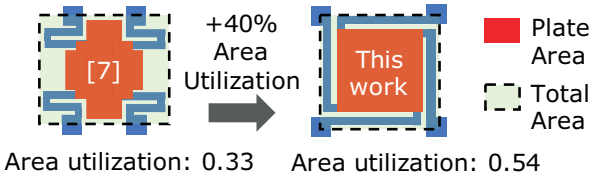

Fig. 3: Area utilization of the NEM relay in [7] (left) versus this work (right).

area to the total relay area. Under this definition, the area utilization of our relay (shown in Fig. 3, right) is 0.54. For comparison, the NEM relay design from [7] (shown in Fig. 3, left) using a folded flexure (and the same NEMS-on-CMOS integration scheme as this paper) has an area utilization of roughly 0.33 (assuming the top anchors are shared across relays in that design as well). The simple layout used in this work allows reduction of the effective footprint by about 40% over the alternative design in [7], while maintaining a reasonable spring constant—critical to our goal of realizing area-efficient NEMS-on-CMOS routing.

B. Finite Element Modeling (FEM)

We model the relay using an accurate COMSOL 3-D finite element model (FEM) in order to extract key parameters, such as the parasitic resistances/capacitances between different relay terminals. These parameters are given in Table II. We also extract an estimate for the *contact forces* between the channel contacts and the sources/drains which they touch down upon. These contact forces can be used to predict the expected contact resistance (R_c). The contact resistance determines the source-drain resistance for the NEM relay ($R_{DS} = 2 \times R_c$).

FEM demonstrates pull-in/pull-out voltages of 4.39 V and 3.43 V, which are within 5% of those predicted by an approximate analytical model based on ideal parallel plates [8]. V_{op} is the gate-to-body voltage (V_{GB}) applied to a relay when we want it to go to the ON state (pull-in operation). In order for pull-in to be possible, we need $V_{GB} > V_{pi}$, so 4.5 V is a reasonable V_{op} choice for our relay. However, this is much higher than the core voltage of most commercial 40 nm technologies. To address this problem, we employ a body biasing strategy, where the relay's body terminal is tied to a negative voltage supplied from off-chip [11]. With a body bias of -3.4 V for our relay, an applied gate voltage of 0 V will result in $V_{GB} = 3.4$ V (OFF-state), while a gate voltage of $V_{DD} = 1.1$ V gives $V_{GB} = V_G - V_B = 1.1\,\text{V} - (-3.4\,\text{V}) = 4.5$ V (ON-state). This means that a full-swing 1.1V CMOS signal at the gate terminal can enable both pull-in and pull-out of the relay.

C. One-Hot Multiplexer (OHMux) Standard Cells

We compose our multi-pole NEM relays into *one-hot multiplexer* (OHMux) standard cells. An OHMux has N selection bits to select from N input signals, with only one selection bit high at a time. To compose an N-input OHMux with NEM relays, we connect N relays by: (1) shorting all of their drains, (2) connecting each source to one of the input signals, and (3) connecting each gate to one of the N selection bits. With multi-pole relays, OHMuxes can multiplex multi-bit inputs. An implementation of an 8-bit-wide 4-input OHmux is shown in Fig. 4.

We design 3D standard cell layouts for {2,4,10}-input 8-bit-wide OHMuxes used in different routing components of

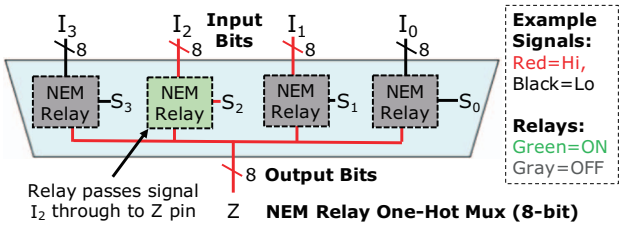

Fig. 4: 4-input 8-bit one-hot multiplexer. Four 8-bit input signals (I_i) are selected by four selection bits (S_i). Only one of the selection bits is hot (S_2 in the example shown).

