IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 June 2022, accepted 25 July 2022, date of publication 28 July 2022, date of current version 4 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3194601

RESEARCH ARTICLE

An ASIC Architecture With Inter-Chip Networking for Individual Control of Adaptive-Metamaterial Cells

LOUKAS PETROU, (Member, IEEE), AND JULIUS GEORGIOU, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, University of Cyprus (UCY), 1678 Nicosia, Cyprus

Corresponding author: Loukas Petrou (petrou.loukas@ucy.ac.cy)

This work was supported in part by the European Union's Horizon 2020 Future Emerging Technologies call through the Research and Innovation Action (FETOPEN-RIA) (Project VISORSURF) under Grant Agreement no. 736876 and in part by the European Regional Development Fund and the Republic of Cyprus through the Research and Innovation Foundation, under Projects INFRASTRUCTURES/1216/0042 (RF-META) and COMPLEMENTARY/0916/0008 (HSADAPT).

ABSTRACT In this paper, the architecture of an application-specific integrated circuit for adaptive metasurfaces is presented. The architecture allows scalable networking over large metasurfaces and reconfiguration of each unit-cell with unique complex impedance settings, for adjusting metasurface electromagnetic performance. The one-chip-design metasurface array, includes self-initialization/addressing, configuration-packet routing, and network adaptation for fault-tolerant dynamic metasurfaces. An asynchronous design is adopted for power efficiency and high scalability, whereas speed is enhanced through multilevel pipelining. The ability to dynamically form reconfigurable networks in conjunction with the clockless operation helps the metasurface to cover arbitrary physical shapes, implemented on both rigid or flexible substrates. The architecture is validated through transistor-level simulations of the complete design implemented in a commercially available process. Then, a variety of scalable and/or flexible networks are presented, exploiting the strengths offered by the architecture, to demonstrate its capabilities and estimated performance. The demonstrated results of the architecture and its networking capabilities, allow the realization of chip-to-chip programmability of metasurfaces with up to 2^{18} ASICs. The individual ASICs can be reconfigured in 2 μs and consume only 342 μW of static power.

INDEX TERMS Metasurface ASIC, fault-adaptive metamaterial ASIC network, ASIC architecture, asynchronous network, inter-tile network, adaptive metamaterial cells, programmable metamaterials.

I. INTRODUCTION

With the advent of reconfigurable processors and software defined networking [1], there have been many architectures [2]–[6] that provide flexibility and energy efficiency in network systems. Recently, Network-on-Chip (NoC) systems have been established as the dominant communication method in multicore processors, mostly because of their increased scalability [7]–[9]. However, until now, there has been limited availability of networked-IC solutions to address the needs for programmable metamaterials and programmable metasurfaces (MSFs). In general, metamaterials are engineered structures that enable unnatural electromagnetic (EM) functionalities, while MSFs are the two-dimensional versions of metamaterials that can be tuned or programmed to enable programmable adjustment of the surface's EM properties. MSF systems in the literature are mostly static and are tailored for a single application, with EM functionalities operating at limited frequencies and angles of incidence [10]–[16].

Reconfigurability in metamaterials has been under research for more than a decade [17], with many systems configuring the MSF using tunable devices and mechanical parts [18]. Also, other approaches utilize more advance materials like graphene [19], [20] or liquid crystals [21] to configure the MSF. However, in all these approaches

The associate editor coordinating the review of this manuscript and approving it for publication was Chinmoy Saha.

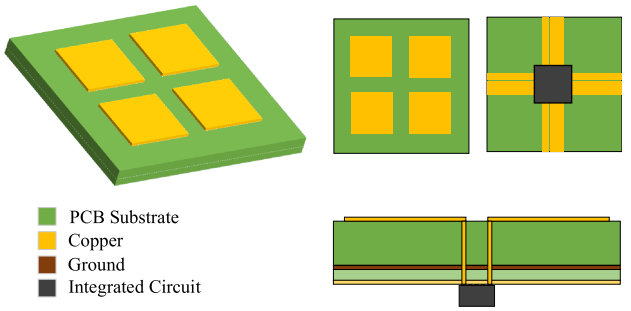

FIGURE 1. Proposed integration of ASICs in MSF meta-atoms as described in [30]. The integrated circuit is soldered at the bottom-side of the meta-atom and is connected to the copper-patches at the top-side using through-via connections. The ground plane is placed in between the routing tracks of the integrated circuit and the copper-patches to isolate the two layers.

the reconfigurability is very limited, since they rely on the properties of the materials and the meta-atoms do not contain elements that include programmatic control over the desired functionality. Programmability on MSFs is mainly achieved by using lumped elements such as PIN diodes and varactor diodes as part of the MSF structure [22]–[24] and an external controller that leads to complex, bulky, non-scalable wiring solutions [25], [26]. These lumped elements are controlled by a Field-Programmable Gate Array (FPGA) providing a binary control over the lumped element state [22]. Although this approach provides some programmability to the static MSF systems, they have shortcomings in terms of their limited tuning range, network scalability, fault-tolerance, power-consumption and commercial viability. The evolution of these discrete-based approaches, which aims to eliminate the above-mentioned shortcomings, is to replace the lumped elements, and a large portion of the control circuitry, with application-specific integrated circuits (ASICs). This takes advantage of customized integrated circuit technology to improve the power, the size and the performance of MSFs. In addition, the MSF system will become software-driven and will be able to host multiple functionalities, which will allow them to adapt to the system needs. Furthermore, the ASIC approach is more suitable for mass production of MSFs both in terms of cost and production time. The ASIC includes loading element circuits such as varistors and varactors in conjunction with digital control circuits in one single tiny package. In doing so, a plethora of novel MSF functionalities will be unlocked and new applications will arise such as programmable wireless environments, dynamic holograms, THz wave manipulation etc. Despite its tremendous theoretical potential, there are many challenges in all design levels, both theoretical and practical, which is one of the main reasons for the lack of network ASICs for programmable MSF systems. This work identifies these design challenges and proposes solutions to overcome them.

Networked-IC programmability requires thousands of homogenous chips to be networked on a large printed circuit board (PCB), to be simpler for mass production. The chips uniquely adjust each meta-atom's properties that make up the programmable MSF. A theoretical approach is described in [27]–[29] whereby the MSFs are configured by a digital device e.g. a computer by utilizing a user-friendly application programming interface, to send configuration data to a network of integrated circuits that are directly connected to the MSF. In these de-centralized, reconfigurable systems, each meta-atom, or a small cluster of meta-atoms, has a corresponding chip on the back side of the PCB, connected to some metal texture patterns on the front side of the PCB, where the EM waves strike the surface. The chips adjust the electrical impedance of all parts of the surface and hold their settings, even if the digital device is removed. Such chips are required to have low-power, should be able to inherently scale over large surfaces, and should have a means of changing the electrical load on each RF port. An example distributed, programmable meta-atom is shown in Fig. 1 and is described in detail in [30]. Here, the same meta-atom design is being adopted to illustrate the proposed ASIC as part of a structure that has already being studied and tested.

