# Synchronization of Clocked Field-Coupled Circuits

Frank Sill Torres[1,2], Marcel Walter[1], Robert Wille[2,3], Daniel Große[1,2] and Rolf Drechsler[1,2]
[1]Group of Computer Architecture, University of Bremen, Germany, [2]Cyber-Physical Systems, DFKI GmbH, Bremen, Germany,
[3]Inst. of Integrated Circuits, Johannes Kepler University Linz, Austria

frasillt@uni-bremen.de, m_walter@uni-bremen.de, robert.wille@jku.at, grosse@uni-bremen.de, drechsler@uni-bremen.de

***Abstract*—**Proper synchronization in clocked *Field-Coupled Nanocomputing* (FCN) circuits is a fundamental problem. In this work, we show for the first time that global synchronicity is not a mandatory requirement in clocked FCN designs and discuss the considerable restrictions that global synchronicity presents for sequential and large-scale designs. Furthermore, we propose a solution that circumvents design restrictions due to synchronization requirements and present a novel RS-latch.

## I. Introduction

*Field-Coupled Nanocomputing* (FCN) offers a promising alternative to conventional circuit technologies. In FCN, computations and data transfer is realized via local fields between nanoscale devices that are arranged in patterned arrays [1]. Theoretical and experimental results indicate that FCN-based approaches have the potential to allow for systems with highest processing performance and remarkable low energy dissipation [2]. Consequently, numerous contributions on their physical realization have been made in the past, e.g. molecular quantum cellular automata (mQCA) [3], atomic quantum cellular automata (aQCA) [4] or nanomagnetic logic (NML) [5].

Clocked FCN circuits apply external clocks in order to circumvent the issue of metastability and to control the data flow. In case of mQCA and aQCA, electric clocks control the tunneling within a cell, while in NML a magnetic clock controls the switching ability of the nanomagnets. Depending on the technology, each device or cell changes during a complete clock cycle between four (mQCA, aQCA) or three (NML) different phases, i.e. a switch, a hold, a reset and a neutral phase (the latter only in case of mQCA and aQCA). For the sake of simplicity and without loss of generality, we will consider a four-phase technology in the following.

In case of four phases, normally four external clocks numbered from 1 to 4 are applied, whereby each clock controls a selected set of cells. For fabrication purposes, cells are usually grouped in a grid of square-shaped tiles such that all cells within a tile are controlled by the same external clock [6, 7]. All four clocks have a phase difference of 90 degrees. It is important to note that correct data flow is only possible between cells controlled by consecutively numbered clocks. That means, cells controlled by clock 1 can solely pass its data to cells controlled by clock 2 etc. and, finally, from clock 4 to clock 1. Hence, there is a local synchronization of signals located in neighboring tiles, and the data flow between tiles is conducted in a pipeline-like fashion controlled by the external clocks.

This behavior leads to the common assumption that clocked FCN circuits must not employ only a local but also global pipeline-like behavior. That means, it is assumed that all signal paths arriving at the same logic gates must have equal length and that all signals must always arrive at the respective logic gates in a synchronized manner.

For small combinational circuits, this so-called *global synchronicity* (GS) can easily be guaranteed. However, for large-scale as well as sequential designs, GS poses a considerable design restriction (as discussed in Section II). Since scalability and sequential behavior are prerequisites for practically relevant applications of FCN, this poses a serious threat to the further development of this technology which has not been considered yet.

In this work, we, to the best of our knowledge, for the first time, address this problem. We show that GS is not a mandatory requirement in clocked FCN circuits and, furthermore, propose a simple but effective solution that enables the synchronization of circuits violating the GS constraint (see Section II). In order to apply this solution in more complex circuits, we introduce a latch-like structure that uses external clocks for signal synchronization and add set and reset functionality (see Section III). Results presented in Section IV indicate the feasibility of the proposed approach. Finally, Section V concludes this work.