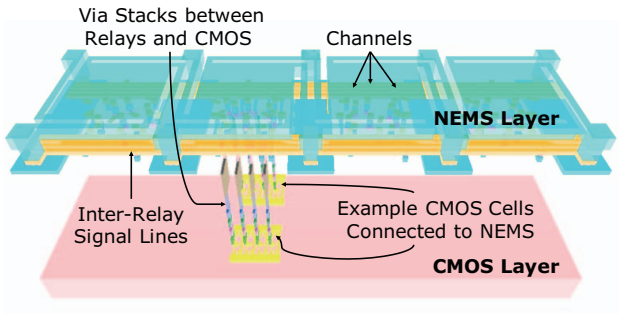

Fig. 5: 3-D view of 4-input 8-bit OHMux layout, rendered with GDS3D. showing the NEMS-on-CMOS integration scheme and the standard cell design.

the CGRA (discussed in Section V). The 4-input 8-bit-wide OHMux layout is given in Fig. 5, showing vias routing down to a few CMOS standard cells as an example. All of our NEM OHMux layouts have zero design rule errors in the 40 nm technology we use. Inter-relay signal lines short relays' drains (corresponding to the same output bit) to each other. Using parasitic extraction, we conservatively estimate the capacitance of inter-relay signal lines to be $C_{sigline,ohmux} = 0.6\,\text{fF}$.

IV. Contact Placement for Multi-Pole Relays

Correct placement of NEM relay contacts is critical for multi-pole relay operation. If contacts are placed naively, some of them may not make good connection with the source/drain, resulting in high and/or variable contact resistance, which can result in reliability problems and increased power/delay.

To ensure that a multi-pole relay works reliably, its contacts must all touch down simultaneously during the pull-in operation. Therefore, the contacts must be located along a *displacement contour* of the relay. For complex NEM relay designs, it is difficult to analytically find displacement contours. We propose *iterative contact placement (ICP)* for placing contacts along a displacement contour. Starting with a circular placement, the method works by iteratively finding a displacement contour via FEM, extracting its *connected components* (contiguous sections of the displacement contour), and placing contacts along these components. This process is repeated until the contact forces roughly equalize, resulting in uniform contact resistance (Fig. 6).

We estimate contact resistance based on the *effective contact area* model [12] [13]. Under this model, ICP yields a contact resistance of around $40\,\Omega$ for each contact with a standard deviation of $< 1\,\Omega$ across contacts. This contact resistance

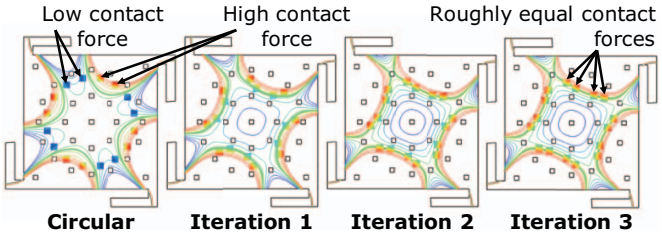

Fig. 6: Iterative contact placement. Contacts are initialized in a circle and displacement contours are extracted using FEM. In each iteration, contacts are placed along the displacement contours from the prior iteration. This is repeated until the contact forces are roughly equalized.

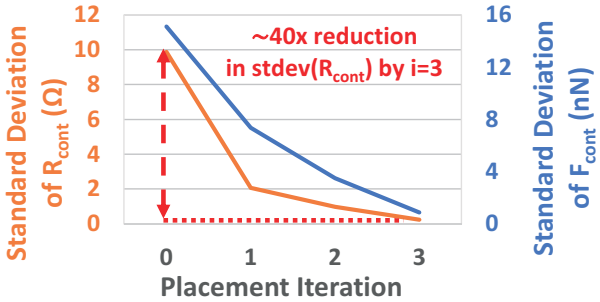

Fig. 7: Reduction in standard deviation of contact force (F_{cont}) and expected contact resistance (R_{cont}), obtained by using iterative contact placement for an 8-bit relay (with 16 poles).

estimate is lower than that reported in the literature for contacts of similar type [10]. This difference may be explained by the fact that our contact forces are much larger, and real contacts include non-idealities in processing conditions, such as oxidation of electrodes or contamination. Regardless, the reduction in contact force variation should improve uniformity across contacts, thereby improving reliability. The predicted contact resistance variation under the model drops by about $40\times$ in just 3 placement iterations (Fig. 7).