While the contribution of this paper is mainly focused on a chip architecture with novel features for increased performance, it is useful to briefly provide an overview of the procedure followed to apply the impedance to the meta-atoms, and how this impedance controls the EM properties of the MSF. The EM impedance is formed at the chip's RF ports/pads by electronically tunable resistive and capacitive integrated components. The RF pads are connected to the texture patterns at the meta-atom plane using through-via or via-in-pad technology, thus setting the desired impedance on the communication plane, this way controlling the EM behavior through a reconfigurable pattern. By extending this approach on an array of meta-atoms forming a metasurface, continuous control over the real and imaginary part of the complex surface impedance is obtained, which leads to shaping the special profile of the reflection amplitude, as well as the reflection phase. The configuration data is pre-calculated on a computer, based on the EM properties of the incidence waves and the communication plane of the metasurface. Once calculated, the configuration data is stored in look-up tables in the gateway device. The gateway device can be any digital device that can convey bitstreams to the chips, for example an FPGA, a microcontroller etc. When the functionality of the system is selected by the user, the gateway device will prepare a serial bitstream consisting of many configuration packets, each one intended to reach a specific chip, so that the appropriate data reaches and get stored in the destination nodes and set the bias voltage of all the varistors and varactors. Note that the computer can be removed once the configuration data has been calculated, leaving only the gateway and the metasurface as part of the system, thus making the overall system mobile, this way extending the system applications to mobile units e.g. vehicles, planes etc. The performance from the EM perspective of a metasurface utilizing the metallic pattern of Fig. 1 is evaluated in [30], showing multiple functionalities achieved like tunable perfect absorption, anomalous reflection and polarization control on

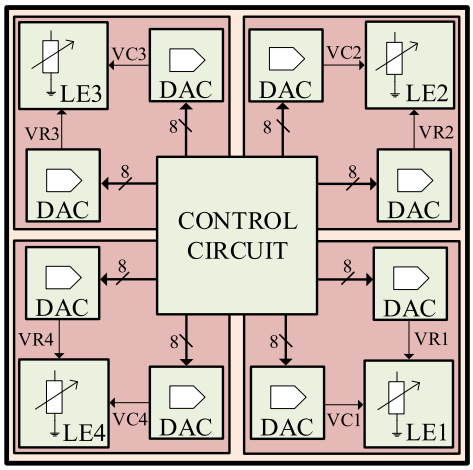

FIGURE 2. Top-Level block diagram of the ASIC. It consists of four RF Loading Elements (LE1-4) for setting the meta-atom/s impedance, eight digital-to-analog converters for converting the digital configuration data to analog bias voltages for the varistors and varactors of the LEs, and one control circuit for communication and networking between neighboring ASICs and/or other external digital devices.

a wide range of incidence waves of both TE and TM polarizations. Utilizing the technology of integrated circuits, the authors showed controlled over amplitude, wave-front and polarization of the output wave. Moreover, control over a frequency range is possible through fast time reconfiguration of the impedance. To achieve fast reconfiguration times, the control circuitry that is incorporated in each chip has to be carefully designed to meet certain specifications and avoid a variety of constraints that limits its performance. These are presented in details in Sections II and III.

In this paper, the architecture of an ASIC, which realizes adaptive and/or reconfigurable networks, as well as dynamically adjustable RF loads is presented. The architecture, implements a dynamic addressing scheme, alongside a routing algorithm and a fault tolerant mechanism. This approach allows the realization of highly scalable, board-to-board MSF systems that can be fitted on large surfaces to co-operate with the already established communication systems. Such MSF systems impose many requirements and constraints on the architecture of any ASICs optimized for MSF applications and have to be addressed in order to get the desired programmability from the system. These constraints include the ASIC size, the EM noise generation from the controllers, the appearances of faults in the network [31], the scalability, the power consumption, the cost and finally the necessity for asynchronous digital circuit design [32]. The proposed ASIC architecture aims to realize an ASIC-based programmable MSF enabler, that will allow for scalable, low-power, low-cost programmable MSFs to be implemented. Also, the architecture aims to significantly increase the number of available states per unit-cell in conjunction with fast reconfiguration and programmability. Furthermore, this work aims to establish both the theoretical and practical limitations for the design of ASICs for MSF applications, by providing the requirements and constraints of such a system and proposing a generic architecture that addresses all issues associated.

Section II presents the ASIC-based architecture and describes the addressing methodology, the fault-tolerant mechanism and the routing topology adopted. Section III presents the control circuit architecture along with the design choices that dictate the performance and the capabilities of this particular design. Section IV shows transistor-level simulation results for the performance of the ASIC including the I/O pads and output buffers. Then in Section V, the capabilities of the architecture are demonstrated through network scenarios that are created using the ASICs presented. The network scenarios exploit both the addressing methodology and the fault-tolerant mechanism. The performance of these network scenarios is also calculated and presented. The paper concludes with a discussion on the results and the future prospects of the architecture.

II. ASIC ARCHITECTURE

The ASIC's top-level block diagram is shown in Fig. 2. It is divided in three parts: The control circuit, the Digital-to-Analog-Converters (DACs), and the RF Loading Elements (LEs).

The control circuit manages the digital communication aspects of the ASIC by sending or receiving routing/communication data to/from its neighboring controllers as well as any other digital device that respects its communication protocol. Also, it consists of an internal memory with sixty-four cells to store the configuration data required for the RF LEs and another internal memory with eighteen cells to store the addressing data required for networking. The memory cells that store the configuration data set the inputs of the eight, 8-bit DACs. A detailed description on the control circuit architecture is delivered in Section III.

The LEs, consist of a MOSFET varistor to adjust the real part of the surface impedance, and a MOSFET varactor to adjust the imaginary part of the surface impedance. The tuning range of the LEs is controlled by the applied voltage on the MOSFETs gates, where the DACs' output voltages are connected. Voltage V_R sets the varistor value of the LE whilst voltage V_C sets the varactor value. The LEs can be realized in both parallel and series configuration which affects the tuning range of the LEs. A detailed elaboration on the design of the LEs and their tuning range can be found in [33]. The LEs presented in [33] are considered part of a conventional microprocessor-based architecture. In this paper, the focus is on a novel control circuit architecture, that is very low power, but also has highly-scalable, fault-tolerant networking capabilities, which is able to control these LEs. The architecture of the ASIC is kept generic, however the control circuit presented offers novel and unique functionalities that could be easily adopted by various designs.

An 8-bit two-stage resistor string architecture was adopted for the DAC design. This architecture guarantees monotonic output [34] and it consists only of passive components thus allowing easy matching and relatively small area on chip.

Also, there is no need for output buffers to drive the LEs since they are MOSFETs with very high input impedance.

As mentioned, the key advantage of the networks which can be implemented by the proposed architecture is their ability to control the EM properties of each meta-atom individually on both rigid and flexible, board-to-board surfaces. The configuration data is provided to the ASICs programmatically in the form of network packets, through the network on the non-EM side of the MSF. The network packet is divided in two parts, the header and the payload. The former consists of data regarding the source and destination addresses while the latter consists of the data to be stored in the destination node's configuration/payload memory. A detailed description for the packet format structure of the proposed architecture is provided in Section III. The complex EM configuration calculations can be performed on a computing device, externally to the MSF. Specifically, a computer can be incorporated in order to implement routing techniques and apply formal methods to exploit and validate the networking capabilities of the ASICs. Then, the computer generates the appropriate configuration packets for every available functionality that the system is able to perform, considering the network topology adopted. Note that the validation process of the routing algorithm needs to be performed once and then the configuration packets are stored in a look-up table in the gateway and the computer is removed. The generated configuration packets are sent to the network grid via the gateway and its I/O ports. The gateway utilizes the same communication protocol as the ASICs and ensures valid communication.

