2014 20th IEEE International Symposium on Asynchronous Circuits and Systems

Low Power Asynchronous VLSI with NEM Relays

Benjamin Z. Tang, Sunil A. Bhave, Rajit Manohar
School of Electrical and Computer Engineering
Cornell University. Ithaca, NY, 14853, U.S.A.
Email: bt48@cornell.edu, sunil@ece.cornell.edu, rajit@csl.cornell.edu

***Abstract*—CMOS technology scaling has reached a point where the circuit's static power is as high as the dynamic power. While further process scaling will only worsen leakage in transistors, it will benefit NEM relay technology. As asynchronous circuit design helps with dynamic power and NEM relays with static power, the use of NEM relays in asynchronous VLSI is ideal for low-power applications. In this paper, we present ways of combining both asynchronous and NEMS technologies and compare them with their CMOS counterpart. NEM relays can effectively implement not only QDI designs, but also bundled-data and power-gated circuits. We show in simulation that a 64-bit C-element, 32-bit PCHB AND and 8-bit PCHB adder implemented with NEM relays can achieve over 16X, 25X and 1.7X better energy-efficiency respectively compared to CMOS in a 90nm technology.**

***Index Terms*—QDI, Bundled-data, asynchronous MEMS, NEMS, Relays, nanoswitch**

I. Introduction

A. Motivation

Over the past several decades, CMOS technology scaling has been the dominant driver in improving energy efficiency of transistors. As the voltage supply is decreased, dynamic energy consumption decreases. Since the scaling of voltage supply results in reduced source-drain currents, the threshold voltage needs to be scaled accordingly as well to avoid any performance penalty. However, decreasing the threshold voltage inevitably increases leakage currents and we have already reached a point where CMOS technology scaling is no longer sufficient to reduce power consumption.

In the era of mobile and wearable computing, power consumption of electronics is a pressing challenge in designing increasingly smaller battery-operated devices. New methods in architecture, circuit design, devices and materials must be adopted to continue lowering power consumption. More parallel and less high-performance circuit architectures are emphasized, resulting in a shift to multicore computing within the last decade. In the area of circuit design, asynchronous circuits have been shown to enable very low power designs. This is because asynchronous subsystems run only as fast as they need to run without clocking. They do not consume dynamic power when there is no computation to be done. However, since leakage in transistors is becoming worse in nanoscale technologies, asynchronous circuits, when idle, may consume much static power. Therefore, alternative device technologies must be explored to replace CMOS. One of the technologies with extensive on-going research is nano-electro-mechanical (NEM) relays. Because they have zero leakage and many other advantages which we will delve deeper in this paper, NEM relays show great potential for low power asynchronous circuits. With an asynchronous design, dynamic power is optimized through the circuits' self-timed properties. When combined with NEM relay technology, static power consumption is minimized as well.

B. Related Work

NEM relays have been shown to be a very energy-efficient alternative to CMOS. [1] demonstrates the use of four-terminal NEM relays in common VLSI building blocks such as logic gates, adders, and latches. In simulation, a 32-bit adder implemented using NEM relays has been shown to achieve 10X lower energy per operation at 0.5GOP compared to its optimized CMOS counterpart. While [1] uses relays with folded flexures, [2] uses cantilever structures and has also shown an order of magnitude improvement in adders, ADC and DAC designs compared to CMOS. Other variants of NEM relays have also been proposed. For example, [3] shows a laterally-actuated relay design and [4] illustrates the use of six-terminal relays to allow compact implementation of complex logic.

This paper builds upon the findings in [1] to create novel logic topologies that are effectively implemented using normally-open and normally-closed NEM relays. These topologies leverage the strengths of NEM relays to further enhance asynchronous circuit designs. In this paper, we introduce two ways of utilizing NEM relays for low power asynchronous designs. The first approach involves synthesizing asynchronous designs directly into NEM relays circuits. The second is a hybrid CMOS-NEM relay approach used for power gating. Section II of this paper provides an overview of asynchronous circuit design, and its strengths and weaknesses. We then discuss NEM relays, their novel topologies and the benefits they can bring to asynchronous designs in Section III. Section IV describes the use of NEM relays in asynchronous QDI with simulation results in Section V.

II. Asynchronous VLSI

Instead of using a global time reference to sequence computation in synchronous (clocked) circuits, asynchronous circuits use handshake protocols. Data tokens are passed from one pipeline stage to the next through handshakes. This data-flow driven behavior allows asynchronous systems to naturally support varying data rates and throughput to achieve average-case pipeline latency rather than worst-case latency seen

1522-8681/14 $31.00 © 2014 IEEE
DOI 10.1109/ASYNC.2014.19

in synchronous pipelines. Besides, as these handshakes are coordinated locally among communicating processes, asynchronous circuits do not face global clock distribution issues. The equivalent of perfect clock gating is also realized in asynchronous systems since dynamic power is only consumed when the circuit is actually processing data.

There are many variants of asynchronous handshake protocols that trade off design complexity, latency and robustness to timing variations. Among the different asynchronous handshakes and design styles, quasi-delay-insensitive (QDI) circuits have excellent self-timed properties and the least timing assumptions, making them the most robust against delay variations caused by circuit operating conditions and process variation. However, due to their use of dual-rail encoding, they have area and power overheads. Especially for wide datapaths, single-rail encoded bundled-data design may be preferred. The bundled-data design resembles synchronous circuits and can be synthesized using commercial synchronous tools. However, careful timing analysis is required for the bundled-data implementation to ensure that its delay-line matches or is longer than the worst case computation delay of the pipeline stage. Numerous other asynchronous design styles are summarized in [5]. Our paper focuses on QDI and bundled-data.

