Muller C-Element for NEMS

Philipp Lehninger*, Axel Jantsch*, Andreas Steininger†, Elliott Worsey‡, Victor Marot‡, and Dinesh Pamunuwa‡
*Institute of Computer Technology, TU Wien, Vienna, Austria
†Institute of Computer Engineering, TU Wien, Vienna, Austria
‡School of Electrical, Electronic and Mechanical Engineering, University of Bristol, Bristol, U.K.

***Abstract*—Nano-electromechanical switches (NEMS) are a promising chip technology for harsh environments. However, their use with conventional synchronous designs may face hurdles, as wear-out limits their number of switching cycles. In this paper we propose a novel implementation of a Muller C-Element for NEMS devices and present two case studies showing that Quasi Delay Insensitive (QDI) design can significantly decrease the number of switching cycles and thus extend the lifetime of the circuit, potentially even without an increase of the device count. Our results indicate that the benefits of QDI become more pronounced for circuits that have a high proportion of flip-flops, and in cases where slow input causes idle clock cycles. Overall, our approach of using a customized Muller C-Element in a QDI design style shows clear benefits for the considered NEMS technology.**

***Index Terms*—Nanoelectromechanical (NEM), Muller C-Element (MCE), Quasi Delay Insensitive (QDI)**

I. Introduction

In harsh environments with temperatures $> 150\,^{\circ}\mathrm{C}$ or radiation levels $> 100\,\mathrm{krad}$, common CMOS technology may not operate correctly. For such environments, a nanoelectromechanical technology (NEMS) has been developed [3, 7], which is based on mechanical relays to realize switching and storing functionality. Just like CMOS, it can implement any logic gate or memory cell. Moreover, it is based on CMOS compatible manufacturing processes which allow it to be used alongside CMOS circuits. Although still immature, it is a promising technology that is not only resilient to high temperatures and radiation but also energy efficient due to near-zero leakage current, which makes it very attractive for edge nodes that have long standby times.

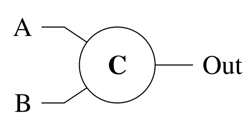
However, NEMS devices are generally larger and slower than CMOS transistors, as can be seen from preliminary figures and estimates for switch sizes and latency stated in section I-B. Furthermore, the device count in NEMS also has a stronger impact on the manufacturing yield than in CMOS, where the yield mostly depends on the chip area. The moving beam of the switches could be broken or unintentionally connected to the surrounding structure, leading to stuck-open or stuck-closed faults. Also, being mechanical, NEM switches wear out quickly when active. Their lifetime is currently limited to several thousand switching cycles. The technology is still in its early development and as it matures lifetime will significantly increase, while the latency and footprint will decrease. It can be expected, however, that mechanical wear and tear will continue to be a limiting factor.

This is only the accepted version, the final version is available at IEEE. This work was supported by the i-EDGE project, which has received funding from the European Union (grant number 101092018), the Swiss State Secretariat for Education, Research and Innovation (SERI) and UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee (grant numbers 10061130 and 10063023).

Therefore we propose to use asynchronous design techniques, as these, in contrast to the strictly periodic switching in synchronous circuits, produce activity only when needed. One of the basic building blocks in asynchronous design is the Muller C-element (MCE). As NEM switches are bigger than state-of-the-art CMOS transistors and designs with a low device count have higher yield, an optimized MCE implementation is necessary. We show how the unique switch properties of NEM relays can be utilised to realise these circuit efficiencies.

A. C-Element

The Muller C-element, introduced by David E. Muller [2], reflects the state of its inputs at the output, if they match. For non-matching inputs the output keeps its previous value – which clearly requires an internal state. The most basic implementation has two inputs A and B and one output. Symbol and truth table are depicted in figure 1.

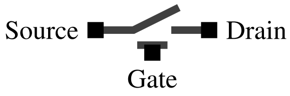


A	B	C_{N+1}
0	0	0
0	1	C_N
1	0	C_N
1	1	1

Fig. 1: Muller C-element symbol and truth table

B. NEM switches

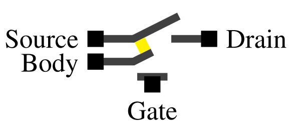
Our designs are based on a cell library that contains three types of NEM switches with 3 [6], 4 [8], and 7 terminals [7] respectively. Further details of their logical behavior can be found in [5, 10]. The simplest is the 3-Terminal switch (3-T) which can be used like a PMOS or NMOS transistor (figure 2), allowing for a CMOS-like circuit structure consisting of a p- and n-stack. The truth table shows that the switch is closed when the voltage difference between the gate and the source exceeds a certain threshold, regardless of the polarity.