The proposed architecture includes two external input ports and two external output ports (Fig. 3(a)). The communication between the ASICs/nodes is serial-unidirectional (bit-by-bit), because of I/O pad limitations for the selected chip package and operation frequency. Details are given in subsection *'ASIC Constraints / Limitations'*. In Fig. 3(b), the ASIC type is categorized according to its orientation, which is uniquely calculated by internal circuitry in each ASIC according to the address stored in its memory. A 'type a' ASIC is considered having its output ports to the right and up directions. Then for types b, c, and d, the orientation is simply a clockwise rotation of the previous, i.e. 'type b' ASIC is the clockwise rotation of 'type a' thus its output ports are to the right and down directions. 'Type c' has its outputs to the left and down directions and finally 'type d' has its output ports to the left and up directions. The reconfigurability of the nodes is enabled by the unique addressing scheme in conjunction with a simple yet robust fault-tolerant mechanism, which operates on top of the default routing algorithm adopted. The user can dynamically allocate unique address to any node in the network grid, or intentionally leave some nodes unaddressed. In addition to this, the fault tolerant mechanism can override the default routing algorithm on specific ASICs in the network and the user can manually decide the output of the corresponding ASICs, for the purpose of avoiding parts of the network that are found faulty. The addressing and fault tolerant operations are described below.

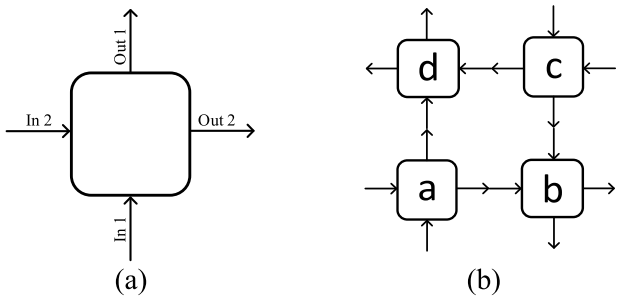

FIGURE 3. ASIC outer view: (a) Input / Output Channels and (b) Orientation type (a, b, c, d) according to the location of the ASIC on the network grid. The type is identified via an internal function in the ASIC.

A. NODE INITIALIZATION

Each incoming packet has a destination address for the node's router, but also each node has an internal flag named *address bit,* indicating the address state of the particular node. When the *address bit* is of logic '0', the node has no address and when a new packet is received, it will automatically copy and store the X and Y addresses from the packet to its address memory. After the address is stored, the *address bit* flag is set to '1' and never changes until the circuit is reset or the power is removed. In the case where the *address bit* is of logic '1' and a new packet is received by the corresponding node, the packet address is interpreted as a destination address, rather than an address to be stored, and so the router decides, based on the local address and the destination address, which is the appropriate output to forward the packet. The addressing algorithm relies on the fact that the routing decision making process is deterministic, based on the routing strategy adopted, and the sequence to program each node is predetermined beforehand in the computer device.

B. FAULT TOLERANCE

The fault tolerant mechanism enables the locking of packet output routing to one of the node's output ports, to force the packets to follow a pre-determined output port, and to bypass the default routing algorithm. This is done by sending a configuration packet to a particular destination node with the fault tolerant bits set to something other than the default case, depending on the desired output direction of that node. The fault tolerant bits are stored in the node's internal memory and this 'route-lock' is forced for all later packets. To change the fault tolerant case or remove the route-lock, a new packet must be sent to the particular node, with the fault tolerant bits having the value that corresponds to the desired case. This enables the user to circumvent faulty nodes or build a non-conventional rectangular-shaped network. In such a case the computer will need to pre-calculate the deadlock[1]-free/livelock[2]-free network architecture configuration. If a possible deadlock situation is predicted, then a

[1]The state where each node of a group waits for another node before it acts and thus the communication is stuck.

[2]The state where a group of nodes continuously repeat the same interaction without any useful work.

preceding packet can be sent to pre-configure a previous node to force its output settings in a way to eliminate the deadlock, send the packet and then change back the preceding node to the default routing.

C. ASIC CONSTRAINTS LIMITATIONS

The ASIC architecture has to address various requirements/constraints in order to be part of MSFs that operate at microwave frequencies.

The *operation frequency* of the MSF restricts the maximum size of the MSF unit-cell and subsequently the maximum size of the ASIC. Communication frequencies for the 5G era are ranging between 2.4 GHz and 60 GHz. Thus, the wavelength varies from 12.5 cm to 5 mm. For a reasonable absorption of an EM wave (in case of MSF absorber), there should be at least five unit-cells per wavelength. For example, at 60 GHz, the unit-cell size must be 1 mm $\times$ 1 mm. Within this space, apart from the ASIC, the unit-cell must accommodate signal routings, vias and board-to-board connectors. From the ASIC perspective, in this limited footprint, the architecture must be able to tune the complex impedance of the unit-cell, send/receive configuration packets, use fault tolerant solutions and generate acknowledgment packets to report its current state.

The *package* of the ASIC affects its operation as well. A typical wire-bond package introduces additional parasitics, especially inductance, because of the gold wires that are used to connect the ASIC pads to the package. The parasitics significantly reduce the performance of the RF LEs. To accommodate for this issue, the ASIC is packaged using the Wafer Level Chip Scale Package (WLCSP) technology. This packaging method has the ASIC pads filled with solder balls at the wafer and directly connects them to the PCB. Following the package decision, the available I/O pins of the IC are restricted. For example, a 2 mm $\times$ 2 mm ASIC in a 0.18 um CMOS process technology, allows the maximum of 25 I/O pins when using WLCSP. Of those 25 pins, twelve are needed for asynchronous communication (request, acknowledge, data), two for global hard and soft reset and four for analog / digital ground and supply. Serial transmission has the benefit of taking less area on-chip however, as it is apparent from the results in Section IV, the delays from the input and output circuits dictate the overall delay of the chip. The *cost* of the ASIC must be kept low enough to be applicable to realistic applications. Taking as an example the 60 GHz MSF absorber, to cover a surface of 1 m $\times$ 1 m, one would require 1 million unit-cells / ASICs since the size of every unit-cell needs to be 1 mm $\times$ 1 mm at maximum. Assuming every unit-cell cost half a dollar to be fabricated, then the MSF cost is 500 thousand dollars, not counting the connectors and the frame needed to keep the boards in place! Several precautions have been accounted to keep the cost of the ASICs as low as possible. First, each ASIC can operate as a standalone device without the need of extra, external components besides the gateway. A gateway device is required for providing configuration packets in the network, but each gateway in the system can provide power and operate hundreds of ASICs. Second, the ASIC has only one design to allow mass manufacturing and have many replacements in case of faulty ASICs that have to be replaced. The ASIC can dynamically identify its orientation in the grid according to its stored address and adjusts the router decisions according to its orientation. Finally, the semiconductor technology selected is mature and relatively low-cost, and at the same time provides decent impedance tuning range.