While asynchronous circuits enjoy many benefits due to its self-timed properties, their implementation in CMOS has limitations and introduces several weaknesses. Due to leakage through the transistors, all circuits, including those involved in handshakes and logic, consume static power when they are not actively switching. Besides, all logic gates have to remain driven as leakage can destroy their states over time; state-holding logic that is common in QDI circuits employs staticizers. Leakage will only get worse with further transistor technology scaling.

Secondly, pull-up and pull-down stacks must be kept small; there must not be too many series transistors in the stacks. Otherwise, the output will have a poor rise or fall time and the short circuit currents through the pull-up and pull-down stacks, which are both momentarily on, will be large. As a result, for circuits with a large fan-in, a tree structure is used. For example, completion trees are common in QDI circuits handling larger number of inputs or outputs.

Thirdly, production rules for QDI circuits have to be bubble-reshuffled. Bubble-reshuffling is a procedure that converts variables in the production rules to their correct inverted and non-inverted senses for them to be CMOS-implementable. This procedure has to be done because PMOS and NMOS do not have similar pull-up and pull-down capabilities; the weak pull-down nature of PMOS requires that all variables involved in the pull-up network be inverted. Conversely, the weak pull-up nature of NMOS necessitates that all variables used in the pull-down network be uninverted. Hence, bubble-reshuffling in CMOS adds circuit complexity and transistor count.

These weaknesses in CMOS implementation of asynchronous circuits can be overcome by NEM relays. The following section introduces the NEM relay technology and contrasts its strengths with the weaknesses of CMOS.

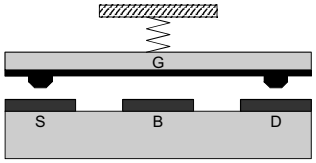

Fig. 1. Cross section view of a normally-open NEM relay

III. NEM Relays

A NEM relay in the normally-open configuration is typically a four-terminal switch that consists of gate, drain, source and body electrodes. The floating gate is suspended over the drain, source and body terminals through a spring in the form of a cantilever or folded flexures as shown in $Fig.$ 1. The relay utilizes electrostatic force to turn it on or off. When the applied voltage between the gate and the body (Vgb) reaches the threshold voltage also known as the pull-in voltage, the electrostatic force generated is strong enough to pull the floating gate structure towards the body, connecting the source and drain [1]. The relay is now switched on. To turn the relay off, Vgb is reduced so that the gate pulls away from the body. This happens when the spring's restoring force overcomes the decreasing electrostatic force. Another type of NEM relay is the normally-closed configuration. It uses the same basic electrostatic principle as the normally-open configuration except that the switch's default state is on. When Vgb exceeds the threshold voltage, the source and drain terminals are disconnected instead; the switch turns off.

Due to the physical nature in which NEM relays switch, they have several advantages over CMOS. First and foremost, NEM relays behave like perfect switches with zero leakage. They turn on or off abruptly when Vgb crosses the threshold voltage. There is no sub-threshold current flowing between the source and drain because the physical conduction path between them is broken when the gate switches. Hence NEM relays are useful in eliminating static power consumption, especially in asynchronous circuits where we can then truly claim that power is only consumed during useful computation. In contrast, leakage in CMOS is increasingly worse as technology scales. In fact, leakage power in modern CMOS technologies below 65nm is becoming a concern and asynchronous circuits need to address it with power-gating techniques [6].

Secondly, NEM relays have sharp voltage-transfer-characteristic (VTC) curves. Unlike CMOS, the steep rising and falling edges for NEM relays contribute to minimal short circuit currents during switching. More NEM relays can be stacked together in series in longer pull-up or pull-down network compared to CMOS transistors.

Thirdly, NEM relays are ambipolar by nature. Since electrostatic force is independent of the polarity of Vgb, as long as the voltage difference between the gate and body is larger than the threshold voltage, the relay switches. Hence, the same relay can be configured to behave like a PMOS or NMOS by connecting the body to Vdd or ground as shown in $Fig.$ 2. This also gives NEM relays both a strong pull-up capability

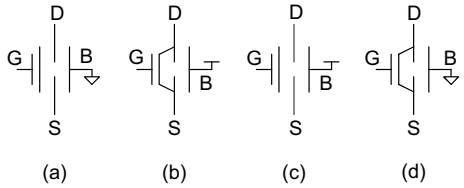

Fig. 2. (a) "NMOS" with normally-open relay (b) "NMOS" with normally-closed relay (c) "PMOS" with normally-open relay (d) "PMOS" with normally-closed relay

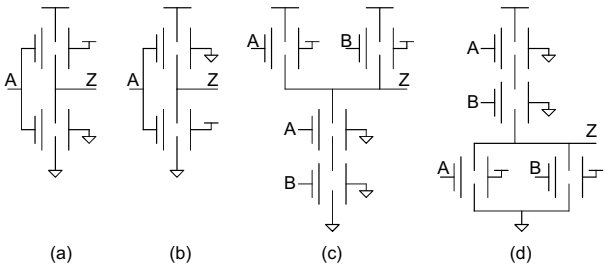

Fig. 3. Normally-open NEM relay implementation of (a) Inverter, (b) Buffer, (c) NAND, and (d) AND