Gate	Source	Drain
0	0	Z
0	1	1
1	0	0
1	1	Z

Fig. 2: 3-T device symbol and truth table

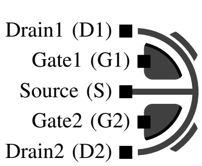
The 4-T device has a similar device architecture, except that the beam has two parts that are connected mechanically, but isolated electrically, which introduces an extra terminal. Figure 3 shows the respective device symbol and truth table. Unlike the 3-T device, the voltage between gate and body now determines whether the switch is closed, and the source terminal does not affect the state of the switch. As the signal that controls whether the switch is on or off is decoupled from the data signal, the 4-T device allows for highly device-efficient and thus area-efficient circuit implementations. For example, A 2:1 multiplexer can be built using just two 4-T devices.



Gate	Body	Source	Drain
0	0	x	Z
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	x	Z

Fig. 3: 4-T device symbol and truth table

Figure 4 shows a symbol of the 7-Terminal device, which has two inner and two outer gates, two drain terminals and a source. The outer gate pairs are not labelled as they are always driven in tandem with a diagonally situated inner gate [7]. The symbol of the 7-T device resembles the physical appearance of the device: The moving beam of the switch has an arcuate and straight segment and is physically connected to the source terminal. To move the beam towards one of the drain terminals, the voltage between the source and the respective gate has to exceed the pull-in threshold, whereas the voltage between the source and the other gate must not exceed the threshold. This device is designed to keep its state without power, through utilising surface adhesion forces between at the tip contact. Thus, when the device has been switched to connect to one of the drains, D1 or D2, it keeps the connection until the conditions for switching to the opposite drain become valid.



POS_N	S	D1	D2
neutral	0	Z	Z
neutral	1	Z	Z
left	0	0	Z
left	1	1	Z
right	0	Z	0
right	1	Z	1

Fig. 4: 7-T device symbol and output truth table

Therefore, the previous state of this switch is noted in the state truth table (Table I). Initially, it is in a neutral state, without connecting to either of the drain terminals. After the initial switching, the device will not return to the neutral state again. Thus, in operation, the source will be connected to one of the drains as shown in Figure 4.

TABLE I: 7-T device state

G1	S	G2	POS_{N+1}
0	X	0	POS_N
1	X	1	POS_N
0	0	1	right
1	1	0	right
0	1	1	left
1	0	0	left

Table II shows current dimensions of NEM switch designs. The turn-on mechanical delay is estimated to be around 450 ns for 3-T and 4-T devices and 350 ns for the 7-T device. Pull out is as nearly instant as can be measured when the controlling voltage reaches the pull-out voltage, and thus the turn-off delay is determined by the slew rate of the waveform.

TABLE II: NEM device dimensions (outline)

Device	W in µm	H in µm
3-T	35.4	13.5
4-T	43.4	13.5
7-T	31.2	24

II. Related Work

The most common CMOS MCE implementations are compared in [9]. Both the conventional as well as the symmetric MCEs shown in that paper require a total of 12 transistors, while the weak feedback implementation has only 8 transistors but has a transient driver conflict when switching, which leaves this choice rather unattractive, considering that low power consumption is meant to be one of the advantages of NEMS .

Another interesting MCE design is presented in [4], shown in Figure 5. Although it has a less efficient CMOS implementation than other circuits, it was suggested for FPGAs as an alternative to an AND-OR implementation with a 3-input LUT, to take advantage of the circuit primitives available in FPGAs. As NEM technology offers area-efficient multiplexers, it can be realized in NEMS with only 6 4-T switches.

