

Efficient FPGAs using Nanoelectromechanical Relays

Chen Chen¹, Roozbeh Parsa¹, Nishant Patil¹, Soogine Chong¹, Kerem Akarvardar¹,
J Provine¹, David Lewis³, Jeff Watt⁴, Roger T. Howe¹, H.-S. Philip Wong¹, Subhasish Mitra^{1,2}

¹Department of Electrical Engineering
Stanford University
Stanford, CA 94305 USA

²Department of Computer Science
Stanford University
Stanford, CA 94305 USA

³Altera Corporation
151 Bloor St W, Suite 200
Toronto, Ontario
Canada M5S 1S4

⁴Altera Corporation
101 Innovation Drive
San Jose, CA 95134 USA

Abstract

Nanoelectromechanical (*NEM*) relays are promising candidates for programmable routing in Field-Programmable-Gate Arrays (*FPGAs*). This is due to their zero leakage and potentially low on-resistance. Moreover, *NEM* relays can be fabricated using a low-temperature process and, hence, may be monolithically integrated on top of CMOS circuits. Hysteresis characteristics of *NEM* relays can be utilized for designing programmable routing switches in *FPGAs* without requiring corresponding routing SRAM cells. Our simulation results demonstrate that the use of *NEM* relays for programmable routing in *FPGAs* can simultaneously provide 43.6% footprint area reduction, 37% leakage power reduction, and up to 28% critical path delay reduction compared to traditional SRAM-based CMOS *FPGAs* at the 22nm technology node.

Categories and Subject Descriptors

B.7.1 [Integrated Circuit]: Types and Design Styles – *Advanced technologies*.

General Terms

Design, Performance, Reliability

Keywords

CMOS-NEM FPGA, Nanoelectromechanical relay

1. Introduction

FPGAs provide popular digital design platforms due to low design costs and fast turnaround times [Kuon 07]. However, because a large number of transistors are required for programmable routing, *FPGAs* incur larger silicon area, lower performance and higher power consumption compared to ASICs. It was estimated that an *FPGA* may be ~30x larger, ~4x slower, and may consume ~10x more dynamic power compared to a standard-cell ASIC at the same technology node [Kuon 07]. Although *FPGA* vendors have developed innovative ways to reduce leakage power, e.g., multi-threshold transistors, body-biasing, thick-gate-oxide transistors, etc., leakage contributed by *FPGA* programmable routing resources is still a large proportion of the overall leakage power [Altera 08]. With technology scaling, leakage power is considered as a major challenge for *FPGAs*

targeting both high-performance and low-power applications [ITRS 07, Srinivasan 05].

There are mainly three types of commercial *FPGAs*: (1) SRAM-based, (2) anti-fuse-based, and, (3) Flash-based. Anti-fuse-based *FPGAs* are non-volatile, but are not reconfigurable [Kuon 07]. Flash-based *FPGAs* also have the benefit of non-volatility, but their integration with standard CMOS processes is challenging. SRAM-based *FPGAs* are currently very popular because, although volatile, they can be fabricated using standard CMOS manufacturing processes and can be reconfigured numerous times during the product lifetime [Kuon 07]. Therefore, in this work, we focus our analysis on SRAM-based *FPGA* architectures. In SRAM-based *FPGAs*, NMOS pass transistors, controlled by SRAM cells, are used for programmable routing. We use the terms “*routing switch*” and “*routing SRAM*” to refer to a routing pass transistor and the corresponding SRAM cell which controls the pass transistor.

In this paper, we analyze the benefits that may be obtained by integrating nanoelectromechanical (*NEM*) relays in CMOS *FPGAs*. *NEM* relays exhibit zero leakage (experimentally verified) and their on-resistance values are predicted to be smaller than that of the NMOS pass transistors [Akarvardar 07, Nathanael 09, Timsit 04]. Therefore, routing switches made out of *NEM* relays present unique opportunities to reduce power and improve performance of *FPGAs*. In addition, hysteresis in the current-voltage characteristics of *NEM* relays can enable replacement of an *FPGA* routing switch together with the corresponding routing SRAM cell entirely using a single *NEM* relay (when certain conditions are satisfied as detailed in Sec. 3). Moreover, *NEM* relays may be fabricated using a *back-end of line (BEOL)* CMOS process (i.e., processing of all metal interconnects, vias and inter-layer dielectric). Hence, *NEM* relays may be placed on top of CMOS transistors which can result in substantial reduction in the footprint area of an *FPGA*.

The main disadvantage of a *NEM* relay is its large mechanical switching delay (>1ns [Akarvardar 07, Chen 08]). However, this drawback can be avoided if *NEM* relays are used for *FPGA* routing switches because they do not change states after *FPGA* configuration.

The major contributions of this paper are:

- Two options for integrating *NEM* relays into SRAM-based *FPGA* architectures. We refer to such integrated *FPGAs* as *CMOS-NEM FPGAs*.
- Evaluation of the power, performance and area benefits of CMOS-NEM *FPGAs* vs. conventional CMOS *FPGAs* (referred to as *CMOS-only FPGAs* in this paper) at the 22nm technology node.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

FPGA '10, February 21-23, 2010, Monterey, California, USA.

Copyright 2010 ACM 978-1-60558-911-4/10/02...\$10.00.

Section 2 introduces NEM relays and their properties. Section 3 describes two options for designing CMOS-NEM FPGAs. Section 4 presents a quantitative comparison between CMOS-only and CMOS-NEM FPGAs. Related work is discussed in Sec. 5 and Sec. 6 concludes the paper.

2. NEM relays

2.1 Introduction of NEM relays

Figure 2.1 shows the structures of electrostatically-actuated 3- and 4-terminal (3T and 4T, respectively) NEM relays. A 3T NEM relay consists of: 1) a deflecting beam (connected to the source electrode), which forms the channel for current flow; 2) a gate electrode with a gap of g_0 from the beam which exerts force to the beam and determines the state of the switch; and 3) a drain electrode, which connects to the beam when the NEM-relay is in its on-state. In this paper, we focus on the beam structure that is made of metal or semiconductor using photolithography. (Carbon nanotube based NEM switches [Zhou 07] are not discussed here.) When V_{GS} is applied, the electrostatic force attracts the beam towards the gate, while the elastic force in the beam resists the beam from deflecting. Beyond a certain V_{GS} , called *pull-in voltage* (V_{pi}), the elastic force can no longer balance the electrostatic force, and the beam collapses toward the gate until contact is made at the drain. Since pull-in is achieved through electromechanical instability, the voltage at which the beam disconnects from the drain (*pull-out voltage*, V_{po}) is smaller than V_{pi} . This causes hysteresis in the current-voltage characteristics of NEM relays (Fig. 2.1). The *van der Waals* surface forces present at the contact area between the beam and drain can modify the *hysteresis window* ($V_{pi} - V_{po}$). The operation of 4T NEM relays (Fig. 2.1b) is similar to that of 3T relays. However, for 4T relays, the beam is electrically isolated from the source and drain electrodes, and two mechanical contacts need to be established in order to connect the source and drain electrodes [Chen 08]. A 3T NEM relay requires fewer fabrication steps compared to a 4T NEM relay. However, the additional electrode can make it more convenient to use 4T NEM relays as FPGA routing switches, as will be discussed in Sec. 3.

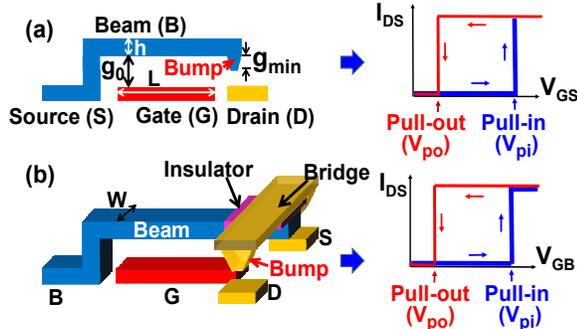


Figure 2.1: (a) A 3T NEM relay and its $I_{DS}-V_{GS}$ characteristics; (b) A 4T NEM relay and its $I_{DS}-V_{GB}$ characteristics. The beam is insulated from the bridge for S/D contacts [Chen 08].