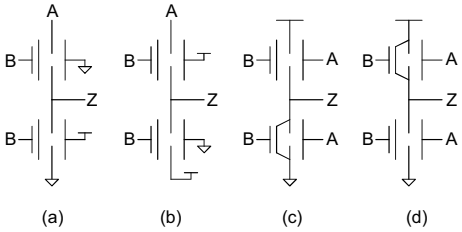

Fig. 4. Effective NEM relay implementations of (a) AND, (b) OR, (c) XOR, and (d) XNOR

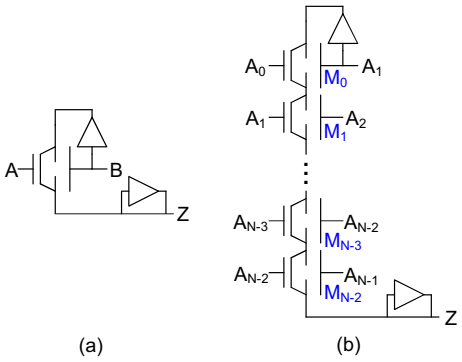

Fig. 5. NEM relay implementation of C-element for (a) 2 inputs and (b) N inputs

as well as pull-down. $Fig.$ 3(c) shows a static-CMOS-like implementation of a NAND gate. Because of the weak pull-down nature of PMOS and weak pull-up nature of NMOS, an AND gate has to be implemented from a NAND coupled with an inverter. NEM relays on the other hand can implement an AND gate directly without an inverter as shown in $Fig.$ 3(d). A buffer can also be implemented directly in $Fig.$ 3(b) without the need for two inverters. Although normally-open NEM relays are shown, these gates can similarly be implemented with normally-closed relays by swapping Vdd and ground at the body terminals. The ability of NEM relays to implement both inverting and non-inverting logic is especially useful for QDI asynchronous circuits where the logic can be synthesized directly without bubble-reshuffling. Moreover, staticizers for state-holding logic that are common in QDI circuits can be implemented with just a NEM relay buffer.

On top of that, NEM relays give circuit designers plenty of flexibility to reduce device count. Pass transistor logic implemented with NEM relays is especially attractive. Not only is device count reduced, the relays can pass both a strong '1' and '0'. $Fig.$ 4(a) and (b) shows that the AND and OR gates respectively can be built from just two relays. The Vgb-induced electrostatic switching principle in NEM relays is also ideal for XOR and XNOR gates. They can be implemented with just a normally-open relay and a normally-closed relay as shown in $Fig.$ 4(c) and (d). To further reduce device count, these combinational logic gates can be built using ratioed logic topologies. For example, an XOR gate can be implemented with just a normally-open relay and a resistive load that replaces the pull-down relay in $Fig.$ 4(c). The resistive load however must have resistance much greater than the on-resistance of the relay for the output to have fuller voltage swing. The trade-offs for using the ratioed logic topology of course include static power dissipation and lower voltage swing.

Another logic gate that can be effectively implemented using NEM relays is the C-element, a very common asynchronous circuit component. In CMOS, the simplest implementation of a 2-input C-element including its staticizer uses 8 transistors. On the other hand, 5 NEM relays are needed for a C-element as shown in $Fig.$ 5(a). Unlike CMOS, an output inverter is not needed for NEM relays because non-inverting logic can be implemented directly. A buffer is used as a staticizer for the output. Another buffer is used to avoid having a self-driving gate by one of the inputs so that the relay switching delay is taken into account. Note that although NEM relays have zero leakage, state-holding logic may still employ a staticizer to prevent circuit noise from corrupting its state through parasitic capacitive coupling. More detailed analysis on the C-element is provided in the next section.

In short, NEM relays have many advantages over CMOS and are very attractive for asynchronous circuit design. They not only consume zero static power and have sharp rising and falling edges, but also can implement inverting and non-inverting logic and full-swing pass gate topologies effectively with minimal number of relays. However, anything that has mechanical moving parts has a limited lifetime. NEM relays' electrodes are subjected to wear and tear during switching but studies have shown that the relays are still functional over 60 billion cycles [1]. Besides that, NEM relays are slow compared to CMOS. Their slowness comes not from electrical

delay but mechanical delay. Electrical delay from junction and parasitic capacitance in a NEM relay is negligible, in the order of picoseconds, and thus contributes to its sharp VTC curve. Mechanical switching delay in contrast is in the order of 10s of nanoseconds in a 90nm technology [1]. The art of balancing mechanical and electrical delays is key to exploiting NEM relays for higher performance VLSI applications. Large complex logic is remapped into circuit topologies that use less relays and facilitate simultaneous mechanical switching [2]. This technique not only favors synchronous circuit modules in the critical path but also the bundled-data design in asynchronous circuits. For mobile applications that emphasize low-power operation rather than high performance, more distributed, parallelized execution of logic functions without a global clock constraint is preferred. This allows the circuit speed to change dynamically to achieve data-flow driven, average-case performance that is one of the salient qualities of asynchronous QDI circuits. The next section delves into QDI synthesis and analysis with NEM relays.