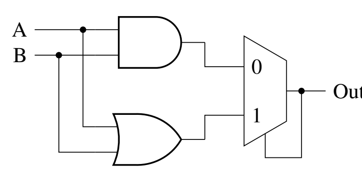


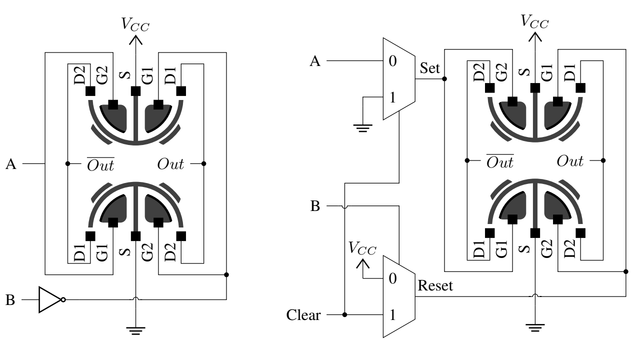
Fig. 5: Muller-C-gate with MUX [4]

III. Novel 7-T Approach

A. Basic Gate Design

In principle, a NEMS implementation can be derived from any of the existing transistor implementations by using the 3-T device as a replacement for a transistor. However, the 4-T and 7-T switches have more functionality than a transistor, so a C-element implementation with significantly fewer switches is feasible. Figure 6a shows a schematic of the proposed C-element which uses two 7-T devices as its core. To use the non-volatile 7-T device, the design must be able to cope with the fact that at any time at least one of both drain terminals will be floating ('Z'). Therefore, two complementary 7-T devices are used in a CMOS-like p-stack/n-stack configuration. Whenever one drain terminal is floating, the voltage of that net should be defined by the other 7-T device. The source of the 7-T device at the top is connected to VCC, so it is responsible for providing the voltage level for '1's at the output. According to the device state table I, the switch only moves when its gate terminals have a different logic level, otherwise it will store its previous state. In fact, the complementary 7-Ts form an SR-latch [10], that is why the two gate inputs can as well be called Set and Reset. To get the desired behavior of a C-element, one of the gates has to be equipped with an inverted input, $\overline{A}$ or $\overline{B}$, such that the switch changes its state when both inputs are the same (if it is already in the desired state, the beam of the switch will not move, it is not a toggling device). As the source terminal of the 7-T device at the bottom is connected to GND, its beam will always move in the opposite direction to the beam of the top 7-T device.

The initial neutral state of the 7-T devices has a floating output ('Z') on both drain terminals, but in most practical applications the inputs of a C-element would be both '0' at the beginning of the circuit operation, so the 7-T device leaves the neutral state immediately.



(a) basic design

(b) clearable version

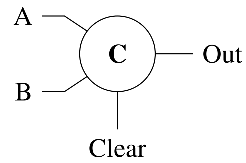
Fig. 6: 2-input Muller C-element with 7-T devices

B. C-element with Clear Input

Some circuits require C-elements with additional functionality such as clear or preset inputs. Figure 7 shows the adapted symbol and a truth table of a C-element with clear input. The schematic is depicted in figure 6b.

Clearing the 7-T C-element requires the following:

- The beam of the lower 7-T device must move to the right position. This produces a '0' at the non-inverting output Out via $D2$, and makes the drain $D1$ on the side of the inverting output $\overline{Out}$ float.
- The beam of the upper 7-T device must move to the right, too. This establishes a floating drain ($D1$) on the side of the non-inverted output, but a '1' on the inverted output (via $D2$).



Clear	A	B	C_{N+1}
0	0	0	0
0	0	1	C_N
0	1	0	C_N
0	1	1	1
1	X	X	0

Fig. 7: Muller C-element with clear input (symbol and truth table)

This can be achieved by applying a '0' on Set and a '1' on Reset

The desired behavior of the gate logic is described by the truth tables for these two signals (Table III).

TABLE III: Truth tables for 7-T gate terminals (Clear)

Clear	A	Set	Clear	B	Reset
0	0	0	0	0	1
0	1	1	0	1	0
1	X	0	1	X	1

The resulting minimum terms are $Set = \overline{Clear} \wedge A$ and $Reset = Clear \vee \overline{B}$. These logic functions can be implemented efficiently using a 4-T multiplexer for each term.

C. C-element with Preset Input

The 7-T gate truth tables for a C-element with preset input are depicted in table IV.