## II. Global Synchronicity of FCN Circuits

### A. GS in Combinational Circuits

A fundamental characteristic of globally synchronized designs is that in each clock cycle new data can be applied to the primary inputs of the circuit. After the first input data passed the circuit, correspondingly new results arrive at the circuit's primary outputs in each clock cycle – resulting in a circuit throughput of 1. Furthermore, a globally synchronized circuit does not require synchronization elements like latches as, by definition, all related data are always synchronized.

However, in contrast to many related statements in the literature [8, 9], GS is not a mandatory constraint in clocked FCN circuits. As shall be shown in the following.

**Example 1**. *Fig. 1 depicts a structural FCN implementation of an exemplary circuit consisting of three operations o1, o2 and o3 and two primary inputs PI1 and PI2. This circuit fulfills the local synchronization requirement, i.e. data is only passed between tiles controlled by consecutively numbered clocks. However, the paths between primary inputs PI1 and PI2 and operation o3 differ in their length by more than 3 tiles. Thus, data sent at the same time from PI1 and PI2 arrive in different clock cycles at o3 and, consequently, GS is not given.*

A common solution for the problem in the given example would be the relocation of *PI1* or *PI2* such that paths have equal

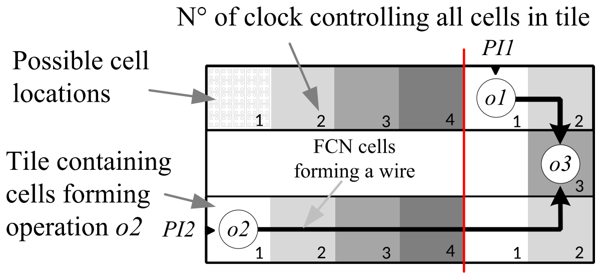
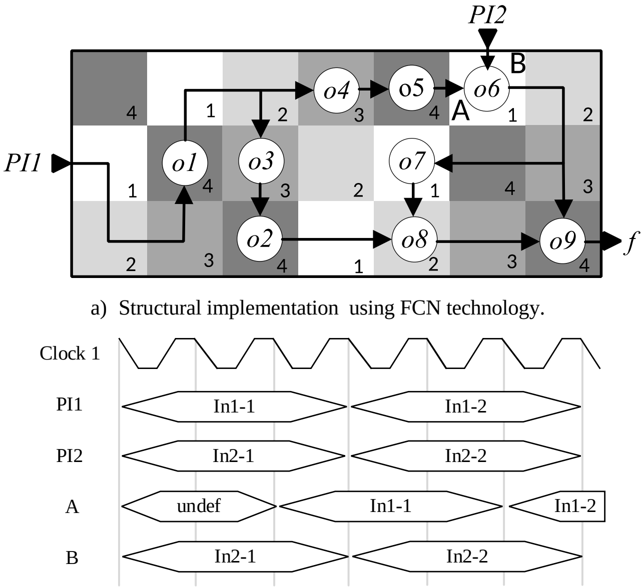

N° of clock controlling all cells in tile
Possible cell locations
Tile containing cells forming operation o2
FCN cells forming a wire
PI1
PI2
o1
o2
o3
1
2
3
4
1
2
3
1
2
3
4
1
2

Fig. 1. FCN circuit failing the global synchronicity. The red line indicates the limit until where *PI1* could be placed such that paths *PI1*→*o3* and *PI2*→*o3* are synchronous.



a) Structural implementation using FCN technology.

b) Selected signal waves if inputs are hold for two additional clock cycles.

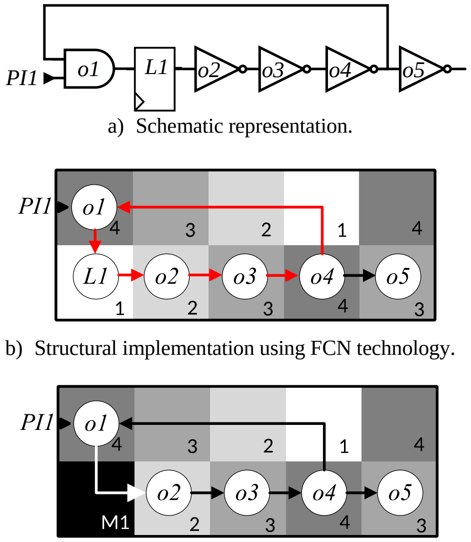
Fig. 2. Combinational clock FCN circuit violating the GS constraint and proposed holding of input signals.

lengths. However, this usually comes at considerable costs in terms of area and design complexity [10].