IV. QDI WITH NEM RELAYS

A. Synthesis with NEM Relays

Numerous NEM relay configurations afford us with circuit design flexibility. First of all, we can choose to use normally-closed or normally-open NEM relays. Secondly, the relays can be configured to be active high or low by connecting its body terminal to Vdd or ground. Like in CMOS QDI, production rules are derived from handshaking expansions. These production rules form the pull-up and pull-down network to drive a signal. However, unlike CMOS, the handshaking expansions do not have to be bubble-reshuffled for NEM relays since they can implement non-inverting logic directly. For example, referring to the NOR gate in $Fig.\ 6$, to turn on a normally-open NEM relay when a variable is asserted, the body terminal of the relay is grounded. As such when the variable is asserted, there exists a voltage difference between the gate and body terminals to electrostatically actuate the relay. This is seen in the pull-down network of the L.e signal. Conversely, if a relay is to be turned on when a variable is de-asserted, the body terminal of the relay is connected to Vdd. This is seen in the pull-up network of L.e.

Therefore, non-bubble-reshuffled production rules can be implemented efficiently using NEM relays in the static CMOS-style. Existing QDI synthesis tools can be easily modified to accommodate NEM relays. Implementation of QDI logic with pass gates style is also possible but requires additional thought and analysis for synthesizing complicated logic. An example in the form of an adder is provided in the example applications section later in this paper.

One of the weaknesses of QDI asynchronous circuits is completion overhead. The completion latency grows with the datapath width of N bits, and is proportional to logN for tree structures. One way of reducing this overhead is by using single-track communication protocols seen in [7], [8], [9]. Unfortunately, some timing assumptions and careful timing analysis need to be made for these types of circuits. Another way of reducing completion overhead while still preserving timing robustness of QDI is through the use of pipelined completion [10]. This is done with more localized completion detection at the level of a single or several bits rather than a giant completion tree for all bits. Consequently, the completion detection is no longer on the critical path. While we can hide the completion latency behind actual logic computation delay, we still cannot run away from detecting completion with the standard OR gates and C-element. Therefore, an efficient method for implementing these gates is still critical.

$Fig.\ 5$ shows how the C-element can be implemented effectively using NEM relays. Overlapping pairs of inputs are compared against each other and whenever any input differs from its pair, the normally-closed relays turn off, allowing the staticizer to hold its output state. Since NEM relays are used here as full-swing pass gates, we can daisy-chain many of them in series to accommodate completion detection of a wide datapath. Even for a 2-input C-element, the NEM relay implementation only uses almost half as many devices compared to CMOS. Moreover, for every additional input, we only need one additional relay whereas CMOS needs two. Not only that, C-elements implemented in CMOS have to resort to using a tree structure to limit the number of transistors stacked in series to avoid slow circuits, large slew rates and large short circuit currents during switching. The number of relays needed for an N-input C-element is $(N-1)+4$ where 4 is the constant overhead due to the relays in the drive buffer and staticizer. As for CMOS, using 2-input C-elements arranged in a tree structure, the number of transistors increase much faster with respect to the number of inputs, to the tune of 8N. For a 64-bit data path, the C-element implementation with NEM relays takes only 67 devices compared to a whopping 512 transistors in CMOS. Even with a more optimized tree structure comprising 2-input and 4-input C-elements, we still need 252 transistors for 64 bits.

B. QDI Templates using NEM relays

The Weak-Conditioned Half Buffer (WCHB) template is fast and is popular for implementing buffers. In CMOS, the WCHB has a forward latency of 2 transitions and a cycle time of 10 transitions. On the other hand, WCHB implemented with NEM relays have a forward latency of 1 transition and a cycle time of 4 transitions. The ability of NEM relays to implement non-inverting logic directly allows it to achieve the minimal possible number of transitions. $Fig.\ 6$ shows a NEM relay implementation of the WCHB with 1-bit input (L) and 1-bit output (R).

The Pre-Charge Half Buffer (PCHB) is another popular QDI template. It is commonly used for buffered logic. With CMOS implementation, the PCHB has a forward latency of 2 transitions and a cycle time of 14 transitions. In contrast, the NEM relay version, again, achieves the minimal possible number of transitions, with a forward latency of 1 transition and a cycle time of 4 transitions. $Fig.\ 7$ shows a NEM relay example of the PCHB with 2-bit inputs (L0, L1) and a 1-bit output (R). Logic can be easily packed into the pull-up

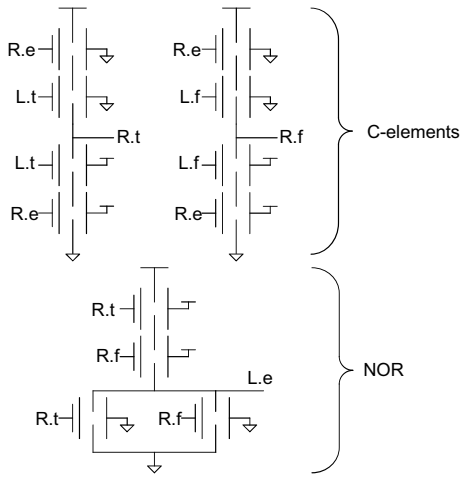

Fig. 6. WCHB with 1 bit input (L) and output (R) using NEM relays, staticizers omitted