TABLE IV: Truth tables for 7-T gate terminals (Preset)

Preset	A	Set	Preset	B	Reset
0	0	0	0	0	1
0	1	1	0	1	0
1	X	1	1	X	0

The resulting minimum terms are now $Set = Preset \vee A$ and $Reset = \overline{Preset} \wedge \overline{B} = \overline{Preset \vee B}$, calling for an OR and NOR function. While the OR function can again be implemented using a multiplexer as in the C-element with a clear input, the NOR function requires an additional inverter. Thus, a conventional complementary style NOR gate consisting of 4 3-T switches is more efficient. Figure 8 shows the circuit of the C-element with preset input.

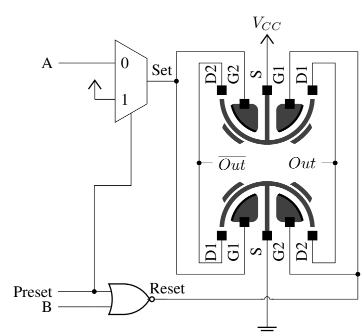


Fig. 8: 2-input MCE with Preset

D. Comparison of C-Element Circuits

Table V shows a device count comparison of relevant implementations. The low number of NEM devices in sequence in the 7-T implementation reduces the overall latency compared to a straightforward implementation of conventional circuit architectures.

TABLE V: Comparison of presented 2-input C-element designs

Design	Device Count
Semi-static [9]	8
Conventional [9]	12
Symmetric [9]	12
MUX-based [4] conservative	14
MUX-based [4] optimized	6
7-T approach	4
7-T approach (+clear)	6
7-T approach (+preset)	8

E. Physical Implementation

To check the behavior of the C-element, analog simulations including Verilog-AMS definitions of all NEM devices have been carried out. These analog models are part of the NEM Process Development Kit (PDK) developed by the University of Bristol [10]. Figure 9 shows the simulation result while the layout of the prototype C-element is shown in Figure 10.

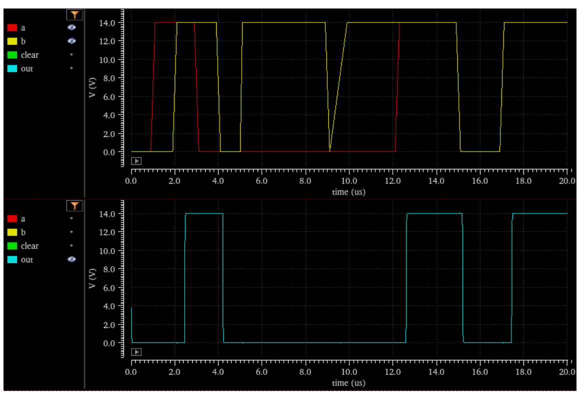


Fig. 9: Verilog-A Simulation of C-element

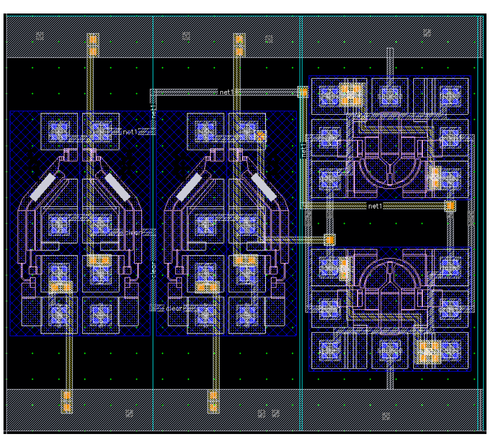


Fig. 10: Layout of NEM C-element with Clear

IV. Demo Circuits

To show the advantages and disadvantages of asynchronous techniques on NEMS, a comparison between a small asynchronous circuit and a synchronous design with equivalent behavior has been carried out. The relevant metrics of the comparison are the total number of switching cycles within both circuits and the device count.

The first selected asynchronous circuit is a Weak-Conditioned Half Buffer (WCHB) based pipeline with 3 stages [1]. This is a simple Quasi Delay Insensitive (QDI) design using a 4-phase handshake between the pipeline stages (Figure 11).