III. CONTROL CIRCUIT

In this section, the control circuit incorporating the unique properties described in Section II is proposed. The block diagram of the control circuit is given in Fig. 4. It mainly consists of *Input / Output Interfaces* for receiving and transmitting configuration packets respectively, the *Mode Decision* block for identifying/selecting the operation, according to the received information and the previous state of the control circuit, and the *Router* for deciding the output port for the configuration packet. The *Analog Parameters* block is not part of the control circuit architecture but is included in Fig. 4 for the sake of completeness, illustrating the complete chip architecture. This block is responsible for configuring the impedance of an MSF unit-cell, thus it consists of DACs and RF impedance LEs. The rest of the blocks which are described in details later in this section, include the buffers (*IN BUF* and *OUT BUF*) for pipelining the system, the memory blocks (*ADDRESS MEMORY* and *PAYLOAD MEMORY*), the global reset signals (*HARD RESET* and *SOFT RESET*) and the packet acknowledgment generation function (*Packet ACK*).

A. ASYNCHRONOUS ROUTING

The blocks/functions use asynchronous circuits and communicate via handshake. Although synchronous digital circuits are by far predominant in control circuit designs, going the asynchronous approach is this situation is necessary because the issues accompanied the clock tree synthesis cannot address the strict requirements of the programmable MSF system. In a synchronous ASIC solution, one would require a clock tree with scalability adjustments and at the same time, satisfy the requirements of flexible surfaces, conforming walls of irregular shapes. Furthermore, the clock skew needs to be well controlled to prevent setup and hold time violations. Given that the ASIC's design is *generic* and can be used in many applications and types of boards, the clock tree synthesis becomes a very challenging, if not an impossible task, given the variability of the structure. On the other hand, with asynchronous circuits, the *scalability* is becoming as easy as connecting boards together. Then, it is a matter of the addressing scheme to provide the valid addresses to the newly connected nodes. Furthermore, the *power consumption* for this type of application favors asynchronous circuits because during static conditions, which an MSF operates for the majority of the time, an asynchronous control circuit only consumes leakage current, whilst a synchronous control

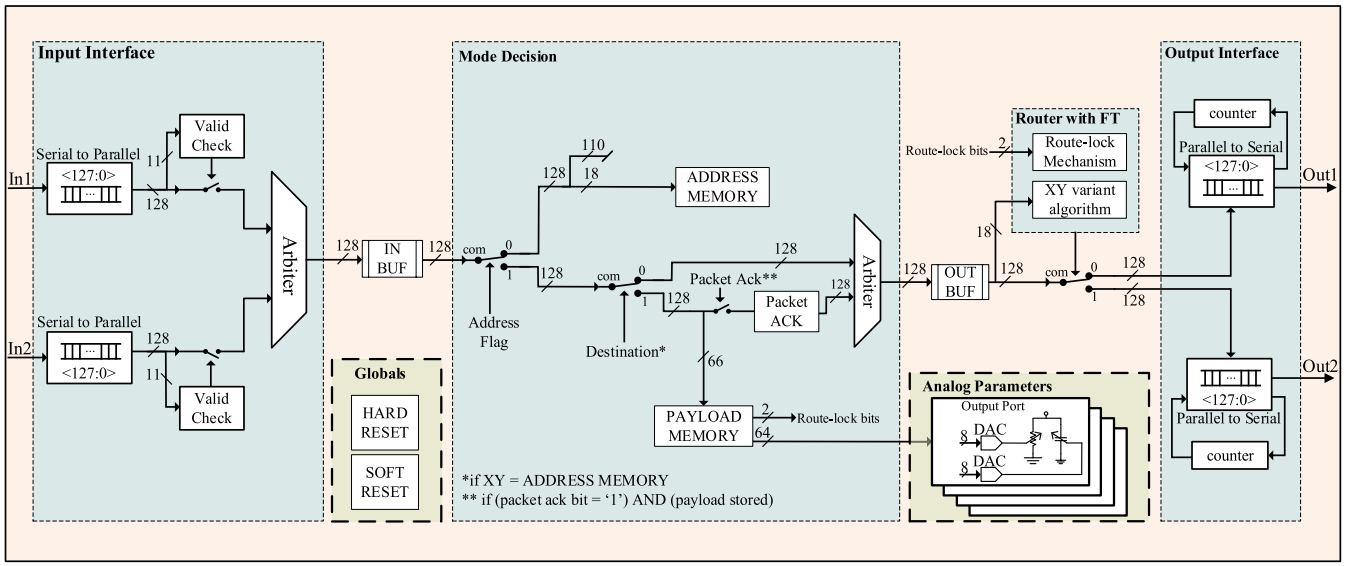

FIGURE 4. Top-Level Block Diagram of the control circuit. The circuit has two serial inputs and two serial outputs. All internal circuits and blocks communicate asynchronously using handshake and the main functions (Input/output Interface, Mode Decision and Router with FT) operate in parallel using pipeline. The Analog Parameters consists of eight digital-to-analog converters and four loading elements optimized for adaptive MSF applications. The Global signals are used to reset all signals to their starting state and erase the data stored in the address memory and/or the payload memory.

circuit consumes at every clock cycle. Attempts for fair comparison between asynchronous and synchronous NoC systems is presented in [35] and [36]. Special circuits or techniques can be adopted on a synchronous control circuit to turn off certain circuits when not in need, but this significantly increases the area on chip and the overall complexity of the control circuit. Another major constraint for synchronous designs, is the *EM noise* generated during switching activity. The switching can undermine the intended operation in the case of an MSF absorber. The MSF absorber, as the name suggests, absorbs EM waves, so transmitting broadband clock signals may defeat the purpose of hiding an object electromagnetically. Though there are emissions in the asynchronous case as well, these are fewer since the transitions are more evenly spread and are of lower amplitude. In addition, the switching only happens in the portion of the circuit that is being programmed, rather than on the entire surface. Despite the clear advantages of using asynchronous circuits, there are some trade-offs in designing asynchronous. Every function exploiting the asynchronous circuit implementation is more complex that the equivalent synchronous implementation. This is due to the additional handshake control circuitry that is needed. Unfortunately, the EDA/CAD tools available are not mature enough to allow easy and efficient high-level programming of asynchronous circuits. An elaboration in asynchronous circuit design can be found in [37] while recent advances are presented in [38] and [39].

B. BUILDING BLOCKS

Each building block is a standalone function and due to the asynchronous operation and design, the control circuit can be optimized at block level without affecting the rest of the building blocks. Moreover, the building blocks can be easily connected due to the asynchronous operation, offering a 'plug-and-play' approach to design.

1) INPUT INTERFACE

The *Input Interface* block is connected to its neighboring *Output Interface* blocks and receives configuration packets. It incorporates two 128-bit asynchronous serial-to-parallel converters to satisfy the pin limitations and also prepare the received packets for parallel processing in the rest of the circuit. Once a bit is received from one of the inputs, a new packet is formed to the corresponding serial-to-parallel converter. All bits are moved one position to the right, with the rightmost bit being dropped. The control circuit identifies a valid packet by checking the 'sequence' of bits at the first eleven bits of the formed packet. In case it matches to the internal, pre-determined sequence stored in the *Input Interface*, the packet is determined as valid and is sent in parallel to the input buffer (*IN BUF*) for further processing, otherwise the serial-to-parallel converter requests the next bit. Finally, the *Input Interface* incorporates an asynchronous arbiter to avoid collision of incoming valid packets from both inputs, at the same time. In such a case, the arbiter will arbitrarily choose the packet that goes in the control circuit for processing. The packet that is left to the arbiter's input, waits until the first packet is processed at the *Mode Decision* block and it then proceeds through the arbiter to the *IN BUF* block.