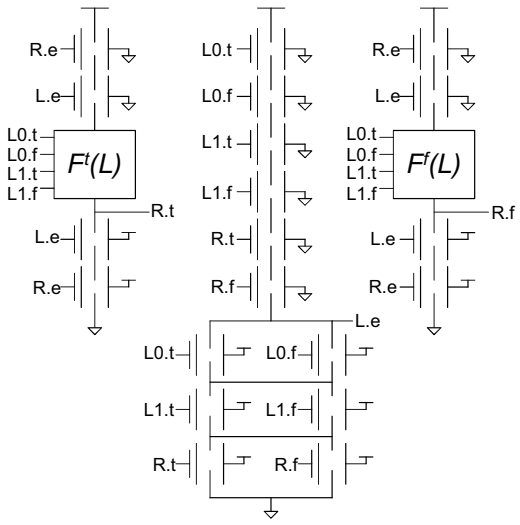

Fig. 7. PCHB 2-bit input and 1-bit output using NEM relays, staticizers omitted

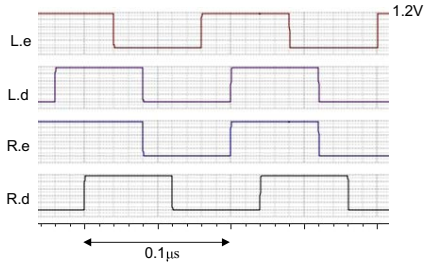

Fig. 8. Simulation of PCHB dataless channel with NEM relays

stacks of the output data rails. Besides that, the input and output completion detection circuit essentially generates the input acknowledge signal (L.e). For a large number of inputs and outputs, such as in buffered merge, split or PCHB copy modules, the pull-up stack for the input acknowledge (L.e) can become quite long. In CMOS implementation, these long transistor stacks need to be broken down into stages, resulting in increased number of gate transitions and transistor count. NEM relays however do not face such limitations; stacks can be long and still be optimal providing the electrical delay through the stack is smaller than the mechanical delay. In fact, as observed in simulations in a 90nm technology, stacks can be as long as over 200 relays in series. To extend beyond this number, one approach is to reduce device count with the novel C-element topology introduced in $Fig.\ 5$. This adds an extra transition to the overall cycle time for dataless channels or 2 transitions for dualrail channels since OR gates are needed too.

It is important to note that although QDI templates implemented with NEM relays exercise much fewer transitions in a cycle compared to CMOS, each transition in current NEM relay technology may be an order of magnitude slower than CMOS due to mechanical delays. This can be seen in $Fig.\ 8$ taken from a scaled 90nm NEM relay technology simulation of the PCHB with dataless input (L) and output (R) channels. The good news is that further relay technology scaling will improve mechanical delays. Besides that, the same NEM relay synthesis and design techniques introduced here are also applicable to other QDI templates to give us the smallest possible number of transitions in a cycle.

Studies in [1] and our power simulation results in the final section of the paper show that NEM relays are indeed very energy-efficient. The main drawback is the device latency due to mechanical delays. Hence there is an obvious trade-off between energy and latency when comparing NEM relays and CMOS. However, while NEM relay technology can still scale further to improve both its energy and latency, CMOS technology cannot do so without significantly increasing leakage energy. Hence a hybrid CMOS and NEM relays approach can be used to balance the strengths and weaknesses in both technologies. High performance sub-systems can continue to use CMOS but power-gated with NEM relays [11] whereas slower sub-systems with operation frequencies below 100 MHz can use NEM relays. In certain applications, NEM relays have the potential to fully replace CMOS. The following section highlights some example applications of asynchronous NEMS.

C. Example Applications

Most baseband processing applications involve signal bandwidths below 100MHz. For example, the bandwidth of the GPS L1 C/A civil signal is about 2MHz. Baseband processing must be capable of handling at least the Nyquist rate of the signal. So even though the latency in NEM relays tends to limit their use below 100MHz, they are fast enough for many baseband processing applications. A data-flow driven architecture can be realized using asynchronous circuits. Such architecture can exploit data-dependent behavior and is able to process the signals through different frequency domains

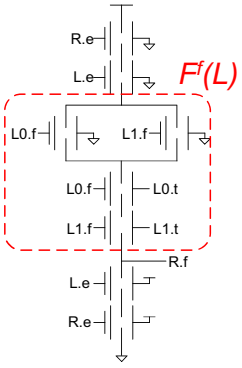

Fig. 9. Optimized output false rail of a 2-input PCHB AND using NEM relays

efficiently [12]. Baseband processing has a typical serial-to-parallel data conversion structure where the data bit(s) from an RF frontend is fed serially into a baseband processor and then accumulated or correlated over time into a larger number. The data throughput is higher near the frontend but the datapaths towards the backend are wider where more parallel multi-bit processing is required. With focus on such data-flow driven baseband processing design, we provide some specific asynchronous circuit implementations using NEM relays.

1) PCHB AND: We use AND as an example of simple logic functions that can easily be incorporated into the PCHB template. For bitwise AND, as the number of bits increases, the output data rails' circuit structure remains the same but the input acknowledge's (L.e) circuit as shown in *Fig.* 7 becomes more complex. The number of transistors in series in the pull-up stack of L.e will be 6N and its pull-down stack 3N, where N is the number of bits. As mentioned earlier, CMOS implementation will need a completion tree to break down long stacks whereas NEM relays need not. NEM relays can implement long pull-up and pull-down stacks, which not only reduces device count, but also power and cycle time.