V. Hybrid CMOS-NEMS CGRA Design

We use an existing open-source CGRA [2] to evaluate our proposed use of multi-pole NEM relays. It has an island-style architecture with *processing element (PE)* tiles, *memory (MEM)* tiles, and a configurable interconnect (Fig. 8). PE tiles perform 16-bit arithmetic operations. MEM tiles contain SRAMs that can be used as scratchpad memory. The interconnect contains horizontal and vertical 16-bit routing tracks. Switch boxes (SBs) implement connections between any two tiles. Each SB receives 5 16-bit input tracks and drives 5 16-bit output tracks on each side (north, south, east, west). Each output track is driven by a multiplexer that selects among the PE/MEM output and the routing tracks coming from the three other sides of the SB (Fig. 8). Each output can also be selectively pipelined. Each connection box (CB) selects a PE/MEM input from among 10 16-bit routing tracks (Fig. 8).

The CMOS multiplexers are obtained using logic synthesis, which results in an AND-OR-INV (AOI) implementation. To create our hybrid CMOS-NEMS PE tile, we replace all the CMOS multiplexers in the SBs and CBs with NEM OHMuxs

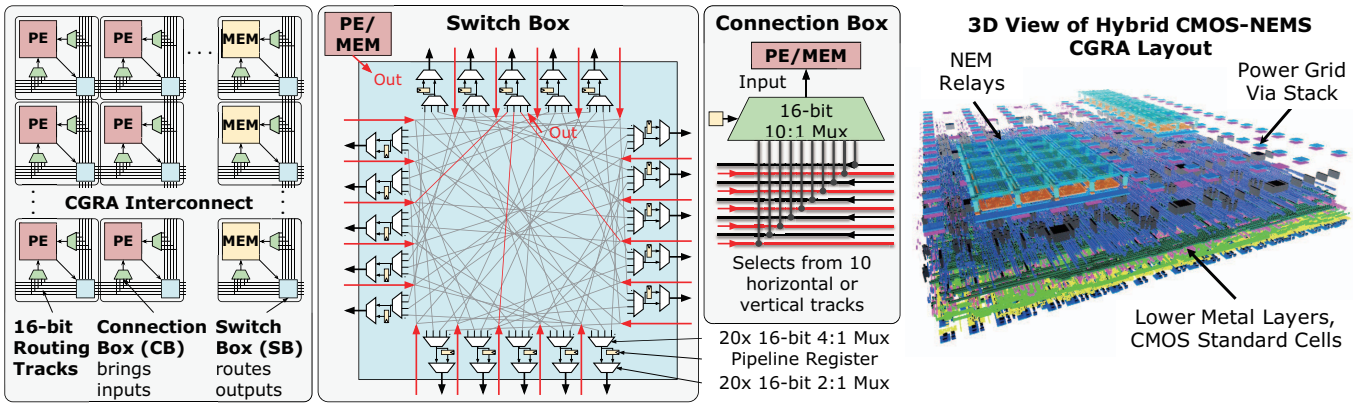

Fig. 8: Left: CGRA with processing element (PE) and memory (MEM) tiles and a configurable interconnect. Also shown are the multiplexers in the switch box (SB) and connection box (CB) that we replace with NEMS-based variants. Right: A 3-D view of a portion of the place-and-routed hybrid CMOS-NEMS CGRA PE tile, displaying several NEM muxes and the metal stack below them.