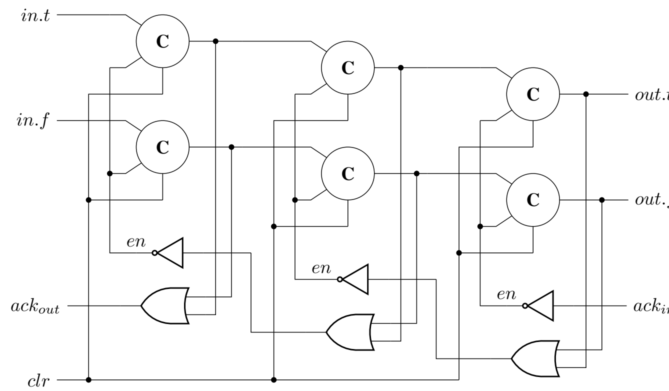


Fig. 11: WCHB pipeline

This asynchronous design with a pipeline can be compared to its synchronous equivalent, a 3-stage pipeline with registers, i.e. D flip-flops (D-FF) with resets, as shown in figure 12).

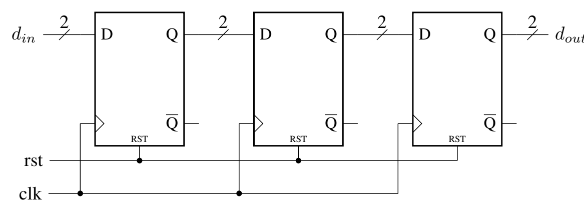


Fig. 12: D-FF based pipeline

Although the data propagated through both pipelines is effectively 1 bit, both transmit over two signal rails for different reasons. For the WCHB with its dual rail encoding, a signal is valid when one rail is set: The WCHB transmits a 'true' by activating $in.t$, and a 'false' by activating $in.f$. The synchronous pipeline simply transmits data on a single rail, but adds a $valid$ signal to indicate when the propagated data represents a new item (as each clock transition may not always convey new and valid data). Figure 12 shows a pipeline of three registers, each consisting of two D-FFs.

To make the inputs and outputs of the synchronous pipeline compatible with the asynchronous one, the following conversions have been made:

$$
\begin{aligned}
in.t &= valid_i \wedge d_{in} & \qquad d_{out} &= out.t \\
in.f &= valid_i \wedge \overline{d_{in}} & \qquad valid_o &= out.t \vee out.f
\end{aligned}
\tag{1}
$$

These conversions (two AND gates and one OR gate) have not been considered in the simulation results.

The setting of the use case scenario is defined such that activity in the circuit is only required if new input data is available and the intervals of data becoming available are changing. As a result, the clock frequency cannot be set to the data rate but might be much higher than the average data rate. This resembles a class of sub-circuits that might benefit from asynchronous design regarding switching activity.

For the experiments we run simulations with custom-defined gates using a Verilog description of the NEM devices. In these descriptions of the NEM devices, the switching activity (opening or closing) is counted, and the trace files of the simulations have been parsed to extract statistical information about the switching cycles inside the simulated designs.

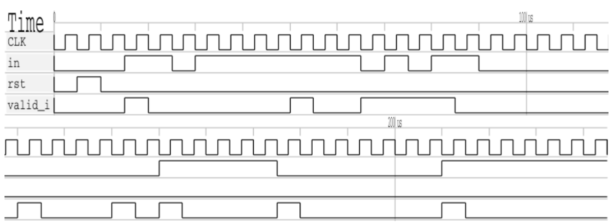


Fig. 13: Test input pattern for the simulation

Figure 13 depicts the waveforms of the input signals and Figure 14 shows the switching activity over all switches of the same type. The two boxplots on the left correspond to the synchronous design, whereas the remaining boxplots on the right correspond to the asynchronous WCHB pipeline.

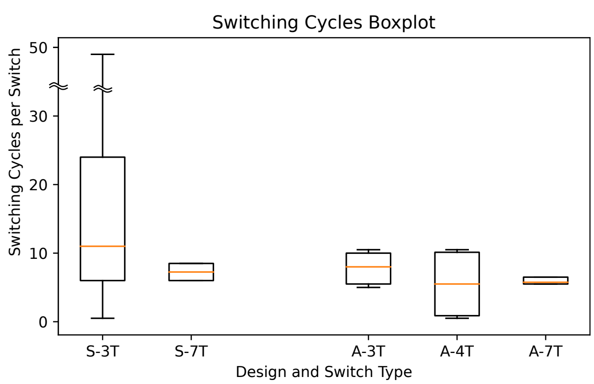


Fig. 14: Boxplot of switching cycles per switch type for both pipeline designs

Tables VI and VII depict the device count and statistics about the switching cycles of both designs numerically, again grouped by the type of switch.