Implementing an N-input AND function is slightly more complicated but allows us to emphasize the flexibility of NEM relays over CMOS. For a 2-bit AND, the logic function in the output true rail will be L0.t∧L1.t whereas that of the output false rail will be (L0.f∧L1.f ∨ L0.f∧L1.t ∨ L0.t∧L1.f). As the number of bits increases, the logic function in the output false rail increases exponentially. Hence, for a large number of inputs, the AND function is usually implemented by cascading smaller function blocks in a tree structure. With the NEM relays, the output false rail logic function can be greatly simplified. We can check the validity of the input rails directly by using the body and gate terminals of the NEM relays as shown in *Fig.* 9.

2) PCHB Single-Mechanical Delay Adder: The ability to add large numbers while satisfying throughput requirements in baseband processing is crucial. Due to the slow mechanical delays in relays, the adder carry chain implemented naively can severely impact the system performance. [1] has shown that an adder can be efficiently implemented using pass-gates

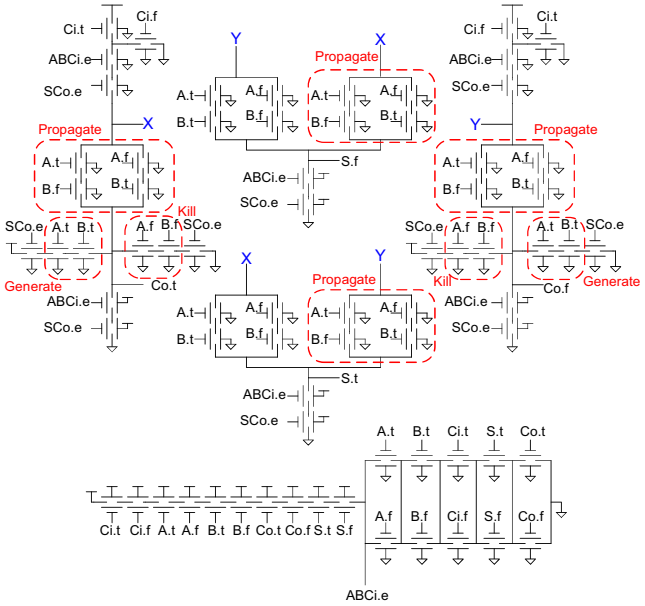

Fig. 10. PCHB full adder with pass-gate logic topology for single mechanical delay through carry chain

topology to allow for only one mechanical delay through the adder's carry chain. Bundled-data can essentially use the same technique and will give us similar advantage over CMOS, which is 10X better energy efficiency at frequencies below 100MHz for a 32-bit adder. As for the bundled-data's delay line, we can easily account for the circuit delay, which is one mechanical delay with some margin for parasitics.

Implementing something similar with PCHB requires a few tweaks for dualrail compatibility. The PCHB logic function is modified to enable a single mechanical delay through the carry chain. These logic functions are shown in red blocks in *Fig.* 10 for the carry out (Co) and sum (S) outputs. The generate, propagate and kill circuits resemble those from [1]. A simulation of this adder can be seen in *Fig.* 11. When cascading full adder cells, the carry out's true rail and false rail are sent to the subsequent full adder cell's X and Y nodes respectively. Like in regular PCHB template, the input acknowledge signal (ABCi.e) is generated from the completion detection circuit of the input and output data rails.

3) Asynchronous Power Gating: NEM relays have zero leakage and hence are more effective than transistors. Besides that, power-gating transistors have to be big in order to provide enough current for the circuit. On the other hand, NEM relays have good current-handling capability and have no penalty to the chip area if they are fabricated in the backend metallization layers [11]. The power-gating benefits that NEM relays bring can be used towards reducing static power in hybrid CMOS and NEMS asynchronous circuits. Power gating of asynchronous circuits is attractive because of Zero-Delay Ripple Turn On (ZDRTO) [6]. In synchronous circuits, upon system wake-up, processing can only begin after stabilization of the voltage supply so that system timing requirements are

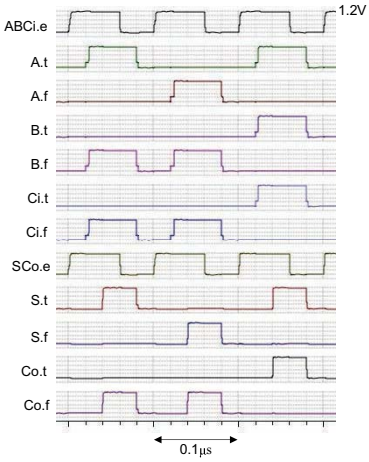

Fig. 11. Simulation of PCHB full adder with NEMS pass gates topology

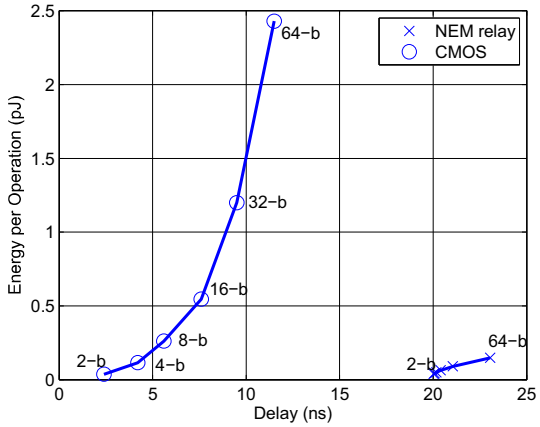

Fig. 12. Energy per operation versus delay comparison for C-element with different number of input bits, implemented with NEM relays and CMOS to drive a 25fF load