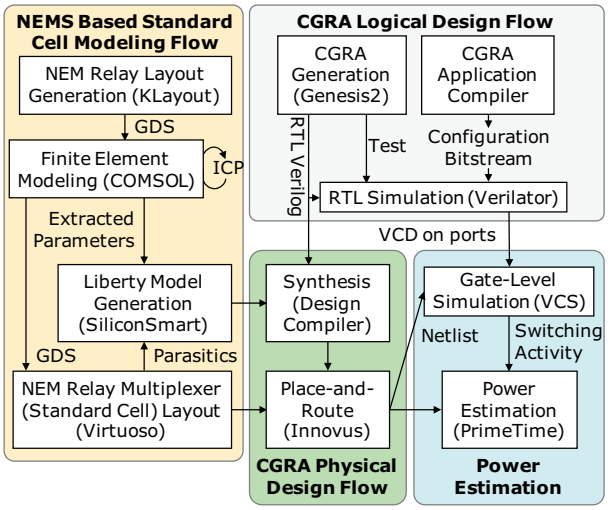

Fig. 9: Start-to-finish hybrid CMOS-NEMS CGRA flow.

(Section III-C). We add CMOS decoders to the outputs of the configuration registers to create one-hot signals that drive the NEM OHMux selection bits. Decoders add a small area overhead of $< 100\,\mu m^2$ per tile ($\sim$1% of the PE tile area).

A. Digital Design Flow with NEM Relays

Adding the NEM OHMuxes in physical design requires several modifications to the standard place-and-route (P&R) flow. Firstly, the OHMuxes must be declared as *cover cells*, to allow CMOS cells to be placed below them. Additionally, power grids must be constructed to supply the body bias to the NEM relays for proper operation. We create a custom legalization script to align the NEM OHMux cells with the power grids while ensuring no overlap between relays (besides legal anchor sharing). Place-and-route is performed using Cadence Innovus. For fairness of comparison, in both NEMS and CMOS designs, we target the same placement density and clock period and use the same design constraints. The full CMOS-NEMS design flow is shown in Fig. 9, and a 3-D view of part of the resulting 3-D PE tile layout is shown on the right in Fig. 8.

B. Power and Delay Analysis

A major challenge in integrating our NEM relays with the rest of the CMOS digital design flow is that without output buffering (i.e., using a CMOS gate to restore the signal at the output of the multiplexer), each OHMux functions similar to a pass gate—it does not directly drive its outputs, but rather, exposes its downstream capacitance to the driving cell that precedes it. Pass gates are not well supported by commercial EDA tools. To produce correct timing and power estimates with standard EDA tools, we employ the following method.

The capacitive load seen by the driving cell of a NEM relay input pin is different depending on whether the relay is OFF or ON. When it is OFF, the capacitance seen by the driver of the source pin is given in Eqn. 1 (the capitalized subscript letters correspond to the terminals of the relay in Fig. 1a).

$$C_{in,relay,off} \approx \underbrace{C_{DB,off} + C_{DG,off} + C_{DC,off}}_{\text{total source cap}} \approx 0.07\,\text{fF} \tag{1}$$

When the relay is ON, this effective capacitance increases, as the relay must also charge its channel capacitance and drain capacitance, as well as the downstream load:

$$C_{in,relay,on} \approx \underbrace{C_{CB,on} + C_{CG,on}}_{\text{total channel cap}} + \underbrace{2(C_{DB,on} + C_{DG,on})}_{\text{total source+drain cap}} + \underbrace{C_{load,relay}}_{\text{downstream cap}} \tag{2}$$

The above equations are for a single NEM relay. For an N-input NEM OHMux, the input pin capacitance is the same as the single-relay case when the input is unselected.

$$C_{in,ohmux,off} = C_{in,relay,off} \approx 0.07\,\text{fF} \tag{3}$$

However, when the input pin is selected (i.e. its corresponding select pin is high), two more components are added to the effective pin capacitance: (1) the drain terminals of the other $N-1$ relays, and (2) the wire capacitance of the signal line connecting the outputs of the NEM relays together. The

resulting pin capacitance of a selected pin in an N-input NEM OHMux is given in Eqn. 4.