V_{pi} and V_{po} can be designed by adjusting the physical dimensions of the NEM relays. According to the parallel plate model [Kaajakari 09], V_{pi} and V_{po} can be calculated as:

$$V_{pi} = \sqrt{\frac{8kg_0^3}{27\epsilon WL}}, \quad V_{po} = \sqrt{\frac{2k}{\epsilon WL} g_{min}^2 (g_0 - g_{min})},$$

where k is the spring constant of the beam depending on its geometry and material, ϵ is the permittivity of the ambient enclosing the relay, e.g., vacuum or oil [Lee 09b], g_0 is the gate-to-beam gap, and g_{min} is the minimum gap between gate and beam when beam is pulled down.

Experimental I-V characteristics of an actuated 3T NEM relay are shown in Fig. 2.2, where zero leakage is confirmed. The measured pull-in and pull-out voltages of the device are $V_{pi} = 6.7V$, $V_{po} = 5.5V$. The measured on-resistance of the NEM relay is $2k\Omega$. Although the actuation voltages of this fabricated device are relatively high, the pull-in and pull-out voltages as well as the hysteresis window of NEM relays can be adjusted through proper choice of their physical dimensions or beam material [Kaajakari 09]. For example, Fig. 2.3 shows the effect of changing beam length (L) on V_{pi} and V_{po} using a commercial micro-electromechanical simulator [COMSOL]. In our simulations, surface forces are not included. Inclusion of surface forces will lead to further smaller values of V_{pi} and V_{po} . The beam thickness (h) and beam-to-gate gap (g_0) are both 10nm in our simulation (which are reasonable values as discussed in [Jang 08, Lee 09a]).

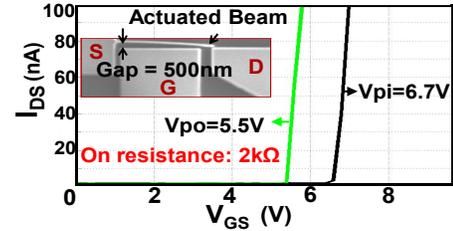


Figure 2.2: SEM image of a laterally actuated (pulled in) NEM relay fabricated in our laboratory and the corresponding $I_{DS}-V_{GS}$ characteristics. The current compliance of the measurement is 100nA.

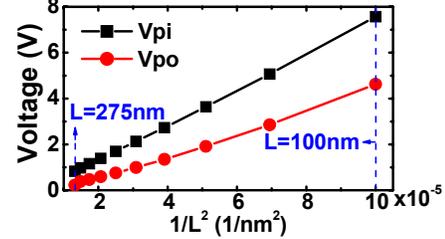


Figure 2.3: Effect of beam length (L) on V_{pi} and V_{po} . The beam material is copper (a typical interconnect material), beam thickness (h) and beam-to-gate gap (g_0) are 10nm.

In this paper, we are targeting the 22nm technology node. The dimensions of the corresponding NEM relays are shown in Fig. 2.4 in terms of λ ($=11nm$). With such dimensions, simulation results show that the pull-in voltage is $\sim 0.8V$ and the pull-out voltage is $\sim 0.5V$ for $g_0=10nm$ (surface forces are not included) [COMSOL].

2.2 Integration of NEM relays

All structural (i.e., beam and electrodes), contact, and sacrificial materials, used to fabricate NEM relays, can be typical materials used in the standard CMOS BEOL process [De Los Santos 04]. For example, the beam can be made with conductive materials such as nickel, platinum, aluminum [De Los Santos 04],

and titanium-nitride [Jang 08]. The low processing temperatures of these materials make NEM relays BEOL-compatible and provide the capability of stacking them between interconnect layers. After fabricating NEM relays between metal layers, they can be encapsulated (experimentally demonstrated by [Cavendish Kinetics, Jahnes 04]) so that further processing for the remaining metal interconnects can continue (Fig. 2.5).

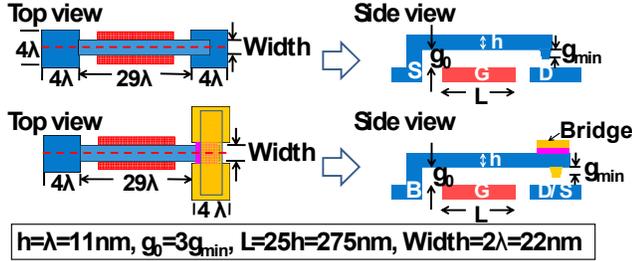


Figure 2.4: Layouts of 3T and 4T NEM relays used in this paper at the 22nm technology node.

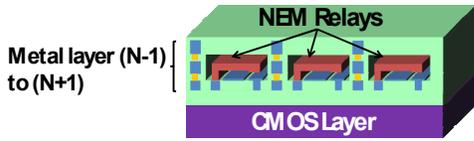


Figure 2.5: Encapsulated NEM relays between metal layers to enable monolithic 3D integration with silicon CMOS.

The operation of NEM relays introduces reliability issues that are distinct from those of CMOS transistors [Akarvardar 09a]. In addition to the presence of surface forces and the related stiction issue (i.e., the beam may not be pulled out after pull-in), the biggest concern is the mechanical contact reliability. Hot switching, high impact velocity of the beam tip during contact, and tip bouncing aggravate such reliability issues. The bump added at the beam tip in Fig. 2.1 is intended to alleviate some of these issues. For ultimately scaled NEM relays, the current density across the nanometer-sized contact spots could be a limiting factor [Akarvardar 09a]. The contact reliability, as well as the reproducibility and consistency of mechanical properties and adhesion forces, require hermetic sealing using either a wafer-bonding process or micro-shell encapsulation [Cavendish Kinetics, Jahnes 04] to provide a controlled environment for the NEM relays by isolating them from humidity and contaminants such as gases and organic compounds [Akarvardar 09a].

Despite these reliability issues, promising experimental data has been demonstrated for NEM relays which can switch reliably up to 10^{11} cycles [Cavendish Kinetics, Nathanael 09]. Since NEM relays will be used for programmable routing resources in CMOS-NEM FPGAs (as detailed later), the number of programming cycles is expected to be much smaller (e.g., < 500 according to [Kuon 07]) for typical FPGA users.

Though promising, lots of challenges remain to be solved before NEM relays can be incorporated into existing state-of-the-art CMOS-FPGAs. More research and experiments are needed to understand the manufacturability of NEM relays on top of CMOS, as well as the associated process costs, yield, and testing costs.

3. NEM relays for SRAM-based FPGAs

In this section, we discuss the use of NEM relays for replacing routing switches and routing SRAMs in SRAM-based FPGAs. We focus on the island-style FPGA architecture, which consists of

Logic Blocks (LBs) and programmable routing wires in routing channels connecting LBs (Fig. 3.1a). We choose the following architectural parameters according to [Kuon 08]. Each LB contains 10 four-input look-up tables (4-LUTs) and 10 flip-flops (FFs), along with 22 input pins and 10 output pins (Fig. 3.1b). To provide interconnections between different LBs, routing wires are distributed along the horizontal and vertical routing channels (Fig. 3.1a). The wires in the channel are directional single-driver wires, i.e., the wires can only be driven from one end [Lewis 03, Lemieux 04]. Channel width ($W = 104$) is defined as the number of wires in each routing channel [Kuon 08]. LB input pin flexibility ($F_{cin} = 0.2$) is the fraction of wires in the channel that can connect to each LB input pin. Similarly, LB output pin flexibility ($F_{cout} = 0.1$) is the fraction of wires in the channel that can connect to each logic block output pin [Kuon 08]. The connection block (CB) (Fig. 3.1c) is defined as the group of multiplexers that are used to connect the wires in the channel to LB input pins. The switch box (SB) is defined as the group of multiplexers used to connect starting points of wires to LB output pins and endpoints of other wires (Fig. 3.1d). Switch box flexibility ($F_s = 3$) is defined as the number of wire endpoints (in addition to LB output pins) that can be connected to the starting point of each wire (Fig. 3.1d). The entire SRAM-based FPGA can be considered as an array of tiles (Fig. 3.1a). Each tile contains one LB, two CBs and one SB. The routing wires in the channel are of length 4 (length-4 wire), i.e., they span four tiles in length [Kuon 08].