met. In asynchronous systems, ZDRTO kickstarts computation just as the pipelines are waking up. This allows asynchronous systems to be powered down when idle and woken up efficiently to process incoming data. The combination of NEM relay power-gating and ZDRTO in asynchronous circuits is ideal for power-cycling baseband processing modules towards the backend which can be idle for milliseconds or more when waiting for correlation data from modules near the frontend [12].

V. Results

Simulations of the NEM relays circuit topologies in this paper were done in Cadence using a Verilog-A model for the NEM relays, with Vdd=1.2V. The model parameters are scaled for the 90nm technology node and are provided in [1]. Normally-closed NEM relays are also simulated with the same model parameters except with the opposite turn on/off condition at the threshold voltage.

We analyzed the energy per operation and the delay of the C-elements for different number of inputs. A 25fF load was used to simulate wire and loading from other NEM relays or CMOS transistors. The NEM relay version uses the topology shown in $Fig.$ 5 and were minimum sized. The CMOS version uses a tree topology of cascaded 2-input C-elements, have average transistor size of 2.5X the minimum gate length, and were SPICE-simulated in a 90nm technology (Vdd=1.2V). $Fig.$ 12 compares the energy per operation and delay of the C-elements with the number of inputs ranging from 2 to 64. As the number of inputs increases, capacitive loading and on-resistance from NEM relays stacked together in series increase. Consequently, the electrical RC delay increases fairly linearly with the number of inputs. However, the NEM relay mechanical delay at $\sim$20ns dominates; in fact, even for a 64-input C-element, the electrical delay is still about one-eighth of the mechanical delay. Hence the method of having long stacks of NEM relays is preferred so that we only incur a single mechanical delay. In contrast, CMOS is over 4 times faster than NEM relays for fewer bits but about 2 times faster for 64 bits. This is not only helped by faster switching in transistors but also the logarithmic increase in the number of stages and latency in the CMOS C-element tree structure, with respect to the number of inputs. Though NEM relays lose out in terms of latency, they shine in energy per operation. For a 64-bit C-element, NEM relays have over 16X lower energy per operation compared to CMOS. For a fairer comparison at 10MHz where CMOS voltage is lowered to 0.79V, NEM relays are still better with 13X less energy. This is because NEM relays have zero leakage and minimal short circuit currents during switching. Furthermore, much fewer relays are needed to build C-elements, almost an order of magnitude fewer for 64 bits compared to CMOS as discussed earlier.

We also compared the PCHB AND function with different number of inputs for CMOS and NEM relays. All signals interfacing with the environment are buffered accordingly with transistors or NEM relays. To accommodate the slowness of NEM relays implementing four-phase handshakes, we simulated the circuit at 10MHz operation frequency, which is practical for asynchronous baseband processing of any signal with bandwidth below 5MHz, including GPS. The CMOS version is constructed by cascading 2-input PCHB AND modules in a binary tree structure whereas the NEM relay version implements the optimized AND function of all inputs in a single module as shown in $Fig.$ 7 and $Fig.$ 9. The NEM relays are minimum sized while the average transistor size is 4.5X the minimum length as some transistors have to be strong enough to overcome the staticizers. The simulation result in $Fig.$ 13 shows that as the number of inputs increases, the average energy consumed by the CMOS version rises much more rapidly than the NEM relay version, although they start off having comparable energy at 2-bits. In fact, for a 32-bit AND, the NEM relay implementation consumes about 25X ($\sim$10X if CMOS uses Vdd=0.78V) lower energy compared to CMOS. The energy savings comes from not only the relays'

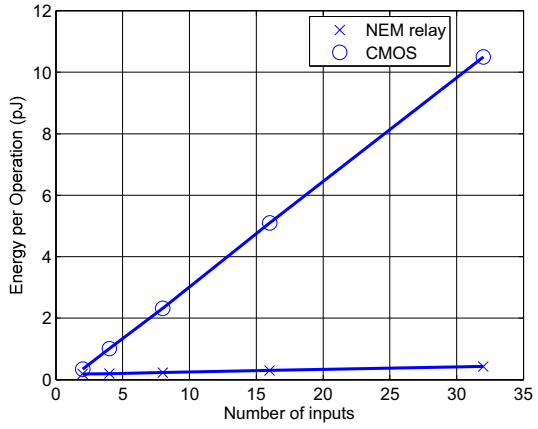

Fig. 13. Average energy per operation at 10MHz for PCHB AND with different number of inputs, implemented with NEM relays and CMOS with 25fF output loads

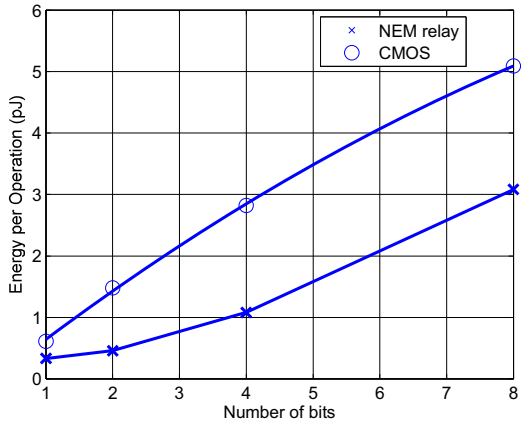

Fig. 14. Average energy per operation at 10MHz for PCHB adder, with different bit widths, implemented with NEM relays and CMOS with 25fF output loads

low switching energy but also logic simplification and compact implementation without bubble-reshuffling or cascading. For the 32-bit PCHB AND, we need almost an order of magnitude fewer relays to build compared to transistors.