$$C_{in,ohmux,on} = C_{in,relay,on} + \underbrace{(N-1)(C_{DB,on} + C_{DG,on})}_{\text{drain cap of other NEM relays}} + \underbrace{C_{sigline,ohmux}}_{\text{output sig line cap}} \quad (4)$$

To create an EDA flow with NEM relays, we first need to make valid Liberty files for the NEM OHMuxes. Liberty files contain lookup tables, which (1) map input transition times and output load capacitances to power/delay estimates, and (2) determine the output transition time to allow the next cell to determine its own power/delay. However, since the NEM relays behave as pass gates, we model them as not having any internal power/delay of their own—rather, they increase the power/delay of the design through the capacitance/resistance they show to their input pin drivers. Therefore, for the NEM OHMuxes, we set all the entries in the power/delay lookup tables to zero. At the same time, we configure the table for the output transition time to simply forward the transition time given at the input pin directly to the output pin. Then we manually adjust the load capacitance seen by the driving cells—specifically, we update the input pin capacitances of the NEM OHMuxes in the Liberty file. We also incorporate the NEM relay drain-to-source resistance by adding R_{DS} onto each of the wires driving the relay inputs.

For logic synthesis and P&R, we provide a Liberty file containing the *worst-case* estimates for the pin capacitances by setting $C_{in,ohmux} = C_{in,ohmux,on}$ at all input pins, with output load estimates taken from the cells in the CMOS-only PE tile. This enables the P&R tool to conservatively size gates to meet timing, despite the fact that the NEM OHMuxes appear to have zero delay of their own. The same worst-case Liberty file is used for signoff timing analysis.

Power analysis is performed for each active PE tile in the CGRA and then averaged. We start with *best-case* estimates for the OHMux input pin capacitances in the Liberty file by setting $C_{in,ohmux} = C_{in,ohmux,off}$. Then we set the pin capacitances of only the selected OHMux input pins to $C_{in,ohmux,on}$, since these are configured to pass their signal through. We also move the extracted downstream parasitics from the output net of the NEM OHMuxes to the input net, so they are appropriately seen by the drivers. Finally, we run power analysis and zero out the contribution from the NEM OHMuxes that arises due to the charging of its output load (this power would otherwise be double-counted, since we have already included it at the OHMux's input driver).

VI. Evaluation Methodology

We evaluate the PPA improvement from NEM multiplexers for several applications on the CGRA. To compile applications to the CGRA, we use the toolchain from [2], which is similar to a high-level synthesis toolchain for a commercial FPGA. It takes as input an application written in a high-level domain-specific language called Halide [14], compiles it to a dataflow graph, and then maps, places and routes the graph on the CGRA. The output of the toolchain is a configuration bitstream that contains both the information on what the PE and MEM tiles should do, as well as the routing information (in the form of values of select bits for each multiplexer in the SBs and CBs). We then run a Verilog simulation of the configured CGRA to obtain the switching activity at all the nodes in the PE tiles that are active. We feed this into PrimeTime PX, along with the post-P&R PE tile netlist and parasitics. After modifying the capacitances/resistances, as detailed in Section V-B, PrimeTime is able to generate accurate delay/power numbers for each application. This is performed for three different values of R_{DS}, starting from $80\,\Omega$ (predicted by effective contact area model), up to $5\,\text{k}\Omega$ (observed experimentally in [10]).

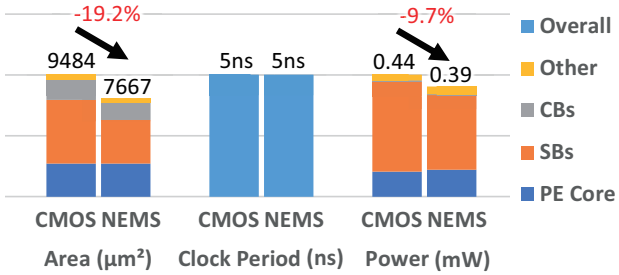

Fig. 10: Signoff power, performance, and area, for one CGRA PE tile of CMOS and NEMS designs ($R_{DS} = 80\,\Omega$).