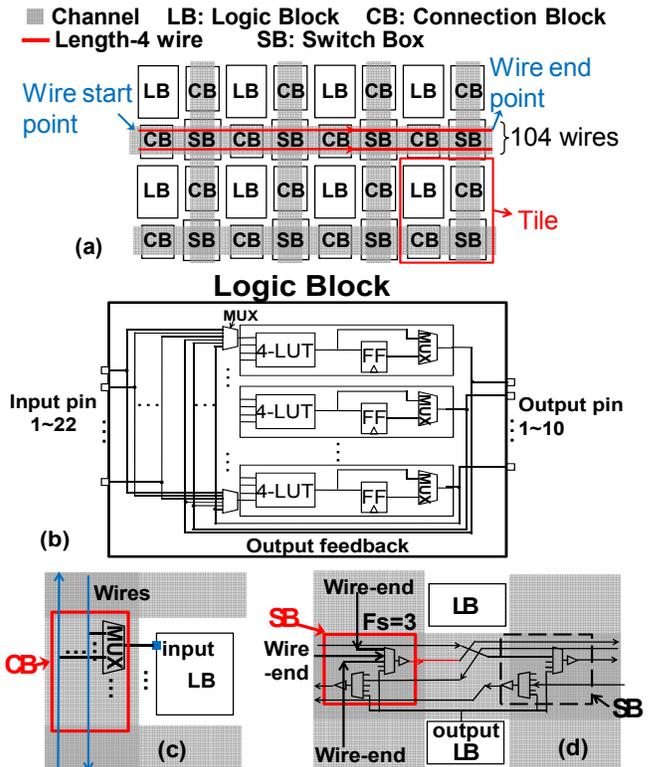


Figure 3.1: SRAM-based FPGA: (a) Overall architecture; (b) Logic block; (c) Connection block; (d) Switch box.

Existing commercial FPGAs are more complex than the architecture shown in Fig. 3.1. For example, LBs could contain fast carry chains, and some commercial FPGAs contain

customized blocks such as multipliers, memories, or even processors. Although we focus on the basic FPGA architecture in this paper, our design and analysis methodologies can be readily applied to such complex FPGAs as well because: 1) customized blocks use the same programmable routing resources; 2) our techniques modify only the programmable routing resources.

3.1 Option 1: NEM relays as FPGA routing switches

The simplest option for CMOS-NEM FPGA design is to replace FPGA routing switches with NEM relays, as shown in Fig. 3.2. Routing switches are grouped together in *routing multiplexers (routing MUXes)*, Fig. 3.2). CMOS-SRAM cells continue to be used to control the states of the NEM relay-based routing switches in this CMOS-NEM FPGA option. As described in Sec. 2.2, NEM relays may be placed between metal layers to enable monolithic 3D integration with CMOS (Fig. 3.2d).

The operation of a 4T NEM relay in this scenario is similar to that of an NMOS pass transistor, as long as the controlling SRAM can provide sufficient *gate-to-beam voltage* (V_{GB}) to ensure pull-in of the NEM relay (Fig. 3.3b). This requires the pull-in voltage of the 4T NEM relay to be smaller than the *output high voltage* of the controlling SRAM (V_{oh_SRAM}).

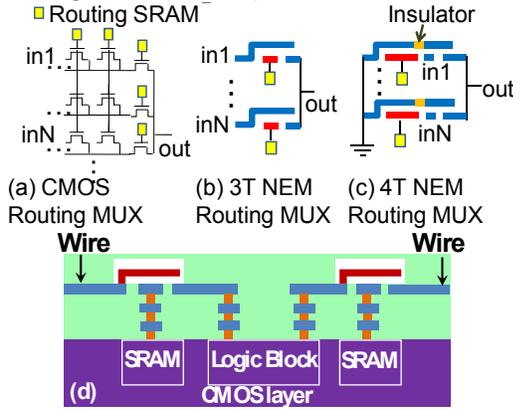


Figure 3.2: Routing MUX structures: (a) CMOS routing MUX; (b) 3T NEM routing MUX; (c) 4T NEM routing MUX; (d) Monolithic 3D integration of NEM relays with CMOS.

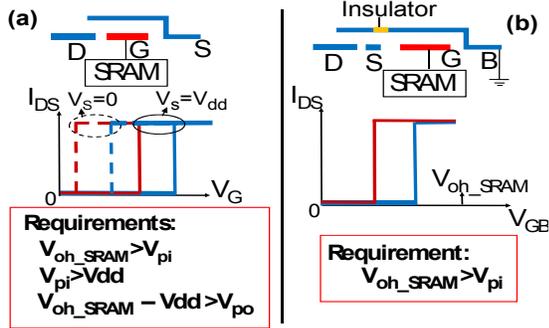


Figure 3.3: NEM relays as FPGA routing switches (Option 1) and the corresponding voltage requirements, where V_{oh_SRAM} is the output high of the SRAM, and V_{dd} is the supply voltage for the mapped circuit on the FPGA: (a) Using 3T NEM relay; (b) Using 4T NEM relay.

For a 3T NEM relay, the voltage of the source (S) electrode can change during normal circuit operation (between 0 and V_{dd} ,

the supply voltage of the circuit mapped on the FPGA). Consequently, the gate-to-source voltage (V_{GS}) is not fixed (Fig. 3.3a). Therefore, the constraints $V_{dd} < V_{pi}$ and $V_{oh_SRAM} - V_{dd} > V_{po}$ must be satisfied to prevent inadvertent pull-in and pull-out, respectively. These additional voltage constraints require that the pull-in voltage for 3T NEM relays be greater than that of 4T NEM relays. Furthermore, V_{oh_SRAM} should be greater than V_{dd} , imposing constraints on the supply voltage of CMOS transistors in SRAM cells. Hence, CMOS transistors in the routing SRAM cells must be able to withstand the higher supply voltage (e.g., using thick-oxide CMOS transistors such as those used in existing commercial FPGAs [Altera 08, Xilinx 09a]).

3.2 Option 2: a single NEM relay to replace both a routing switch and its routing SRAM

Hysteresis in $I_{DS}-V_{GS}$ characteristics of a NEM relay (Fig. 2.2) may be utilized to replace both the routing switch and its corresponding routing SRAM cell simultaneously using a single NEM relay. This is possible using a half-select programming scheme (Fig. 3.4a for 4T NEM relays) [Braun 08, Olsen 64].

In Fig. 3.4a, the row and column lines serve as programming lines. To preserve values stored in the array, all row lines are held at a constant *holding voltage* level (V_{hold}), and all column lines are connected to a voltage level defined as *select voltage* (V_S). V_{hold} must be set between V_{pi} and V_{po} to ensure that NEM relays in the array hold their states (Fig. 3.4c, requirements i and ii). During FPGA configuration, to pull in a particular NEM relay in the array (Fig. 3.4a), $V_{hold} + V_S$ is applied to the row line connected to this relay, and ground (GND) is applied to the corresponding column line. Hence, the gate-to-beam voltage (V_{GB}) of the relay to be programmed is $V_{hold} + V_S$, which must be greater than V_{pi} to guarantee pull-in (Fig. 3.4c, requirement iii). For any other NEM relay in the array, its gate-to-beam voltage is either V_{hold} or $V_{hold} - V_S$ depending on whether or not the NEM relay is in the same column as the NEM relay being currently programmed. In order to prevent other NEM relays from being inadvertently pulled out, $V_{hold} - V_S$ must be greater than V_{po} (Fig. 3.4c, requirement i). Since V_{hold} is smaller than V_{pi} , these NEM relays will not be pulled in inadvertently and will hold their states. Connecting all row and column lines to GND will reset the entire NEM array, i.e., will pull out all NEM relays. Figure 3.4d shows the overall organization of the NEM relays and the corresponding programming circuitry (multiplexers, controlled by shift registers, are used to select different programming voltage levels).