2) OUTPUT INTERFACE

The Output Interface transmits the packet to the output port. It 'handshakes' with the Input Interface block of the receiving-end control circuit. To send the packet to the output, two 128-bit asynchronous parallel-to-serial converters, one

at each output, align the bits to be sent in the correct order. A counter is used per parallel-to-serial converter to count the number of bits that have left the control circuit. Due to the asynchronous handshake operation between the two control circuits, when an acknowledge signal comes back to the Output Interface block, the corresponding counter adds one up to the count and keeps track of the number of outgoing bits. Also, in every cycle, the parallel-to-serial converter shifts to the right by one bit and the rightmost bit is being sent out. When all bits are sent, the counter is reset, and the parallel-to-serial converter prepares the next available packet for output.

3) ROUTER

The *router* block decides the output port of the processed packet. The authors in [40] and [41] propose two routing algorithms that can be used in this architecture, the Loop-Free-Algorithm (LFA) and the Reliable Delivery Algorithm (RDA). The former, uses the XY algorithm until a faulty channel is detected in its path. Then, it switches to YX routing and continues. When another faulty channel is detected, it switches back to the XY routing. The same process is repeated, and results showed improved delivery rates compared to other XY variants. However, this routing algorithm is prone to livelocks in the presence of faults. At this situation, a version of the turn model can be used on top of the XY-YX to break potential cycles. The latter (RDA) employs variants of the XY and YX to provide two disjoint paths between every controller in the network. When a fault is detected in a node, the packet is sent to the other output of the node prior to the faulty one. The paths are selected based on the location of the destination node. For this first implementation, a variant of the XY algorithm proved to be effective. In [42] the authors verified the XY variant routing algorithm on a Manhattan topology with edge wraparounds. The normal XY algorithm routes the packet to the 'right' until it reaches the X destination address and then continues 'up' until the packet reaches the Y destination address. The variant in this algorithm is to stop sending to the 'right' once the destination address is (X-1) and starts going 'up' until it reaches the Y destination. Then, it goes once to the 'right' to reach its destination. In case it cannot go 'up' directly at the (X-1) due to the Manhattan topology, it will move once to the 'right' and then continue as usual on the 'up' direction. In case it cannot go 'right' directly then it will move 'up' once, then move to the 'right' once and finally, it will go 'down' once to reach the destination node. The adoption of the variant algorithm instead of the normal XY, proved to have better performance and fewer deadlocks when the nodes form a Manhattan with edge-wraparounds network topology. The XY-variant algorithm operates in conjunction to the route-lock mechanism whereby if it is enabled, the output port is designated without running the algorithm. This procedure saves time and power consumption and it is used to avoid possible faulty/damaged nodes.

4) MODE DECISION

The *Mode Decision* block enables the appropriate operation and decomposes the packet for further processing if required. The *Mode Decision* block guides the necessary bits to the appropriate path in the control circuit according to the operation which is calculated according to the previous state of the control circuit and/or the processed packet. There are three main operations namely 'address mode', 'route mode' and 'store mode'. The 'address mode' is checked first and is enabled when the address flag is '0' and in such case, the address from the packet is stored in the address memory of the control circuit. The (X) and (Y) coordinates of the address consist of nine bits each, thus each node can have one of 2^{18} possible addresses. Note that the network can be further extended to more than 2^{18} nodes by having multiple gateways being part of the grid. Since the gateways have full computing capabilities, they can be inserted at key locations in the grid and when they receive a packet, they can modify its contents and then send it again to continue further in the network. The 'route mode' is enabled when the received packet has a different destination address than the corresponding control circuit's, and in that case the whole packet is send to the *OUT BUF*. Once the router decides the output port, the packet is moved to the corresponding *Output Interface* block. Finally, the 'store mode' is enabled to store the payload in the memory, in case the destination address bits match with the address stored in the control circuit. During 'store mode', the control circuit can also enable the '*Packet ACK*' function, whereby the control circuit generates a different kind of packet in order to report its payload to a gateway. The generated packet travels in the network grid as a normal packet with its destination address such that it will pass through the destination gateway. As an example, if the destination address is X = '111111111' and Y = '111111111' then the packet will go to the top right corner where a gateway can be easily placed and retrieve this packet.

5) MEMORY BLOCKS AND BUFFERS

The control circuit stores its address, the DAC bits and the fault tolerant bits. The allocated memory for the address (*ADDRESS MEMORY*) is 18 bits (9 for the X-address and 9 for the Y-address). The allocated memory for the DAC bits and fault tolerant bits (*PAYLOAD MEMORY*) is 66 bits because there are eight, 8-bit DACs and two fault tolerant bits. The two fault tolerant bits enable three cases. The first is the normal operation without route-locks, the second is with route-lock on output 1 and the third is with route-lock on output 2.

There are two buffers in the control circuit, located before and after the *Mode Decision* block. If the *Mode Decision* block is busy, then the *IN BUF* holds any valid input packet for the duration of the *Mode Decision* operation. During this time, the *IN BUF* sends acknowledge signal to the *Input Interface* block enabling parallel operation between the *Input Interface* block and the *Mode Decision* block. Note that

the *Input Interface* can receive two packets simultaneously and the arbiter decides which packet moves to the *IN BUF* block, in order to avoid losing any packets and also enable parallelism that increases the throughput of the control circuit. The *OUT BUF* stores the packet that is entering the *Router* block to enable parallel operation between the *Mode Decision* block and the *Router* block. Moreover, the *Output Interface* block can process two packets simultaneously. Thus, the control circuit, can have up-to six packets being processed at the same time, two of them in the *Input Interface* block, one in the *Mode Decision* block, one in the *Router* and two in the *Output Interface* block. Note that due to the asynchronous operation, even if the preceding stage finishes earlier, there will not be any conflicts in the signals, just like it would happen in the synchronous design case but rather, the preceding stage will hold and wait for acknowledge signal from its successor stage to continue.

6) GLOBAL RESET OPERATIONS

The reset operation is necessary in asynchronous circuits so it ensures that all state-holding cells are in a known state during startup. If this is not guaranteed then some of the state-holding elements may switch on to a state where they cannot process any data and thus the control circuit will be stuck. In this architecture the global reset operations trigger all connected control circuits at the same time because they are hardwired on all of them on the PCB. The available global reset operations are two:

1) *HARD RESET*: The *HARD RESET* is used to erase all stored data in the control circuit (both *ADDRESS MEMORY* and *PAYLOAD MEMORY*). Also, it stops all operations and all logic cells of all control circuits go to their starting/reset state.
2) *SOFT RESET*: The *SOFT RESET* stops all operations of all control circuits and erases the DAC bits (*PAYLOAD MEMORY*). The address of the control circuit is not affected (*ADDRESS MEMORY*). This way, the addressing of the control circuits can be performed one time only. Once the addressing is finished, then the soft reset is used for resetting the network's data.