We also performed similar tests on the PCHB full adder. In the CMOS version, adders with more bits are implemented with a cascaded full adder topology. The NEM relay version is implemented with the single-mechanical delay, pass-gates topology shown in $Fig.$ 10. For an 8-bit adder, the number of relays needed is about half the number of transistors needed for the CMOS version. The savings in device count is also roughly reflected in the $\sim$1.7X reduction in average energy shown in $Fig.$ 14.

VI. Conclusion

NEM relays have zero leakage, sharp VTC curves and are ambipolar. We demonstrated ways to exploit these properties to further enhance asynchronous circuit design. As asynchronous design helps with dynamic power, NEM relays help with static power. In addition, the flexibility of NEM relay implementations allows us to further reduce circuit complexity, device count and thus dynamic power as well. We introduced examples in QDI, bundled-data and asynchronous power-gating designs where NEM relays are a compelling choice for low power, moderate-speed applications. Our simulations of a 64-bit C-element, 32-bit PCHB AND and 8-bit PCHB adder show that NEM relays can achieve over 16X, 25X and 1.7X better energy-efficiency respectively compared to CMOS in a 90nm technology. The slow mechanical delays in NEM relays can be improved through technology scaling and circuit design. The latter emphasizes designing circuits with simultaneous mechanical switching, with compact logic and with reduced transitions such as with the use of two-phase handshake protocols.

References

[1] F. Chen, M. Spencer, R. Nathanael, C. Wang, H. Fariborzi, A. Gupta, H. Kam, V. Pott, J. Jeon, T.-J. K. Liu, D. Markovic, V. Stojanovic, and E. Alon, "Demonstration of Integrated Micro-Electro-Mechanical Switch Circuits for VLSI Applications," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, 2010, pp. 150–151.

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

[3] K. Amponsah, N. Yoshimizu, S. Ardanuc, and A. Lal, "Near-kT Switching-Energy Lateral NEMS Switch," in *Nano/Micro Engineered and Molecular Systems (NEMS), 2010 5th IEEE International Conference on*, 2010, pp. 985–988.

[4] D. Lee, W. Lee, C. Chen, F. Fallah, J. Provine, S. Chong, J. Watkins, R. Howe, H.-S. Wong, and S. Mitra, "Combinational Logic Design Using Six-Terminal NEM Relays," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 5, pp. 653–666, 2013.

[5] S. Nowick and M. Singh, "High-Performance Asynchronous Pipelines: An Overview," *Design Test of Computers, IEEE*, vol. 28, no. 5, pp. 8–22, 2011.

[6] C. Ortega, J. Tse, and R. Manohar, "Static Power Reduction Techniques for Asynchronous Circuits," in *Asynchronous Circuits and Systems (ASYNC), 2010 IEEE Symposium on*, 2010, pp. 52–61.

[7] B. R. Sheikh and R. Manohar, "Energy-Efficient Pipeline Templates for High-Performance Asynchronous Circuits," *J. Emerg. Technol. Comput. Syst.*, vol. 7, no. 4, pp. 19:1–19:26, Dec. 2011.

[8] I. Sutherland and S. Fairbanks, "GasP: A Minimal FIFO Control," in *Proceedings of the 7th International Symposium on Asynchronous Circuits and Systems*, ser. ASYNC '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 46–.

[9] M. Ferretti and P. Beerel, "Single-Track Asynchronous Pipeline Templates using 1-of-N Encoding," in *Design, Automation and Test in Europe Conference and Exhibition, 2002. Proceedings*, 2002, pp. 1008–1015.

[10] A. Martin, A. Lines, R. Manohar, M. Nystrom, P. Penzes, R. Southworth, U. Cummings, and T. K. Lee, "The Design of An Asynchronous MIPS R3000 Microprocessor," in *Advanced Research in VLSI, 1997. Proceedings., Seventeenth Conference on*, 1997, pp. 164–181.

[11] H. Fariborzi, M. Spencer, V. Karkare, J. Jeon, R. Nathanael, C. Wang, F. Chen, H. Kam, V. Pott, T.-J. K. Liu, E. Alon, V. Stojanovic, and D. Markovic, "Analysis and Demonstration of MEM-relay Power Gating," in *Custom Integrated Circuits Conference (CICC), 2010 IEEE*, 2010, pp. 1–4.

[12] B. Tang, S. Longfield, S. Bhave, and R. Manohar, "A Low Power Asynchronous GPS Baseband Processor," in *Asynchronous Circuits and Systems (ASYNC), 2012 18th IEEE International Symposium on*, 2012, pp. 33–40.