The row-column structure enabled by the half-select programming scheme for 4T NEM relays requires only one CMOS driver per row or column for the programming circuitry. Since contemporary FPGAs can contain $\sim 100M$ programmable bits [Xilinx 09b], arranging the NEM relays in such a row-column structure, the corresponding programming circuitry will require $\sim 2 \times 10^4$ drivers for the entire FPGA (row and column). We assume that the programming circuitry is built using CMOS multiplexers controlled by CMOS shift registers (Fig. 3.4d). Using a commercial 90nm CMOS technology, the layout area of a flip-flop is estimated to be $\sim 8000\lambda^2$ ($\lambda=45nm$). Hence, the estimated area of the programming shift registers is $\sim 1.6 \times 10^8 \lambda^2$. This is comparable to the area of 20 CMOS-NEM FPGA tiles (Sec. 4.6). Since commercial FPGAs usually contain more than a thousand tiles [Xilinx 09a, b], the area occupied by the

programming circuitry is expected to be less than 2% of the overall FPGA area.

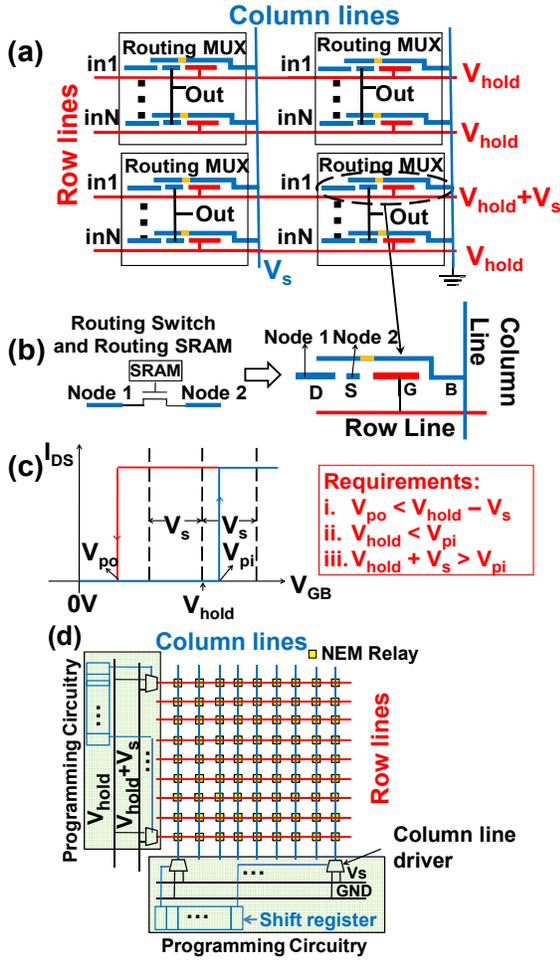


Figure 3.4: (a) Illustration of half-select programming scheme for 4T NEM relays; (b) Each 4T NEM relay in (a) can replace an NMOS routing switch and its corresponding routing SRAM cell; (c) Voltage requirements for half-select programming; (d) Overall architecture with programming circuitry.

For 3T NEM relays with half-select programming scheme, two issues must be addressed:

1) Similar to option 1 (Sec. 3.1), the 3T NEM relay will not have a fixed V_{GS} because the S electrode is used for routing, and it can switch between 0 and V_{dd} during normal circuit operation.

2) The S electrode of the NEM relay needs to be connected to the column line during programming mode and to the CMOS circuitry during normal circuit operation.

It may be challenging to address all these issues in order to use a 3T NEM relay for replacing an FPGA routing switch and its corresponding routing SRAM simultaneously (detailed discussion in the Appendix).

4. Simulation results

In this section, we present the analysis of speed, power, and area benefits of incorporating NEM relays into CMOS FPGAs. In commercial FPGAs, multi-threshold transistors may be used for power-performance tradeoffs [Altera 08, Xilinx 09a]. We use the

22nm low-power CMOS transistor model for routing switches and routing SRAMs in our simulations [Zhao 06, PTM]. For all other CMOS transistors, we use 22nm high-performance CMOS transistor model [Zhao 06, PTM]. The PTM interconnect model is used to estimate parasitic resistance and capacitance values of interconnect wires. We extract wire lengths from tile layouts. The supply voltage of the high-performance transistor is assumed to be equal to the nominal supply voltage provided by PTM model (0.8V). We choose the supply voltage of the low-power transistors to be 1.2V, which is 26% higher than their PTM nominal supply voltage (0.95V). This higher supply voltage is intended to: 1) avoid circuit malfunction due to threshold voltage drop caused by the low-power NMOS pass transistors; 2) speed up the low-power NMOS pass transistors.

4.1 NEM relay model

For performance estimation, we consider the case when the FPGA has already been configured (i.e., no switching of NEM relays). The impact of configuration time will be discussed later in this section. A NEM relay in the off-state (i.e., beam not pulled in) is modeled as an open circuit. To model a NEM relay in the on-state, we use the equivalent circuit shown in Fig. 4.1, where R_{on} is the total resistance between the source and the drain. R_{on} is dominated by the contact resistance [Akarvardar 07, Timsit 04]. The value of the contact resistance depends on many factors, such as contact material, fabrication process, etc., and is highly device and technology dependent. In [Chen 08], 100 Ω (gold beam) and 1k Ω (tungsten beam) were assumed for R_{on} . $R_{on} \sim 8k\Omega$ has also been demonstrated experimentally for 4T NEM relays [Nathanael 09]. The measured R_{on} of our fabricated 3T NEM relay is around 2k Ω (Fig. 2.2). However, to be conservative, we provide an analysis where we sweep the value of R_{on} from 100 Ω to 100k Ω [Akarvardar 07, Timsit 04]. C_{tot} in Fig. 4.1 is the total capacitance of a NEM relay. For a 3T NEM relay, C_{tot} mainly arises from the gate-to-beam capacitance (Fig. 4.1a). For a 4T NEM relay, C_{tot} arises from the overlap between the bridge (Sec. 2.1) and the gate electrode (Fig. 4.1b) [Chen 08]. Since the gate-to-beam overlap of the 3T NEM relay is much larger than the bridge to beam overlap of the 4T NEM relay (Fig. 2.4), a 4T NEM relay incurs smaller C_{tot} than a 3T relay. For the layouts in Fig. 2.4, simulation results using a commercial micro-electro-mechanical simulator [COMSOL] indicate that $C_{tot} \leq 20aF$ for both 3T and 4T relays. As a comparison point, an NMOS transistor with $width=4\lambda$ in a 22nm technology has a source and drain junction capacitance of 26aF [PTM, Zhao 06].

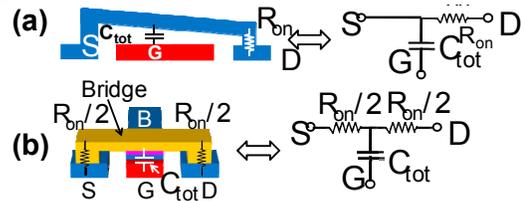


Figure 4.1: Equivalent circuit for an on-state NEM relay: (a) 3T NEM relay; (b) 4T NEM relay.

The area difference between 3T and 4T NEM relays is small (Fig. 2.4) because the beam area dominates the total area. Furthermore, when sweeping R_{on} , the delay differences between the two types of relays (Fig. 4.1) are also very small (<1%).

Hence, in the following analyses, we do not differentiate between 3T and 4T NEM relays.

4.2 Process variations and noise

In addition to voltage constraints (Sec. 3), sufficient margins should be allowed to tolerate noise (during normal FPGA operation) and variations in V_{pi} and V_{po} (which may result from manufacturing uncertainties). These challenges may be mitigated by carefully adjusting V_{pi} and V_{po} during design by properly choosing the dimensions of the NEM relays.

For example, for Option 2 using 4T NEM relays, to retain the configured state of the NEM relay, three margins (shown in Fig. 4.2) should be greater than the variations in V_{pi} and V_{po} , and the noise during normal FPGA operation. For Option 2, larger noise and variations in V_{pi} and V_{po} essentially require larger hysteresis window. Several parameters (e.g., beam length (L), beam thickness (h), gate-to-beam gap (g_0 and g_{min})) can be used to enlarge the hysteresis window of NEM relays.