C. PACKET FORMAT

The control circuit can identify and process two types of packet formats namely *routing packets* and *ACK packets*. *Routing packets* contain the payload of a specific control circuit and move in the network grid until they reach the destination control circuit and store the payload to its memory. *ACK packets* are only generated by the control circuits and their destination can only be a gateway. They consist of the payload stored in the control circuit that generated the ACK packet and its address.

The format of the *routing packet* is shown in Fig. 5(a). It consists of the header followed by the payload. The header is further divider to the start sequence, the fault tolerant (FT) bits and the address. The separation bits are included in

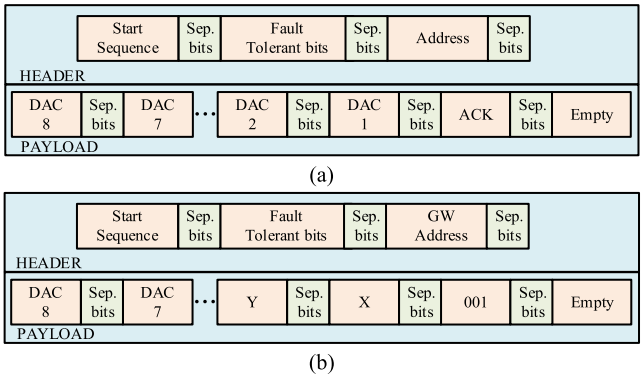

FIGURE 5. Two packet formats processed by the control circuit. (a) the routing packet which delivers the configuration data to the ASICs in a network grid and (b) the ACK packet which is generated by an ASIC to report its payload to the controlling system.

between the words to separate them and prevent the 'start sequence' to be formed anywhere else in the packet besides the first 11 bits. The 'start sequence' is the binary number $(10101010101)_2$. This sequence is ensured that it will not be found anywhere else in the packet besides the beginning of the packet since it consists of eleven bits while the following largest word consists of nine bits and all words are separated by the separation bits $(00)_2$ which break the $(10101010101)_2$ sequence. The FT bits are exploited by the gateway to force lock-on to one of the outputs of a specific control circuit. Two bits are required since there are three route-lock states (normal, lock on output 1 and lock on output 2). The address consists of two, 9-bit words, separated by a pair of separation bits. They contain the X and Y coordinates of the destination control circuit in the network grid. But, when a control circuit has not been addressed before, it will store this address in its *Address Memory*. This is due to the addressing mode explained in Section II. The payload contains the DAC bits, the ACK bits and the empty bits. The *routing packet* uses three bits for the *Packet ACK* operation. The first bit is used to enable the *Packet ACK* function. The second bit is used to point to the destination gateway address for the *ACK packet*. The user has two options which are pre-determined during the architecture phase of the control circuit. In case the bit is '0', the packet will travel to the bottom left corner of the tile. In case the bit is '1', the packet will travel to the top right corner of the tile. The third bit is exploited only by the gateway to distinguish between the *routing packets* and the *ACK packets*.

The packet format is slightly changed for *ACK packets* as depicted in Fig. 5(b). The address is replaced with the gateway address since the destination of an *ACK packet* is a gateway. Also, DAC2 and DAC1 bits are replaced with the address of the control circuit that generated the *ACK packet*, so that the gateway identifies the sender. In both packet formats, DAC1 and DAC2 are 9-bit words instead of 8 bits and the 9^{th} bit is only used during *ACK packets* so that the X and Y coordinates fit. The last change for the *ACK packet* is that the ACK bits are fixed to 001 for the gateway

TABLE 1. ASIC static current consumption.

Circuit	Static Current (μA) @ 1.8V
Control Circuit	14
DAC (x8)	176
LE (x4)	Negligible [33]
Total	190

to distinguish this type of packets from the *routing packets*. Finally, both packet types contain two empty bits at the end to allow improvements in later implementations of the control circuit. The length of the packets is 128 bits.

IV. PERFORMANCE EVALUATION

The ASIC architecture was designed using analog IC design flow in a 0.18 μm mixed-signal CMOS process technology. The control circuit and the DACs were designed using full-custom IC design in Cadence® and were simulated at transistor-level with Spectre® Simulation Platform and Spectre® AMS Designer. The RF LEs have been designed and simulated using Cadence® and verified in Keysight® ADS®. The achievable tuning range of the RF LE circuits are presented in [33]. All circuits operate at a 1.8 V power supply.

The digital asynchronous circuit blocks that form the control circuit have been custom-designed and simulated. As it is known, asynchronous logic cells (e.g. C-elements) are not included in the standard cell libraries provided by the foundries. These state-holding asynchronous cells are the backbone of asynchronous circuit blocks by maintaining the synchronization of the circuits, since there is no clock signal. The asynchronous blocks were simulated at transistor-level, across corners and process variations to ensure they do not have any internal timing related problems. Once the blocks have been validated, a plug-and-play approach is adopted to implement the control circuit architecture and validate its operation.

Table 1 shows the static current drawn by the three main parts of the ASIC. As evident, the control circuit draws considerably less current compared to the DACs. Each DAC draws 22 μA and thus 176 μA for all eight DACs, whilst the control circuit draws 14 μA. The current drawn by the LEs is the leakage current from the gate of a few RF transistors and can be safely neglected, as stated in [33]. Thus, the total static current drawn by the ASIC is 190 μA. The power consumption from the computer and the gateway are not included since these devices can be turned off after they calculate and send the configuration data and also each user can use the digital device of its choice.

The delay and energy consumption of the control circuit is shown in Table 2. The latency of the control circuit varies according to the operation. However, due to serial communication, the variation is very small since the time to send / receive a packet is three orders of magnitude higher.

TABLE 2. Simulated at circuit-level control circuit delays and energy consumption per function.

Operation	Delay (μs)	Energy (nJ)
Packet Reception	2.150	7.424
Packet Transmission	2.150	21.186
Addressing	0.003	0.111
Storing	0.010	0.243
Routing without Route-locks	0.043	0.192
Put Route-lock	0.010	0.243
Routing with Route-lock	0.009	0.181
Acknowledgment	0.050	0.378

The control circuit, as described earlier, has three operations: addressing, storing and routing. Storing can be extended to storing with or without acknowledgement. Routing can also be extended to routing with or without route-locks whereas mentioned, route-locks are used to designate one of the two outputs of an ASIC. To reconfigure an ASIC, a series of operations have to take place. As an example, consider one reconfiguration cycle of one un-addressed ASIC. First, the ASIC receives a packet (Packet Reception) and stores the address from the packet to its memory (Addressing). Then, a second packet is received (Packet Reception) and its payload is stored in the ASIC (Storing). Assuming the gateway asked for acknowledgment, then the ASIC generates the packet acknowledgment (Acknowledgment) and the router decides the output port, assuming no route-locks are present (Routing without Route-locks) and the packet is sent out of the ASIC (Packet Transmission). Adding the delays from Table 2, the total delay is 6.556 μs and the total energy consumed is 36.684 nJ. The same example, without acknowledgment, would require 4.313 μs and would consume 14.928 nJ since only addressing and storing of two packets would require. Considering that the addressing is only performed one time at the beginning in a practical application, then most of the times, the ASICs will only have to store the updated payload, which implies that only one packet will be received, and one storing operation will be executed which reduces the reconfiguration delay and consumption even further. The route-locks, as mentioned, are used as a fault tolerant method in order to bypass faulty ASICs. In addition, the route-locks make the routing procedure faster and consume less energy. This is because the output decision algorithm stays idle and the output is determined by the route-lock. As evident in Table 2, routing with route-locks saves 0.034 μs and 0.011 nJ per packet. Considering thousands of packets may pass through an ASIC, these savings eventually become significant. To put a route-lock in an ASIC, the same procedure as storing is performed since the route-lock bits are part of the payload of a packet.