Instead, we propose to reduce the frequency with which new input data are applied. That means, we require that input data can be kept stable for more than one clock cycle. On the downside, this reduces the throughput of the design. On the other hand, this approach allows for the reduction of area costs and design complexity.

**Example 2.** *As discussed in the first example, the circuit depicted in Fig. 1 has misses the GS constraint. In order to solve this restriction, one can define that data connected at PI1 and PI2 must be kept stable for two clock cycles – leading to a reduced throughput of 1/2.*

The frequency with which new input data can be sent to the circuit depends on the maximum difference between the arrival times of all inputs of an operation of the FCN circuit. That means, it must be assured for all operations that their inputs are synchronous at least for one clock cycle before new inputs arrive.



a) Schematic representation.

b) Structural implementation using FCN technology.

c) Structural implementation using an artificial latch.

Fig. 3. Sequential circuit failing GS. Red lines indicate the path that should have a maximum length of 4 tiles.

**Example 3.** *Fig. 2a depicts an exemplary circuit that violates the GS constraint. Several of the operations oX have two inputs which have diverging arrival times. In detail, the inputs of o6 arrive after 1 and 9 clock phases, the inputs of o8 after 10 and 14 clock phases, and the inputs of o9 after 12 and 16 clock phases. As each clock cycle last 4 clock phases, the maximum difference in terms of clock cycles results from the ceiling division by 4. That means, in case of o6 the difference results to* $\lceil 9/4 \rceil - \lceil 1/4 \rceil = 2$*. Similar, the maximum difference in terms of clock cycles for o8 and o9 results to 1. Hence, both inputs PI1 and PI2 must not change for two additional clock cycles in order to assure correct operation. Fig. 2b shows the resulting signal waves of clock 1, the inputs and the signals at points A and B, both highlighted in Fig. 2a. One can note that only at the third clock cycle, operation o6 has inputs that are synchronous, i.e. both inputs A and B have data (In1-1 and In2-1) that have been sent at the same time. The presented circuit has a latency of 16 clock phases, i.e. 4 clock cycles. If the input frequency is reduced to 1/3 of the clock frequency then the first correct results will arrive after 6 clock cycles. Next, every three clock cycles new correct outputs will be available.*

### *B. GS in Sequential Circuits*

The problem of GS represents itself in more restricting manner in sequential circuits. A common characteristic of sequential circuits are feedback paths that require synchronicity of new input data as well as data coming from the feedback paths. Following example shall highlight this restriction.

**Example 4.** *Fig. 3a depicts an exemplary circuit with a feedback. The computations in the circuit are controlled by the latch L1. Fig. 3b shows the structural implementation using clocked FCN technology, in which the latch is implemented in one tile. In order to avoid the latch, the output data of latch L1 must arrive within one clock cycle at the input of L1 in order to assure a correct operation. Consequently, the physical path length between output and input of L1 must be less or equal to 4 tiles. However, due to the given configuration of the circuit,*

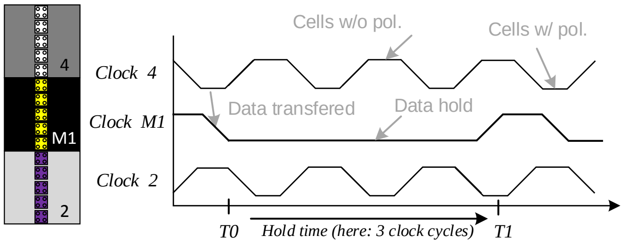


Fig. 4. Memory *clock M1* reproducing latch-like behavior applied for an QCA-like circuit.