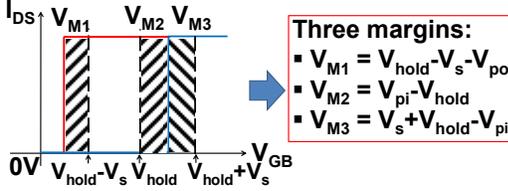


Figure 4.2: Margins needed to tolerate variations in V_{pi} and V_{po} due to manufacturing uncertainties, and noise during normal FPGA operation for CMOS-NEM FPGA Option 2 using 4T NEM relays (figure not drawn to scale).

4.3 Layout and transistor sizing

As described in Sec. 2.2, NEM relays may be placed on top of CMOS to enable 3D-integration. To provide a fair comparison between CMOS-only FPGA and CMOS-NEM FPGAs, actual FPGA layouts are needed because of the following reasons:

- 1) Interconnect wire capacitances cannot be neglected during transistor sizing, which affect overall speed and power.
- 2) Placing NEM relays on top of CMOS may block some interconnects. Hence, actual layout is essential to obtain reasonable area estimates, based on which we extract the interconnect wire capacitances.

An SRAM-based FPGA is an array of tiles (Sec. 3). Hence, we generated the entire FPGA by repeating the layout of a single tile [Kuon 07]. Using a commercial 90nm technology library, we created the *baseline CMOS-only FPGA* tile and CMOS-NEM FPGA tile layouts for both Options 1 and 2. The baseline CMOS-only FPGA tile layout is shown as an example in Fig. 4.3a. According to [ITRS 07, PTM], we scaled the layouts down to 22nm and extracted the corresponding interconnect capacitances.

For CMOS-NEM FPGAs, NEM relays are distributed between metal 3 and metal 5 (Fig. 2.5). All NEM relays in the same routing multiplexer (MUX) share the same encapsulation because they are physically close to each other (Sec. 2.2).

To size CMOS transistors in the baseline CMOS-only and CMOS-NEM FPGAs, we focus on a path from a Logic Block (LB) flip-flop output to another LB flip-flop input, as shown in Fig. 4.3b. The interconnect wire capacitances along the path are extracted from actual layouts. We divide the path into several circuit components, such as 4-LUT, Connection Block routing MUX, Switch Box routing MUX, etc. For each circuit component, we size the transistors for minimum area-delay

product [Kuon 08], where the area is calculated as the sum of all transistor widths in that circuit component.

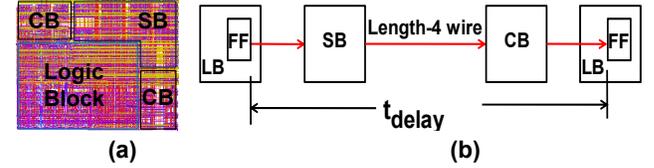


Figure 4.3: (a) Baseline CMOS-only FPGA tile layout; (b) A path from a FF output to another LB's FF input.

There are three different ways to build a CMOS routing MUX (Fig. 4.4a). Figure 4.4b shows the comparison between the area-delay products of these three MUX structures. The two-stage MUX structure has the minimum area-delay product because it is faster than the tree MUX (fewer serial pass transistors) and takes smaller area than the one-stage MUX (smaller number of SRAM bits). Hence, we choose the two-stage MUX structure to build routing MUXes in our baseline CMOS-only FPGA.

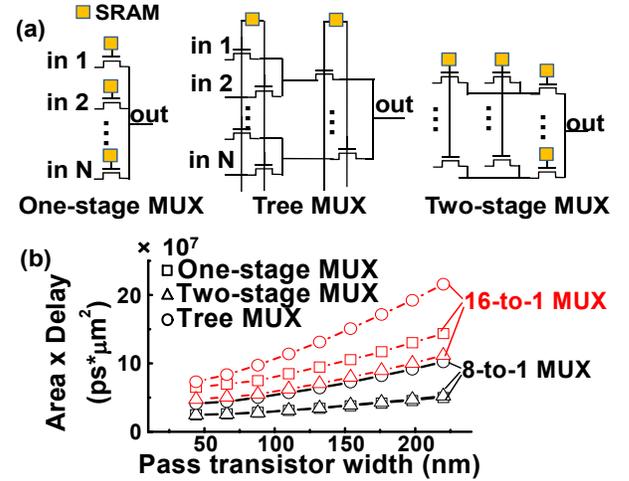


Figure 4.4: (a) Three different CMOS routing MUX structures; (b) Comparison between the area-delay products of the three MUX structures.

For CMOS-NEM FPGA Option 1 (i.e., only NMOS pass transistors are replaced with NEM relays), since CMOS SRAM cells are still needed, a two-stage MUX structure is still the best option in terms of area-delay product. However, for Option 2, the use of half-select programming eliminates SRAM cells and, hence, the number of NEM relays determines the layout area. Therefore, for Option 2, the optimal MUX structure for NEM routing MUX is no longer the two-stage but the one-stage structure. Also, because no NEM relays are in series in the one-stage MUX structure, the one-stage NEM-MUX is faster than the two-stage NEM-MUX over a wide range of R_{on} (Fig. 4.5c).

4.4 Overall performance improvement

Compared to the baseline CMOS-only FPGA, the path (shown in Fig. 4.3b) delay reduction of the two CMOS-NEM FPGA options are shown in Fig. 4.6. Option 2 can result in higher delay reduction than Option 1. This is because: 1) Option 2 can lead to smaller tile layout area, as will be shown in Sec. 4.6; 2) one-stage MUX structure used in Option 2 is faster than the two-stage one in Option 1, especially when R_{on} is high (Fig. 4.5).

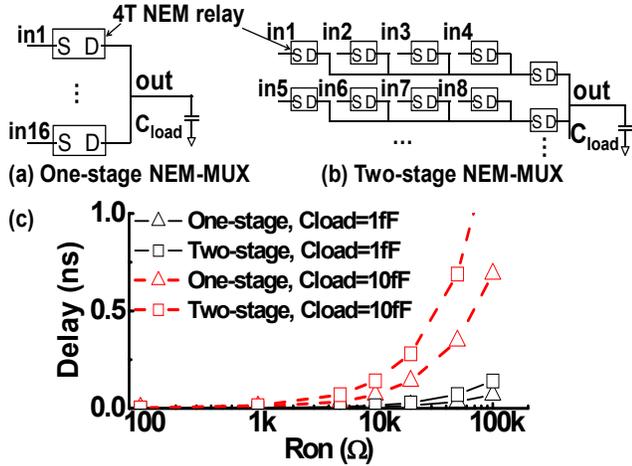


Figure 4.5: Comparison between two 16-to-1 NEM routing MUX structures for Option 2 using 4T NEM relay: (a) One-stage NEM-MUX; (b) Two-stage NEM-MUX; (c) Speed comparison between the two MUX structures with different load and R_{on} .

However, the overall performance benefits may not necessarily be the same as the delay reduction obtained for a single path. The path in Fig. 4.3b is the shortest path between two flip-flops in two different LBs, which may not be the same as the critical path of the circuit mapped on the FPGA. Hence, we use the Versatile Place and Route (VPR) tool (FPGA placement and routing tool) [Betz 97] to place and route 20 largest MCNC benchmark circuits [Yang 91] and 3 relatively large ($>10k$ LUTs) circuits from QUIP benchmark design set [Altera QUIP, Pistorius 07]. The inputs needed by VPR are extracted from the simulation results of the path in Fig. 4.3b. Figure 4.7 shows the reduction in critical path delay for the circuits mapped on a CMOS-NEM FPGAs compared with those mapped on the baseline CMOS-only