Fig. 6 presents signal transitions in a channel between two ASICs. The data exchanged is the binary sequence $(0101)_2$ represented in the dual rail asynchronous communication

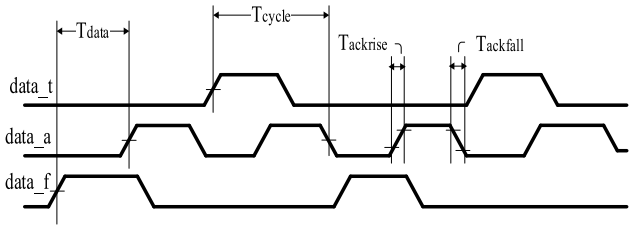

FIGURE 6. Signal transitions in the channel between two consecutive ASICs exchanging four bits. The word sent from the sender ASIC to the receiver ASIC is the binary sequence '0101'.

TABLE 3. Timings for two ASICs to exchange one bit of data.

Symbol	Description	Time (ns)
T_{data}	Time from data valid to ack	5.992
$T_{ackrise}$	Time for ack to rise	0.169
$T_{ackfall}$	Time for ack to fall	0.157
T_{cycle}	Time to complete a bit cycle	16.800

TABLE 4. Maximum rate for exchanging bits and packets and energy required to exchange one bit and one packet.

Description	Value	Units
Maximum Bit Rate	59.52	$Mbits/s$
Maximum Packet Rate	465	$kpackets/s$
Energy Consumption Per Bit	223.5	pJ/bit
Energy Consumption Per Packet	28.61	$nJ/packet$

protocol. Every bit is represented using two signals, namely '*data_t*' for the actual value of the bit and '*data_f*' for the complementary value of the bit. The '*data_a*' signal represents the acknowledge signal that acknowledges the previous ASIC on the reception of data. The timings for the control circuit to react to input signals are presented in Table 3. The ASIC needs 5.992 ns to receive valid data and respond with acknowledge signal. The rest of the time for a cycle completion, is for returning the signals back to their reset state. The rise and fall times of the signals are 0.169 ns and 0.157 ns respectively, taking into consideration the pads and I/O circuits.

Table 4 shows the maximum rate at which the ASICs can exchange bits and packets, and the energy consumed per bit and packet. The energy consumption is divided to the energy of the transmitting ASIC and the receiving ASIC. The transmitting ASIC consumes 165.5 pJ to send/transmit one bit while the receiving ASIC consumes 58 pJ to receive one bit. Extending this calculation to packets, the transmitting ASIC consumes 21.186 nJ and the receiving ASIC consumes 7.424 nJ, giving a total of 28.61 nJ consumption of energy in the channel for the exchange of one packet.

V. ADAPTIVE NETWORKING

The following five scenarios demonstrate the major capabilities of both the addressing and routing strategies and at the same time exploit the fault tolerant (route-locking) capabilities. Scenario 1 presents tile-to-tile interconnections and automatic addressing of newly added nodes/tiles. Multiple intermediate gateways can be connected in various locations to speed up the communication. Scenario 2 expands upon Scenario 1 by considering faulty nodes in the network. The faulty nodes can be bypassed using route-locks while the rest of the network is fully functional. In Scenario 3, an hourglass-like network is described, using route-locking to block the paths to both inputs of the unused nodes. This network topology can be most efficient when used on a flexible surface. In Scenario 4, the nodes are addressed using only even addresses, so that the input / output ports are of the same direction. This concept benefits flooding communications where all nodes of one tile must have the same payload/EM settings. Finally, Scenario 5 presents a flexible programmable MSF utilizing the proposed ASICs, in a realistic environment, showing its potentials in terms of reconfiguration delay, energy efficiency, storage requirements and computational complexity.

A. SCENARIO 1 – MULTI-TILE ARCHITECTURE

Scenario 1 considers a 2×2 tile-array (Fig. 7) connected in a Manhattan configuration whereby each consecutive row and column change directions. Each tile consists of 4×4 nodes. The connectors at the edges of the tile are used to connect the tiles together. They also connect the edge nodes forming edge-wraparound connections. The gateway devices provide the configuration packets to the nodes. Note that, multiple gateway devices can be connected to the network through the connectors, for example, *GW 2* is connected at the input of node (4,4). This can make the system faster by having, for example, *GW 2* sending packets to the tiles at the top row and *GW 1* to the tiles at the bottom row. The most reliable case is to have a gateway per tile providing to the nodes configuration packets and supplying them with power. This approach allows for scaling the system up to its memory limitations by simply connecting the tiles together. Furthermore, each gateway can be responsible only for the chips in its tile, which leads to parallel configuration of all tiles and thus reconfiguration of the whole metasurface at the same time as the reconfiguration of one tile. Introducing multiple gateways in the metasurface system significantly reduces the system's latency, trading off power consumption due to the increased resources committed.

B. SCENARIO 2 – NETWORK WITH REGIONAL DEFECT

In this scenario, the same 2×2 tile-array from Scenario 1 is considered. However, to illustrate the capabilities of the route-locking mechanism, consider the following hypothetical case. The MSF is hit by an object that broke some of the ASICs from the network. This is depicted in Fig. 8 whereby the ASICs in the gray shaped area are considered faulty. Even in this case, the rest of the network can still be functional by using route-locks to completely avoid sending configuration packets to the faulty area of the network. As seen in Fig. 8, the outputs that lead to a faulty node (e.g. (0,0) right output, (0,1) right output, (1,3) down output, (4,0) left output etc.) are

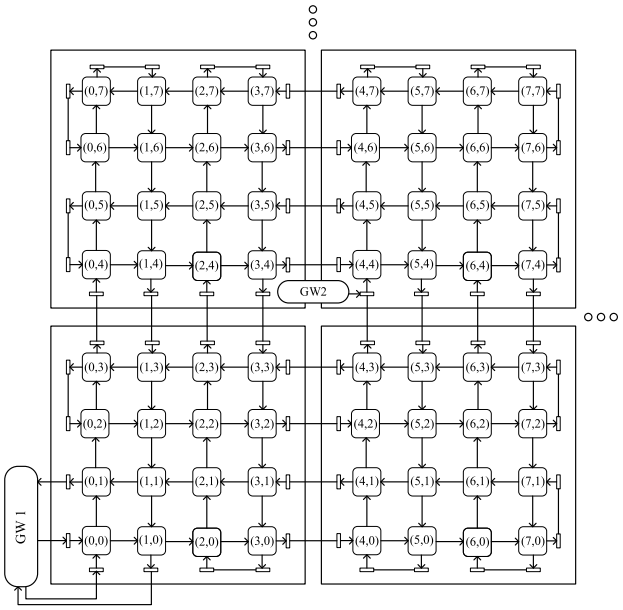

FIGURE 7. Scalable wall consisting of four tiles for a tile-to-tile interconnection. The wall consists of 64 nodes. The topology is Manhattan with edge-wraparounds.