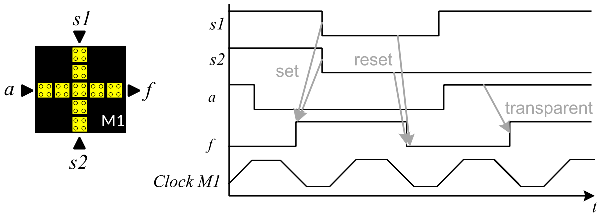


Fig. 5. RS-Latch controlled via external clock M1 with set and reset functionality.

*this path length is not achievable, preventing the global synchronicity of this circuit.*

A possible solution for this problem would be the implementation of a D-latch circuit as e.g. proposed in [11]. This comes at high costs, though. Hence, in order to circumvent this problem, we propose again to hold the data at primary inputs as shall be shown in following example.

**Example 5.** *In order to assure correct functionality of the circuit depicted in Fig. 2b, the period with which new data are connected with PI1 must be increased to two clock cycles. This assures that data coming from o4 and going to o1 arrive at the same time new inputs are coming from PI1.*

## III. Artificial Latch

As stated above, circuits that completely fulfill the GS constraint do not require any latches or flip-flop elements. In contrast, circuits that fail to comply with the GS constraint and, consequently, are required to hold data, have the need of latches and/or flip-flop circuits.

### A. Basic Latch

Having in mind the routing overhead of an additional control signal for latches and/or flip-flops, we propose the use of an additional external memory clock, similar to an idea presented in [12], in order to create an artificial latch[1]. This clock, which we call clock Mx, is configured such that it can receive data from cells clocked by the antecedent clock, i.e. clock x – 1, and pass data to cells controlled by the subsequent clock, i.e. clock x + 1. Moreover, the clock can be configured such that it holds data over several clock cycles. That means, the clock phase in which data are hold can be extended. Consequently, this clock enables the implementation of a wire that has a latch-like behavior as shall be discussed in following example.

**Example 6.** *Fig. 4 depicts an QCA-like wire structure, where data are flowing from top to down. The middle tile is controlled by the memory clock M1. This tile controls the data flow between the tile controlled by clock 4, i.e. the antecedent clock of M1, and the tile controlled by clock 2, i.e. the subsequent clock of M1. Clock M1 is synchronized with clock 4 such that the cells controlled by M1 can receive new data during a falling slope of clock M1 (at time T0 in Fig. 4). Further, clock M1 is configured to keep these data stable for three clock cycles (until time T1 in Fig. 4). That means, for three consecutive clock cycles, cells in the tile controlled by clock 2 receive the data stored in the tile controlled by clock M1, independently of any changes of the data within the tile controlled by clock 4.*

**Example 7.** *Fig. 3c shows how the proposed latch can be applied to enable a feasible implementation of the sequential circuit depicted in Fig. 3a. Here, the latch structure is realized by a wire controlled by clock M1. This clock must be configured such that only every second clock cycle new data coming from o1 are read in.*

The frequency of each clock Mx must be chosen depending on the longest time data have to be hold in the design in order to guarantee synchronicity. If desired, one can also implement more than one clock Mx. This, however, comes at the cost of higher complexity and requires adequate design environments.

Simulations in a modified version of the QCADesigner[2] [10] revealed that during hold phase, the latch controlled by clock Mx acts also as input for cells located in a tile controlled by the antecedent clock. Consequently, during evaluation phase of this tile, its cells can assume logically wrong values. For example, in case of a wire, cells close to the actual input might assume the new input value, while cells close to the latch assume the value of the latch. This poses no problem for logic operations and wires, having in mind that the output is not processed by the latch. However, for fanout structures, i.e. operations with more than one output, this behavior might lead to errors. Consequently, one should avoid to place any fanout structure before an artificial latch.