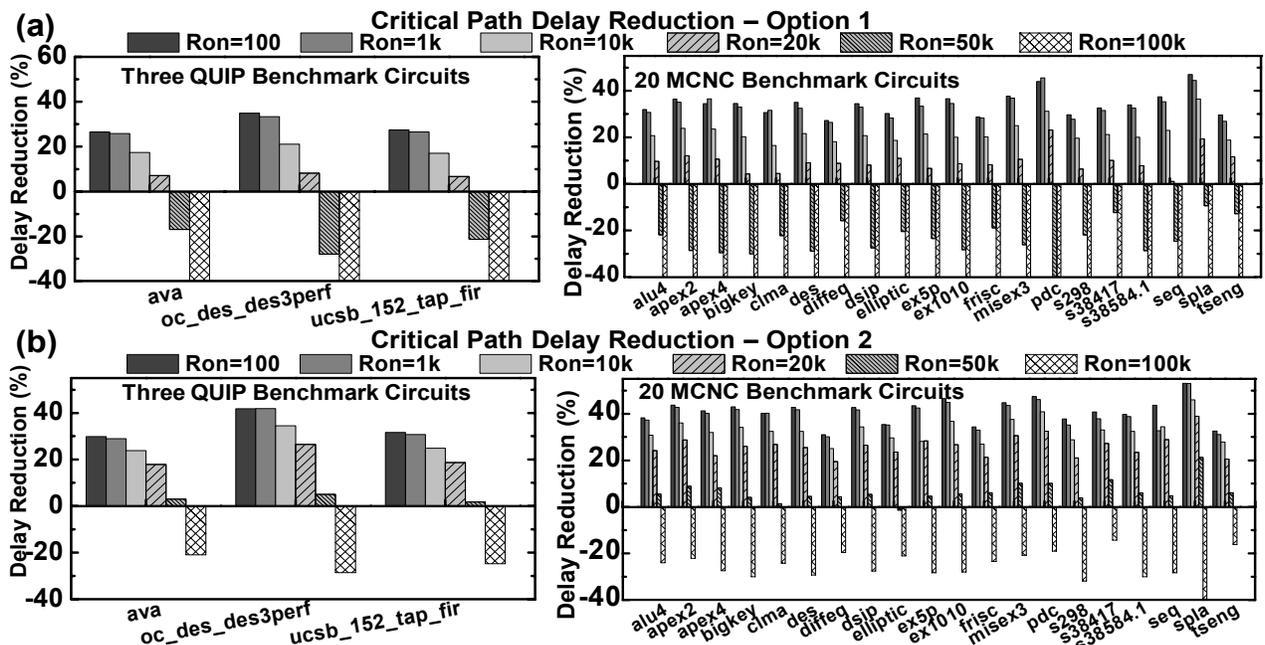


Figure 4.7: Critical path delay reduction compared to the baseline CMOS-only FPGA: (a) Option 1; (b) Option 2.

FPGA (Sec. 4.3). The highest average delay reduction is $\sim 40\%$ when the value of R_{on} lies between 100Ω to $1k\Omega$. The delay benefit reduces when R_{on} increases, and CMOS-NEM FPGAs becomes slower than the baseline CMOS-only FPGA when R_{on} approaches $100k\Omega$.

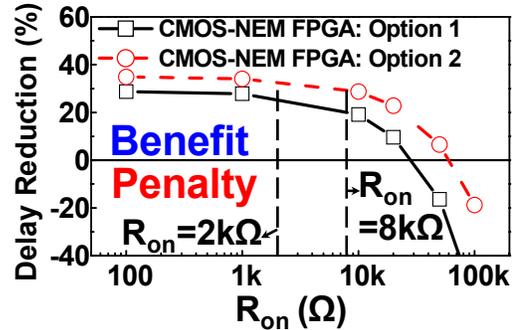


Figure 4.6: Delay reduction for the path shown in Fig. 4.3b compared to the baseline CMOS-only FPGA (NEM relay R_{on} values of $2k\Omega$ and $8k\Omega$ have been experimentally demonstrated).

For reconfiguration time, Option 1 is still dominated by programming the controlling SRAM cells. Hence, the additional programming time (after the SRAM cells have been programmed) is equal to the mechanical delay of the NEM relay (e.g., $1ns$ [Akarvardar 07]). For Option 2, the NEM relays can be programmed row by row (Fig. 3.4d). We assume there are 10^4 rows ($100M$ programmable bits [Xilinx 09b] arranged in a half-select array with equal number of rows and columns) and the mechanical delay of the NEM relay is $1ns$ [Akarvardar 07]. Excluding the shift-in delay for the programming shift registers and the transition delay of the row and column lines, the additional reconfiguration time imposed by NEM relays is $\sim 10\mu s$ ($10^4 \times 1ns$).

4.5 Power reduction

To estimate FPGA leakage power, we use the method described in [Tuan 03]. We divide the FPGA into several basic components, such as LUTs, MUXs, and wires. We use HSPICE to simulate the leakage power for each component. The total FPGA leakage power is obtained by adding up the leakage power of all the basic components in the FPGA.

Figure 4.8a shows the average (over all the benchmark circuits in Fig. 4.7) leakage power reduction of CMOS-NEM FPGAs compared to the baseline CMOS-only FPGA.

Since the optimized (i.e., minimum area-delay product) CMOS-NEM FPGAs can provide up to 40% critical path delay reduction compared to the baseline CMOS-only FPGA, it is possible to trade-off speed for further leakage power reduction. For example, we can resize the routing buffers (i.e., reduce the size of the routing buffers) in Option 2 to further reduce power (referred to as *low power* entry in Fig. 4.8b). As shown in Fig. 4.8b, resizing the buffers improves the leakage power reduction of Option 2 to 37% while the average critical path delay reduction drops to ~28% (instead of ~40% in Fig. 4.7b).

For dynamic power estimation, we focus on the dynamic power associated with the programmable interconnects because of two reasons: 1) the programmable interconnects can contribute up to 70% of the total FPGA power [Li 05]; 2) for both of our CMOS-NEM FPGA options, we only use NEM relays as programmable routing switches without any changes in CMOS LUTs. Therefore, the dynamic power associated with the CMOS LUTs does not change. As shown in Fig. 4.9, the dynamic power reduction is 4% for Option 1 and 22% for Option 2 (when optimized for minimum area-delay product). The dynamic power reduction mainly comes from the layout area reduction, as shown in Sec. 4.6.

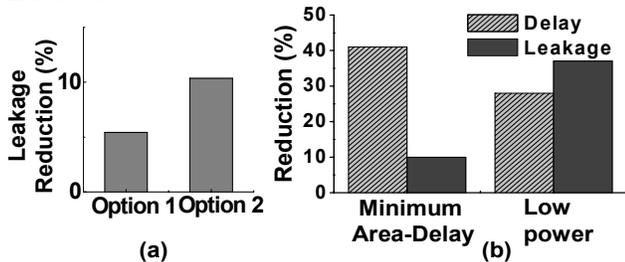


Figure 4.8: Leakage power reduction compared to the baseline CMOS-only FPGA: (a) Average leakage power reduction; (b) Delay-Leakage trade-off for Option 2.

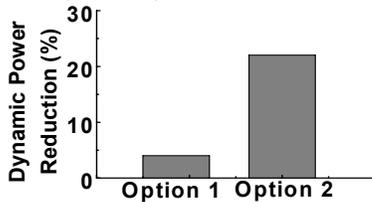


Figure 4.9: Dynamic power reduction for CMOS-NEM FPGAs.

4.6 Area benefits

Table 1 summarizes the areas of the FPGA tile layouts we have created (Sec. 4.3), including the baseline CMOS-only FPGA and the two CMOS-NEM FPGA options (optimized for minimum area-delay product). The layouts are created using a 90nm technology library, and the results are shown in terms of λ

(=45nm). By stacking NEM relays on top of CMOS, the area benefit of CMOS-NEM FPGA is 12.8% for Option 1, and 43.6% for Option 2.

Table 1: FPGA tile layout area report along with corresponding delay and power reduction

| | CMOS-only | CMOS-NEM-FPGA Option1 | CMOS-NEM-FPGA Option2 |
|-------------------------|---------------------|-----------------------|-----------------------|
| Area (λ^2) | 3300 \square 2600 | 3400 \square 2200 | 2200 \square 2200 |
| Normalized to CMOS | 1 | 0.872 | 0.564 |
| Delay reduction | - | ~30% | ~40% |
| Leakage power reduction | - | 5% | 10% |
| Dynamic power reduction | - | 4% | 22% |

4.7 Further FPGA architecture optimization

In this paper, we replaced FPGA routing switches and routing SRAMs using NEM relays without any additional changes in the FPGA architecture. The breakdown of the contributions of the various components of the baseline CMOS-only FPGA to tile area, path delay and leakage power are shown in Fig. 4.10. Due to the directional single driver FPGA architecture [Lewis 03, Lemieux 04], routing buffers contribute to a large portion of leakage power and path delay (dynamic power is still dominated by interconnect (Sec. 4.5)). Since we are only replacing routing switches and routing SRAMs with NEM relays, routing buffers limit the maximum benefits that may be achieved using NEM relays. Future research is necessary to explore FPGA architectural modifications that may result in further benefits from using NEM relays.