We evaluate the baseline and our hybrid PE tile on three image processing applications (Table III). *Conv. 3 × 3* is a 2-D convolution with a 3 × 3 kernel. *Cascade* has two back-to-back 3 × 3 convolutions. *Harris* is a corner detector. We measure the power after the CGRA bitstream has been loaded, since the cost of configuration is amortized over long-running operation.

VII. Results

The power and area results are summarized in Fig. 10, with more detail on the application-level breakdown and results vs. R_{DS} in Table III. Our design shows 19% better area and 10% better power at iso-critical path delay (clock period = 5 ns) across the applications. The power reduction mainly occurs in the switch boxes, where having fewer CMOS standard cells results in less leakage and smaller dynamic power dissipation. The area reduction comes from direct mapping of the CMOS multiplexers in the SB/CB to NEM OHMuxes integrated in 3D. In order to evaluate how efficiently we integrated NEMS with CMOS, we examine the following area figure of merit (FOM):

$$\text{Area FOM} = \frac{\text{CMOS Design Stdcell Area} - \text{NEMS Design Stdcell Area}}{\text{SB/CB Mux Area}} \quad (5)$$

This FOM indicates what percentage of the maximum possible area savings (from moving SB/CB multiplexers to the NEM relay layer) was realized. We achieve an area FOM of **73.3%**. The reason this is less than 100% is due to the area overhead from decoders for the one-hot select signals, and increased area from logic surrounding the NEM routers to meet timing.

VIII. Conclusion

We explore the use of multi-pole NEM relays as multi-bit routers in CGRAs. We design an area-efficient multi-pole

TABLE III: CGRA PE tile power and area (per tile) of NEMS-based vs. the CMOS-only design. All entries at iso-delay (clock period = 5 ns).

PPA Metrics Summary			CMOS	NEMS	
		R_{DS}	Total	Total	% Improvement
Power (mW)	Conv 3x3 (20 tiles)	80 Ω	0.510	0.460	9.8
		1 kΩ		0.465	8.8
		5 kΩ		0.475	6.9
	Cascade (71 tiles)	80 Ω	0.401	0.362	9.8
		1 kΩ		0.366	8.8
		5 kΩ		0.373	7.0
	Harris (154 tiles)	80 Ω	0.396	0.357	9.9
		1 kΩ		0.360	8.9
		5 kΩ		0.368	7.1
Area (μm^2)	Overall	80 Ω	9484	7667	19.2
		1 kΩ		7673	19.1
		5 kΩ		7808	17.7

relay that can share anchors and be laid out in a single layer on top of CMOS. We demonstrate a methodology for reducing contact resistance variation by $>40\times$ in multi-pole NEM relays, equalizing contact forces using iterative FEM. Finally, we show that integration of these multi-pole relays into the PE tiles of a hybrid CMOS-NEMS CGRA can yield 19% lower area and 10% lower power at iso-delay. All of our tools/models are available on GitHub[1]. The results show a way for bridging the performance gap between programmable logic devices (such as CGRAs) and ASICs, using multi-pole NEMS technology.

List of Abbreviations

AOI—AND Gate-OR Gate-Inverter (AND-OR-INV)
ASIC—Application-Specific Integrated Circuit
BEOL—Back End Of Line
CB—Connection Box
CGRA—Coarse-Grained Reconfigurable Array
CMOS—Complementary Metal-Oxide-Semiconductor
FEM—Finite Element Model
ICP—Iterative Contact Placement
LUT—Look-Up Table
MEM—Memory
MOSFET—Metal-Oxide-Semiconductor Field-Effect Transistor
NEM—Nanoelectromechanical
NMOS—N-type Metal-Oxide-Semiconductor
OHMux—One-Hot Multiplexer
P&R—Place-and-Route
PE—Processing Element
PLD—Programmable Logic Device
PPA—Power Performance Area
SB—Switch Box

References

[1] M. Wijtvliet, L. Waeijen, and H. Corporaal, "Coarse grained reconfigurable architectures in the past 25 years: Overview and classification," in *2016 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS)*, pp. 235–244, IEEE, 2016.