To present a more complex design, the pipeline example has been extended to a 3 stage 4x4 bit unsigned multiplier (Figure 15). The asynchronous multiplier uses WCHBs instead

TABLE VI: Switching cycle statistics: **synchronous** pipeline

		Switching Cycles			
	Device Count	Total	Min	Avg	Max
3T	204	3664	0.5	17.96	49
4T	0	-	-	-	-
7T	24	174	6	7.25	8.5
Total	228	3838	0.5	16.83	49

TABLE VII: Switching cycle statistics: **asynchronous** pipeline

		Switching Cycles			
	Device Count	Total	Min	Avg	Max
3T	24	190	5	7.92	10.5
4T	24	132	0.5	5.5	10.5
7T	12	71	5.5	5.92	6.5
Total	60	393	0.5	6.55	10.5

of registers. Both designs have to wait the same amount of time until the next inputs arrive (Figure 16).

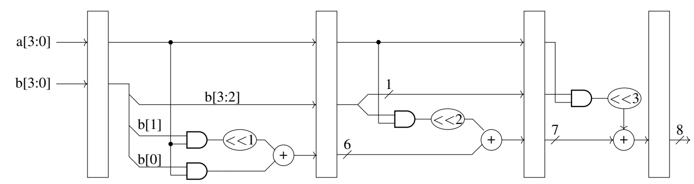


Fig. 15: Circuit of the 3 stage multiplier (synchronous)

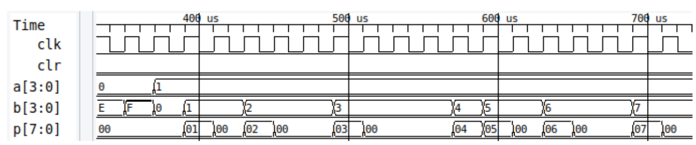


Fig. 16: Snippet of the test pattern for the multipliers

The varying time span between the inputs is produced by the following statement:

```
for (i=0; i < (a_cnt*b_cnt) % 4; i=i+1)
    @(posedge clk);
```

The resulting boxplot (Figure 17) shows that the switching activity of some 3-T NEM devices in the synchronous multiplier is much higher than the rest. The activity in the asynchronous design is more equally distributed and less than in the synchonous multiplier as the statistics in tables VIII and IX show.

TABLE VIII: Switching cycle statistics: **synchronous** multiplier

		Switching Cycles			
	Device Count	Total	Min	Avg	Max
3T	1810	270848.5	0	149.64	518.5
4T	24	1695	33	70.625	96
7T	160	5506	1	34.413	128
Total	1994	278049.5	0	139.443	518.5

V. Discussion

As a key result of the pipeline use case, a comparison of Table VI and Table VII shows the potential of using QDI in NEMS implementations: The switching activity is significantly

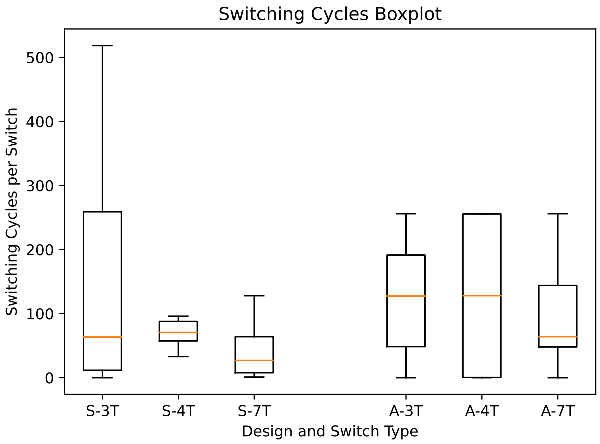


Fig. 17: Boxplot of switching cycles per switch type for multiplier designs

TABLE IX: Switching cycle statistics: **asynchronous** multiplier

		Switching Cycles			
	Device Count	Total	Min	Avg	Max
3T	1252	146483.5	0	116.99	256
4T	320	40960	0	128	256
7T	592	61952	0	104.649	256
Total	2164	249395.5	0	115.247	256

lower on average, and most importantly the maximum value (which determines wear out) is reduced to roughly 20% relative to the synchronous case. Clearly, this is due to the fact that the synchronous pipeline suffered from idle cycles where no data was transmitted. These idle cycles also made the introduction of the $valid$ signal necessary, whereas the asynchronous design could leverage the strength of its dual-rail encoding. While this setting seems to favor QDI, it is not unrealistic for many applications.