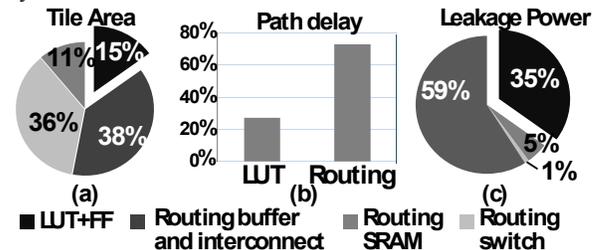


Figure 4.10: Breakdown of the contributions of different components to tile area, path delay and leakage power of the baseline CMOS-only FPGA: (a) Tile area; (b) Path delay; (c) Leakage power.

5. Related work

Design of FPGAs using emerging nanotechnologies is an important research field. Many researchers have explored the use of novel devices in FPGAs, such as carbon nanotubes (CNTs), nanowires, etc. [Chilstedt 09, Gojman 06, Tahoori 09]. Our focus in this paper is on NEM relay-based switches.

Several publications have reported possible benefits of NEM switches. Most of these publications discuss non-FPGA applications. For example, [Akarvardar 07] provides insights of using NEM relays as complementary logic gates. [Choi 07] discusses the use of NEM relays as non-volatile memory. [Dadgour 07] discusses the feasibility of using NEM switches in dynamic gates, SRAM cells and sleep transistors for ultra-low power applications. [Chong 09] presents a detailed discussion of

the benefits of using NEM relays inside SRAM cells. The use of NEM relays to design digital logic and analog circuits has also been explored [Chen 08, Akarvardar 09b].

There are a few papers discussing the use of NEM relays for FPGAs. [Zhou 07] describes a hybrid CMOS-NEMS approach for FPGA design using carbon nanotube (CNT)-based NEM switches. However, they focus on CNT-based NEM switches to replace SRAM cells in LUTs.

In this paper, we focus on NEM relays as FPGA routing switches. NEM relays can significantly improve FPGA performance, and reduce leakage and area when used as routing switches. The impact of large mechanical delays of NEM relays can be avoided except during FPGA configuration.

6. Conclusion

This paper demonstrates that NEM relays are promising candidates for improving FPGA performance and reducing FPGA power and area. To achieve such benefits, we present two different options of integrating NEM relays into CMOS FPGAs using 3-terminal and 4-terminal NEM relays. The speed, leakage power, and area of such NEM relay-based FPGAs are estimated and compared to those of CMOS-only FPGAs at the 22nm technology node. Moreover, technology parameters of NEM relays that directly impact speed, area and power benefits are identified and their effects are quantified. The best benefits are obtained by replacing both an FPGA routing switch and its corresponding routing SRAM using a single NEM relay. This can result in 28% critical path delay reduction, 37% leakage power reduction and 43.6% area reduction (simultaneously) compared to CMOS-only FPGA at the 22 nm technology node. Although 3T NEM relays require fewer fabrication steps, the use of 3T NEM relays can be more challenging due to the additional voltage constraints detailed in this paper.

Future research questions that remain to be explored include:

- 1) incorporating NEM relays into LUT designs;
- 2) detailed analysis of the voltage requirements for CMOS-NEM FPGA Option 1 and Option 2 to address noise and variation issues;
- 3) architectural exploration of FPGAs to obtain benefits from integrating NEM relays.

7. Acknowledgements

This work was sponsored by DARPA (NBCH 1090002). The authors would like to thank DARPA program managers Dr. Akintunde I. Akinwande and Dr. Amit Lal for their support. We would also like to thank Prof. Deming Chen of the University of Illinois at Urbana-Champaign for his comments on the paper.

References

- [Akarvardar 07] Akarvardar, K., *et al.*, "Design Considerations for Complementary Nanoelectromechanical Logic Gates," *Proc. Intl. Electron Dev. Meeting*, pp. 299-302, 2007.
- [Akarvardar 09a] Akarvardar, K., *et al.*, "Nanoelectromechanical Logic and Memory Devices," *ECS transactions*, Vol. 19, No. 1, pp. 49-59, 2009.
- [Akarvardar 09b] Akarvardar, K., *et al.*, "Analog Nanoelectromechanical Relay With Tunable Transconductance," *IEEE Electron Device Lett.*, Vol. 30, No. 11, pp. 1143-1145, 2009.
- [Altera 08] "40-nm Power Management and Advantages," Altera, 2008.
- [Altera QUIP] "Quartus II University Interface Program," Altera, www.altera.com/education/univ/research/unv-quip.html
- [Betz 97] Betz, V. and J. Rose, "VPR: A new packing placement and routing tool for FPGA research," *Proc. Intl. Workshop Field-Programmable Logic Appl.*, pp. 213-222, 1997.
- [Betz 99] Betz, V., *et al.*, "Architecture and CAD for Deep-Submicron FPGAs," *Kluwer Academic Publishers*, 1999.
- [Braun 08] Braun, S., *et al.*, "Row/Column Addressing Scheme for Large Electrostatic Actuator MEMS Switch Arrays and Optimization of the Operational Reliability by Statistical Analysis," *J. MEMS*, Vol. 17, No. 5, 2008.
- [Chen 08] Chen, F., *et al.*, "Integrated Circuit Design with NEM Relays," *Proc. Intl. Conf. CAD*, pp. 750-757, Nov. 2008.
- [Chilstedt 09] Chilstedt, S., *et al.*, "Design and Evaluation of a Carbon Nanotube-Based Programmable Architecture," *Intl. J. Parallel Programming*, Vol. 37, No. 4, pp. 389-416, 2009.
- [Choi 07] Choi, W.Y., *et al.*, "Compact Nano-Electro-Mechanical Non-Volatile Memory (NEMory) for 3D Integration," *Proc. Intl. Electron Dev. Meeting*, pp. 603-606, 2007.
- [Chong 09] Chong, S., *et al.*, "Nanoelectromechanical (NEM) Relay Integrated with CMOS SRAM for Improved Stability and Low Leakage," *Proc. Intl. Conf. CAD*, Nov. 2009.
- [COMSOL] <http://www.comsol.com/>
- [Dadgour 07] Dadgour, H. F., and K. Banerjee, "Design and Analysis of Hybrid NEMS-CMOS Circuits for Ultra Low-Power Applications," *Proc. Design Automation Conf.*, pp. 306-311, 2007.
- [De Los Santos 04] De Los Santos, H. J., *et al.*, "RF MEMS for Ubiquitous Wireless Connectivity: Part 1-Fabrication," *IEEE Microwave Magazine*, pp. 36-49, Dec. 2004.
- [Gojman 06] Gojman, B., *et al.*, "3D Nanowire-Based Programmable Logic," *Proc. Intl. Conf. Nano-Networks and Workshops*, pp. 1-5, 2006.
- [ITRS 07] ITRS Report, *Executive Summary*, 2007.
- [Jahnes 04] Jahnes, C. V., *et al.*, "Simultaneous Fabrication of RF MEMS Switches and Resonators using Copper-based CMOS Interconnect Manufacturing Methods," *Proc. Intl. Conf. Micro Electro Mechanical Systems (MEMS)*, pp. 789-792, 2004.
- [Jang 08] Jang, W. W., *et al.*, "Fabrication and characterization of a nanoelectromechanical switch with 15-nm-thick suspension air gap," *Appl. Phys. Lett.*, Vol. 92, No. 10, pp. 103-110, 2008.
- [Kaajakari 09] V. Kaajakari, *Practical MEMS, Small Gear Publishing*, 2009.
- [Kuon 07] Kuon, I., *et al.*, "FPGA Architecture: Survey and Challenges," *Foundations and Trends in Electronic Design Automation*, Vol. 2, No. 2, pp. 135-253, 2007.
- [Kuon 08] Kuon, I. and J. Rose, "Area and Delay Trade-offs in the Circuit and Architecture Design of FPGAs," *Proc. ACM Intl. Symp. FPGA*, pp. 149-158, 2008.
- [Lee 09a] Lee, D., *et al.*, "Scaling Limitations for Flexural Beams Used in Electronmechanical Devices," *IEEE Trans. Electron Devices*, Vol. 56, No. 4, pp. 688-691, 2009.
- [Lee 09b] Lee, J.-O., *et al.*, "3-Terminal Nanoelectromechanical Switching Device in Insulating Liquid Media for Low Voltage Operation and Reliability Improvement," *Proc. Intl. Electron Dev. Meeting*, Paper 9.5, 2009.
- [Lemieux 04] Lemieux, G., *et al.*, "Directional and Single-Driver Wires in FPGA Interconnect," *Proc. Intl. Conf. Field-Programmable Technology*, pp. 41-48, 2004.