### B. RS-Latch

A tile controlled by a clock Mx can also contain more elaborated structures. Following this observation, we propose an artificial latch with set and reset function for QCA-like technologies as depicted in Fig. 5. This circuit possess the three inputs *a*, *s1* and *s2* and the output *f*. The *set* function is activated if both inputs *s1* and s2 assume the value '1', while the *reset* is activated if both inputs *s1* and *s2* assume the value '0'. In both cases, the output value *f* is identical to the values of *s1* and *s2* due to the majority function of the circuit. In case of opposing values of *s1* and *s2*, the output *f* follows the input *a*. The exemplary signal waves in Fig. 5 highlight this behavior.

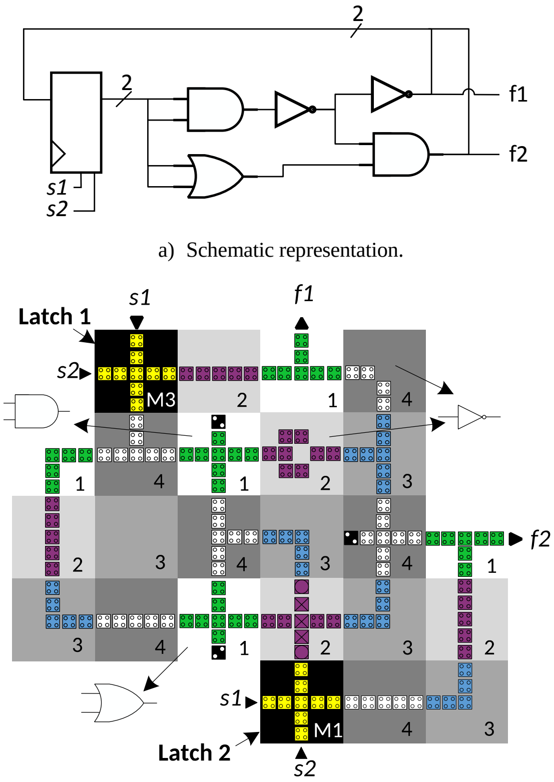
## IV. Results

### A. Tradeoff-Analysis

In order to analyze the possible tradeoff between area and throughput by ignoring the GS constraint in FCN circuits, we implemented an automatic layout tool for clocked QCA-like

[1] Artificial means here that the actual latch function is implemented via a technological modification and not via a specific circuit.

[2] The corresponding software QCADesigner-E is publicly available at https://github.com/FSillT/QCADesigner-E.



a) Schematic representation.

b) QCA-like implementation.

Fig. 6. 2-bit counter with set and reset.

circuits [13]. This tool generates the exact solution for the smallest layout for a given circuit. We designed and verified several combinational circuits in two versions – one fulfilling the GS constraint and one violating it. Next, we compared both versions in terms of area and reduced throughput due to the requirement of holding input data. The related results are listed in Table 1. The 2nd column lists area reduction by choosing the version that violates the GS constraint, while the 3rd column contains the consequent decrease of the throughput. The results indicate that ignoring GS can lead to considerable improvement of area costs due to the lack of synchronization wires. On the downside, this comes at the cost of lower throughput. Hence, designers must choose which parameter to prioritize.

### B. *Examplary Sequential Ciruit*

In a following step, we implemented a 2-bit counter with set and reset function, depicted in Fig. 6a, in a QCA-like technology, shown in Fig. 6b. The outputs *f1* and *f2* of the counter are set and reset if signals *s1* and *s2* are both ‘1’ or ‘0’. If *s1* and *s2* have opposing values, the counter is incremented depending on the periods of clock M1 and M3.

The minimum required periods follow from the difference of clock cycles until both inputs of the 2-bit latch are stable. One can determine four following paths with its corresponding length given in clock cycles: Latch M1 → Latch M1 (3), Latch 1 → Latch 2 (4), Latch 2 → Latch 1 (4), Latch 2 → Latch 2 (3). Both latches are identified in Fig. 6. Hence, the clock M1 and M3 should have a minimum hold time of 4 – 3 = 1 clock cycles, leading to a minimum clock period of 2.

| Circuit | Area Gain | Throughput |
|---|---|---|
| 4:1 MUX | 8 % | ½ |
| Parity Generator | 43 % | ½ |
| ISCAS85 c17 | 30 % | ⅓ |

Table 1. Comparison of QCA circuits implemented with and without GS.