Furthermore, it is interesting to observe the reduction in device count for the QDI implementation. This is quite unexpected, as the latter circuits are generally known to incur an area overhead. The reason here is that while the implementations of the SR-latch and the C-element in NEMS are efficient, the current implementations of the D-latch and the D-FF have a high device count. The D-latch could potentially be optimized with 4-T devices. However, a D-FF requires two D-latches, and D-latch implementations will not reach the same low device count as the proposed C-element, at least not without an implementation having a weak inverter feedback loop.

For the multiplier use case we have retained the setting where the input does not supply new data with every clock cycle. In spite of that, however, the QDI implementation's advantages with respect to switching activity are lower now – but still there is a reduction to roughly 83%. In another experiment (data not shown here) we also investigated the extreme case of having no idle cycles for the synchronous circuit (i.e., data is provided with every clock edge). There QDI exhibits approximately 50% more transitions than synchronous design. This clearly indicates that the case for QDI cannot be generalized for all types of application. Finally, we observe roughly 8.5% overhead for QDI in Tables VI and VII in terms of device count, which is due to the fact that flip-flops are less dominant in this circuit than in the pipeline use case.

VI. Conclusion

NEMS technology can provide substantial advantages over standard CMOS in terms of robustness and non-volatile storage capability, but, due to the mechanical movement of switches, suffers from wear out. In this paper we have investigated the potential of QDI design to help mitigate this drawback. In the first contribution, we have presented the NEM-based design of a Muller C-Element, including versions with clear and preset inputs, a vital building block in QDI design. Optimisation of the circuit architecture to take advantage of the logic efficiencies inherent in NEM switches means that the proposed design is much more efficient than straightforward CMOS-like implementations. In the second contribution, we have performed a comparison of switching activity and NEM device count for synchronous and QDI implementation for two simple but representative use cases. Our results show that QDI implementation can yield significant benefits in cases where (a) synchronous design suffers from idle states and (b) the proportion of flip-flops in the circuit is high. The latter is due to the fact that flip-flops seem to be less suited for NEMS implementation.

In summary, this case study has shown that asynchronous design techniques fit well to NEMS technology. Detailed timing and metastability investigations will be carried out in future work.

References

[1] Florian Huemer, Robert Najvirt, and Andreas Steininger. "Identification and Confinement of Fault Sensitivity Windows in QDI Logic". In: *2020 Austrochip Workshop on Microelectronics (Austrochip)*. IEEE. 2020, pp. 29–36.

[2] David Eugene Muller. *Theory of asynchronous circuits design*. Tech. rep. University of Illinois, 1955.

[3] D. Pamunuwa et al. "Theory, Design, and Characterization of Nanoelectromechanical Relays for Stiction-Based Non-Volatile Memory". In: *Journal of Microelectromechanical Systems* 31.2 (2022), pp. 283–291.

[4] Petr Pfeifer and Heinrich Theodor Vierhaus. "On implementation and usage of Muller C-element in FPGA-based dependable systems". In: *2017 International Conference On Applied Electronics (AE)*. IEEE. 2017, pp. 1–4.

[5] Tian Qin et al. "Performance analysis of nanoelectromechanical relay-based field-programmable gate arrays". In: *IEEE Access* 6 (2018), pp. 15997–16009.

[6] Sunil Rana et al. "Nano-crystalline graphite for reliability improvement in MEM relay contacts". In: *Carbon* 133 (2018), pp. 193–199.

[7] Sunil Rana et al. “Nanoelectromechanical relay without pull-in instability for high-temperature non-volatile memory”. In: *Nature Communications* 11.1 (2020), p. 1181.

[8] James D Reynolds et al. “Single-Contact, Four-Terminal Microelectromechanical Relay for Efficient Digital Logic”. In: *Advanced Electronic Materials* 9.1 (2023).

[9] M. Shams, J. Ebergen, and M. Elmasry. “Modeling and comparing CMOS implementations of the C-element”. In: *IEEE T. on VLSI Systems* 6.4 (1998).

[10] Elliott Worsey. “Nanoelectromechanical Technology for Radiation and Temperature Harsh Environments”. PhD thesis. University of Bristol, 2024.