[Lewis 03] Lewis, D., *et al.*, “The Stratix™ Routing and Logic Architecture,” *Proc. ACM Intl. Symp. FPGA*, pp. 10-12, 2003.

[Li 05] Li, F., *et al.*, “Power Modeling and Characteristics of Field Programmable Gate Arrays,” *IEEE Trans CAD*, Vol. 24, No. 11, pp. 1712-1724, Nov. 2005.

[Nathanael 09] Nathanael, R., *et al.*, “4-Terminal Relay Technology for Complementary Logic,” *Proc. Intl. Electron Dev. Meeting*, Paper 9.4, 2009.

[Olsen 64] Olsen, K.H., *et al.*, “Magnetic Core Memory,” *U.S. Patent 3161861*, 1964.

[Pistorius 07] Pistorius, J., *et al.*, “Benchmarking Method and Designs Targeting Logic Synthesis for FPGAs,” *Proc. Intl. Workshop Logic and Synthesis* pp. 230-237, 2007

[PTM] <http://www.eas.asu.edu/~ptm>

[Srinivasan 05] Srinivasan, S., *et al.*, “Leakage control in FPGA routing fabric,” *Proc. ASP-DAC*, Vol. 1, pp. 661–664, 2005.

[Tahoori 09] Tahoori, M. B., “Low-Overhead Defect Tolerance in Crossbar Nanoarchitectures,” *ACM J. Emerging Technol. Comput. Syst.*, Vol. 5, No. 2, July 2009.

[Timsit 04] Timsit, R.S., “Electrical Conduction through Small Contact Spots,” *Proc. IEEE Holm Conf. Electrical Contacts*, pp. 184-191, 2004.

[Tuan 03] Tuan, T., *et al.*, “Leakage Power Analysis of a 90nm FPGA,” *Proc. Custom Integrated Circuits Conf.*, pp. 57-60, 2003.

[Xilinx 09a] “Power Consumption at 40 and 45nm,” Xilinx, 2009.

[Xilinx 09b] Xilinx Inc., *Virtex-6 Family Overview*, Xilinx, www.xilinx.com/support/documentation/data_sheets/ds150.pdf

[Yang 91] Yang, S., “Logic synthesis and optimization benchmarks, version 3.0,” *Tech. Report, MCNC*, 1991.

[Zhao 06] Zhao, W., Y. Cao, “New generation of Predictive Technology Model for sub-45nm early design exploration,” *Proc. Intl. Symp. Quality Electronic Design*, pp. 585-590, 2006.

[Zhou 07] Zhou, Y., *et al.*, “Low Power FPGA Design Using Hybrid CMOS-NEMS Approach,” *Proc. Intl. Symp. Low Power Electronics and Design*, pp. 14-19, 2007.

Appendix

For 3T NEM relays with half-select programming scheme, two issues must be addressed:

1) Similar to option 1 (Sec. 3.1), the 3T NEM relay will not have a fixed V_{GS} because the S electrode is used for routing, and it can switch between 0 and V_{dd} during normal circuit operation.

2) The S electrode of the NEM relay needs to be connected to the column line during programming mode and to the CMOS circuitry during normal circuit operation.

To address the first issue for Option 2 using 3T NEM relay (Sec. 3.2), $V_{hold} < V_{dd}$ (which is the smallest possible gate-to-beam voltage) must be greater than V_{po} in order to avoid erroneous pull-out during normal FPGA operation (Fig. A.1c, requirement i). This requires 3T NEM relays to have: a) hysteresis window greater than V_{dd} ; b) pull-in voltage greater than that of 4T relays.

The second issue may be addressed by adding an additional *programming transistor* for each routing MUX that connects the S node of the NEM relay to the column line during configuration and disconnects the column line from the S node during normal circuit operation (Fig. A.1a, b). The gate of the programming transistor is connected to a control voltage signal ($V_{program}$). When configuring the FPGA, $V_{program}$ will be connected to V_{dd} to turn on the

programming transistor, thus connecting the beam of the NEM relay to the corresponding column line to enable half-select programming. After programming all the NEM relays, $V_{program}$ will be connected to GND to turn off the programming transistors, so that the S electrodes can be used for FPGA routing. The programming transistor can be shared by all the NEM relays within the same NEM routing MUX because the source nodes of these NEM relays are wired together to form the output of the routing MUX.

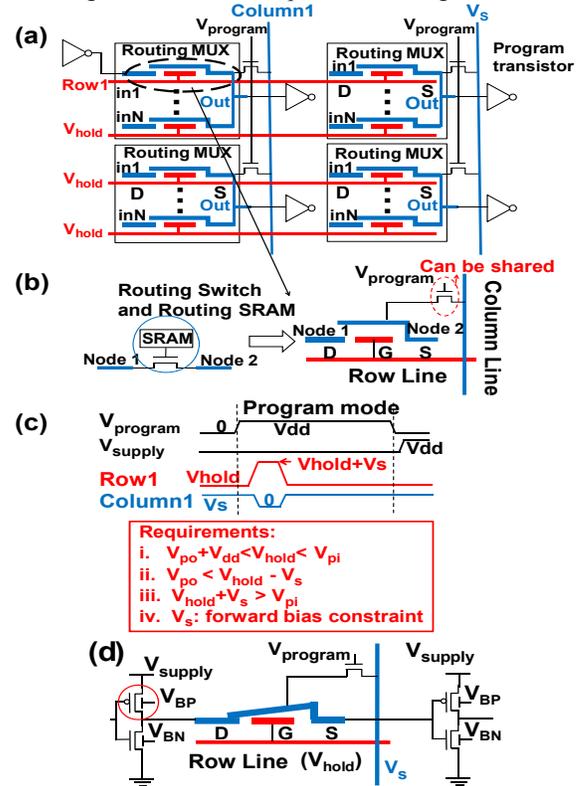


Figure A.1: (a) Illustration of half-select programming scheme for 3T NEM relays; (b) Each 3T NEM relay in (a) can replace an NMOS routing switch and its corresponding SRAM cell; (c) Waveforms to pull in the highlighted NEM relay in (a) and the voltage requirements for half-select programming; (d) Example of 3T NEM relay connecting two inverters during configuration.

The hold voltage (V_{hold}) on the row lines must be greater than V_{dd} by at least V_{po} (Fig. A.1c). However, CMOS transistors are not subject to this high voltage because V_{hold} is only applied to the gate electrode (which is isolated from the source and drain) of the NEM relay. CMOS transistors are connected to the column lines only during configuration (Fig. A.1d). Since the 3T NEM relay is driven by a CMOS buffer (Fig. A.1d), when the column line connected to the NEM relay is at V_s during configuration, the drain junction of the PMOS in the CMOS buffer (Fig. A.1d) will be forward biased (V_{supply} is 0 during FPGA configuration). This will cause current to flow through the column lines, causing voltage drops along the column lines. Higher V_s will lead to larger column line current, imposing constraints on the value of V_s (Fig. A.1c, requirement iv). Two possible solutions to overcome the forward biasing issue are: 1) set the body terminal of the PMOS to V_{dd} during configuration; 2) set V_{supply} to a small positive value to compensate for V_s during configuration.