[2] R. Bahr, C. Barrett, N. Bhagdikar, A. Carsello, R. Daly, C. Donovick, D. Durst, K. Fatahalian, K. Feng, P. Hanrahan, T. Hofstee, M. Horowitz, D. Huff, F. Kjolstad, T. Kong, Q. Liu, M. Mann, J. Melchert, A. Nayak, A. Niemetz, G. Nyengele, P. Raina, S. Richardson, R. Setaluri, J. Setter, K. Sreedhar, M. Strange, J. Thomas, C. Torng, L. Truong, N. Tsiskaridze, and K. Zhang, "Creating an agile hardware design flow," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, IEEE, 2020.

[3] A. Vasilyev, N. Bhagdikar, A. Pedram, S. Richardson, S. Kvatinsky, and M. Horowitz, "Evaluating programmable architectures for imaging and vision applications," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 1–13, IEEE, 2016.

[4] F. Chen, H. Kam, D. Markovic, T.-J. K. Liu, V. Stojanovic, and E. Alon, "Integrated circuit design with NEM relays," in *2008 IEEE/ACM International Conference on Computer-Aided Design*, pp. 750–757, IEEE Press, 2008.

[5] C. Chen, R. Parsa, N. Patil, S. Chong, K. Akarvardar, J. Provine, D. Lewis, J. Watt, R. T. Howe, H.-S. P. Wong, and S. Mitra, "Efficient FPGAs using nanoelectromechanical relays," in *18th ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pp. 273–282, ACM, 2010.

[6] C. Chen, W. S. Lee, R. Parsa, S. Chong, J. Provine, J. Watt, R. T. Howe, H.-S. P. Wong, and S. Mitra, "Nano-electro-mechanical relays for FPGA routing: Experimental demonstration and a design technique," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012*, pp. 1361–1366, IEEE, 2012.

[7] B. Osoba, B. Saha, L. Dougherty, J. Edgington, C. Qian, F. Niroui, J. H. Lang, V. Bulovic, J. Wu, T.-J. K. Liu, D. Marković, E. Alon, and V. Stojanović, "Sub-50 mV NEM relay operation enabled by self-assembled molecular coating," in *2016 IEEE International Electron Devices Meeting (IEDM)*, pp. 26–8, IEEE, 2016.

[8] M. Spencer, F. Chen, C. C. Wang, R. Nathanael, H. Fariborzi, A. Gupta, H. Kam, V. Pott, J. Jeon, T.-J. K. Liu, *et al.*, "Demonstration of integrated micro-electro-mechanical relay circuits for VLSI applications," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 308–320, 2010.

[9] C. W. Low, S. F. Almeida, E. P. Quévy, and R. T. Howe, *Poly-SiGe Surface Micromachining*, pp. 69–97. John Wiley & Sons, Ltd, 2021.

[10] R. Nathanael, "Nano-electro-mechanical (NEM) relay devices and technology for ultra-low energy digital integrated circuits," tech. rep., University of California Berkeley, Department of Electrical Engineering and Computer Science, 2013.

[11] R. Nathanael, V. Pott, H. Kam, J. Jeon, and T.-J. K. Liu, "4-terminal relay technology for complementary logic," in *2009 IEEE International Electron Devices Meeting (IEDM)*, pp. 1–4, IEEE, 2009.

[12] R. Holm, *Electric contacts: theory and application*. Springer Science & Business Media, 2013.

[13] S. C. Bromley and B. J. Nelson, "Performance of microcontacts tested with a novel MEMS device," in *Proceedings of the Forth-Seventh IEEE Holm Conference on Electrical Contacts (IEEE Cat. No. 01CH37192)*, pp. 122–127, IEEE, 2001.

[14] J. Ragan-Kelley, C. Barnes, A. Adams, S. Paris, F. Durand, and S. Amarasinghe, "Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines," *ACM SIGPLAN Notices*, vol. 48, no. 6, pp. 519–530, 2013.

[1] https://github.com/users/akashlevy/projects/4