## V. Conclusions

In this paper, we discussed global synchronicity (GS) in clocked FCN circuits and revealed that GS is not a mandatory constraint for this technology. Moreover, especially for sequential and large-scale circuits, ignoring GS might be fundamental for enabling the feasibility of these designs.

We proposed a straight-forward approach for assuring synchronicity via the delaying of input signals. Further, we introduced an artificial RS-latch which applies additional external clock signals. Simulation results indicate the feasibility of the approach and highlight possible improvements in terms of area at the costs of performance.

## References

[1] N. G. Anderson and S. Bhanja, *Field-coupled Nanocomputing: Paradigms, Progress, and Perspectives*, 1st ed. New York: Springer, 2014.

[2] J. Timler and C. S. Lent, "Power gain and dissipation in quantum-dot cellular automata," *J Appl Phys*, vol. 91, pp. 823-831, 2002.

[3] V. Arima, M. Iurlo, L. Zoli, S. Kumar, M. Piacenza, F. D. Sala, *et al.*, "Toward quantum-dot cellular automata units: Thiolated-carbazole linked bisferrocenes," *Nanoscale*, vol. 4, pp. 813-823, 2012.

[4] T. R. Huff, H. Labidi, M. Rashidi, M. Koleini, R. Achal, M. H. Salomons, *et al.*, "Atomic White-Out: Enabling Atomic Circuitry through Mechanically Induced Bonding of Single Hydrogen Atoms to a Silicon Surface," *ACS Nano*, vol. 11, pp. 8636-8642, 2017.

[5] I. Eichwald, A. Bartel, J. Kiermaier, S. Breitkreutz, G. Csaba, D. Schmitt-Landsiedel, *et al.*, "Nanomagnetic Logic: Error-Free, Directed Signal Transmission by an Inverter Chain," *IEEE TMag*, vol. 48, pp. 4332-4335, 2012.

[6] J. Huang, M. Momenzadeh, L. Schiano, M. Ottavi, and F. Lombardi, "Tile-based QCA Design using Majority-like Logic Primitives," *JETC*, vol. 1, pp. 163-185, 2005.

[7] C. A. T. Campos, A. L. Marciano, O. P. V. Neto, and F. S. Torres, "USE: A Universal, Scalable, and Efficient Clocking Scheme for QCA," *IEEE TCAD*, vol. 35, pp. 513-517, 2016.

[8] J. Huang, M. Momenzadeh, and F. Lombardi, "Design of sequential circuits by quantum-dot cellular automata," *Microelectronics Journal*, vol. 38, pp. 525-537, 2007.

[9] L. Lee Ai, A. Ghazali, S. C. T. Yan, and F. Chau Chien, "Sequential circuit design using Quantum-dot Cellular Automata (QCA)," in *IEEE ICCAS*, 2012, pp. 162-167.

[10] F. S. Torres, R. Wille, P. Niemann, and R. Drechsler, "An energy-aware model for the logic synthesis of quantum-dot cellular automata," *IEEE Trans. on CAD of Integrated Circuits and Systems*, 2018.

[11] D. A. Reis, T. B. Soares, C. Campos, A. Marciano, O. P. Vilela Neto, and F. Sill Torres, "A Methodology for Standard Cell Design for QCA," in *IEEE International Symposium on Circuits and Systems (ISCAS2016)*, Montreal, Canada, 2016.

[12] M. Ottavi, S. Pontarelli, E. P. DeBenedictis, A. Salsano, S. Frost-Murphy, P. M. Kogge, *et al.*, "Partially Reversible Pipelined QCA Circuits: Combining Low Power With High Throughput," *IEEE TNANO*, vol. 10, pp. 1383-1393, 2011.

[13] M. Walter, R. Wille, D. Große, F. S. Torres, and R. Drechsler, "An exact method for design exploration of quantum-dot cellular automata," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 503-508.