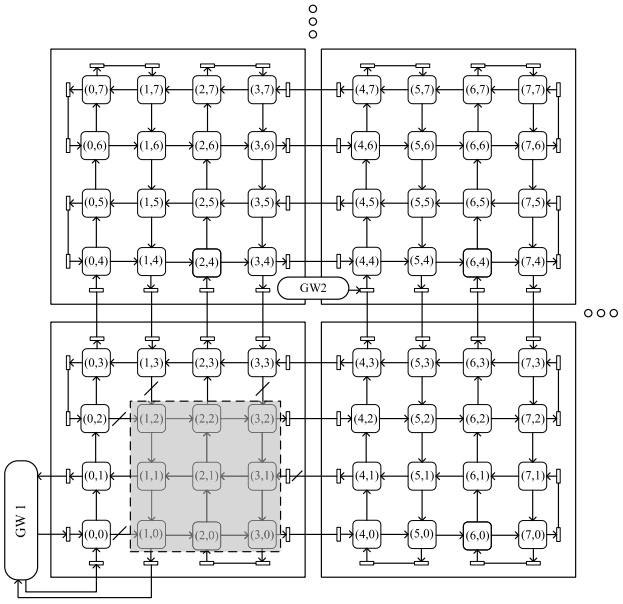

FIGURE 8. Network with damaged nodes (grey shaped area) that exploits route-locks to send packets around the faulty part of the network and provide configuration data to the rest of the nodes.

blocked. These ASICs are not using the default routing algorithm to decide their output port but rather they immediately output to the unblocked output port.

This increases the robustness of the system because it can provide functionality even in case of broken parts of the MSF without the need for changing any mechanical parts. Unavoidably, the tolerance allowed by the network depends on the number of faulty nodes and their location on the board. But, the 'key' nodes of the network can be protected by other means (external protection) and keep them safe from environmental dangers.

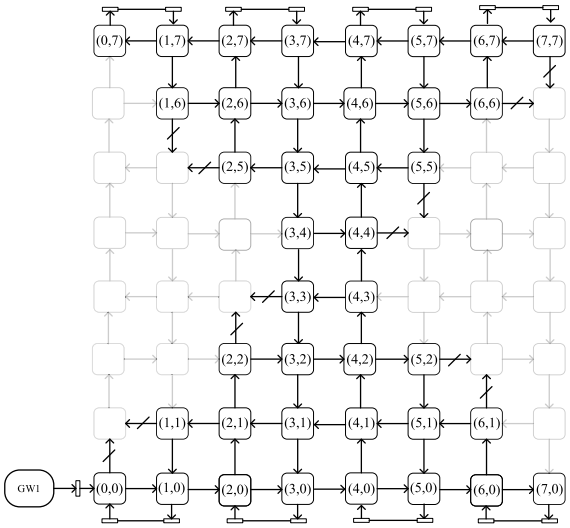

FIGURE 9. Hourglass configuration network suitable for flexible surfaces. The route-locks, block all outputs that end-up in the un-addressed nodes. The un-addressed nodes can be dynamically added to the network by removing their route-locks and sending them addressing data.

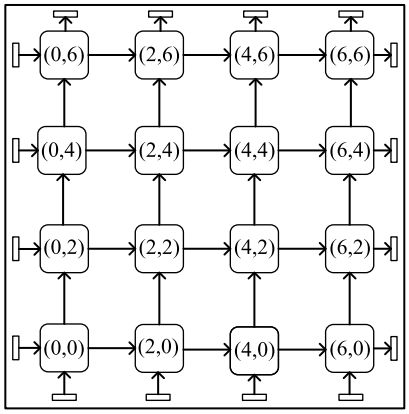

FIGURE 10. Even-only addressing. This changes the default XY variant routing algorithm to the normal XY routing algorithm and all ASICs are of 'type a' with outputs at the 'right' and 'up' directions.

C. SCENARIO 3 – FLEXIBLE SURFACE

The second scenario exploits the capabilities of the unique addressing scheme and route-locking. Fig. 9 shows an example topology with the shape of the hourglass. The output ports that lead to un-addressed nodes are blocked. The un-addressed nodes can become part of the network by simply removing the route-locks that block the inputs of the un-addressed nodes and sending them addressing data. This topology can be reconfigured to adapt to the size and shape of flexible surfaces without addressing and powering any unnecessary nodes. Unavoidably, all nodes consume static energy, but this is relatively small especially when using techniques or communication protocols that minimize static power consumption, as in the case of the proposed architecture. The network of Fig. 9 is only an example network with one tile. Numerous networks can be configured by connecting multiple tiles. Combining this ability with flexible surfaces, allows the metasurface to be applied on geometries that were

not accessible with a rigid surface. The flexible metasurfaces allow stretching and bending, thus enabling extra functionalities to existing metasurfaces by being able to wrap-around non-flat surfaces.

D. SCENARIO 4 – EVEN-ONLY NETWORK

In this scenario, a 4 × 4 array is considered (Fig. 10). However, the addressing of the nodes uses only even numbers. This is feasible because of the dynamic addressing mechanism. As mentioned, once an un-addressed node receives a packet, it stores the destination address to its memory. By addressing only with even addresses, the network is formed with only one orientation type ('type a' in the example of Fig. 10). Similarly, by addressing with odd addresses, the network will have only 'type c' oriented nodes. This affects the internal routing algorithm in case it uses the (X-1) address. In the adopted routing algorithm, as explained earlier, the XY routing algorithm is used, but the packets start moving 'up' once they reach the (X-1) address, as an optimization for the Manhattan topology. With this scenario, the routing algorithm is becoming a typical XY because none of the nodes has (X-1) address. This topology leads to simpler and faster setting adaptation, but lacks programmability because the packets can only move to the right and up directions.

E. SCENARIO 5 – METASURFACE ON DRONE

The last scenario presented aims to illustrate the capabilities of the proposed architecture in implementing programmable MSFs for realistic structures, such as a small camera drone. An illustration of the envisioned network grid is shown in Fig. 11, on a schematic view of a typical camera drone. Note that the dimensions of the drone are not shown on purpose in Fig. 11 since the system can be easily expanded to cover larger or smaller surfaces. Also note that the shape of such a surface is not flat. Despite the shape and structure, the chip-network can be constructed on a flexible MSF substrate and therefore being able to bend where necessary to cover the whole surface. The number of chips required to cover this particular surface is dependent on the unit-cell size. As an example, consider that the main body of the drone requires 64 chips to cover the surface with the unit-cell structure from Fig. 1. The unit-cells are distributed on the surface as shown in Fig. 11 to the right. Regarding the gateway connection to the MSF, there are two approaches that can be adopted. The first approach is to temporary connect the gateway to the MSF, pass the required configuration settings to the unit-cells and then disconnect it. The second approach, which is recommended since it is the most adaptive of the two, is to connect a very small gateway device on the side of the camera drone and keep it connected to the MSF. The configuration data is stored in the gateway beforehand and through the drones remote for example, the gateway can receive directives and convey the appropriate data from its look-up tables into the network grid. This approach enables the option of real-time programmability through wireless communication but at the same time, the complexity of the system is kept

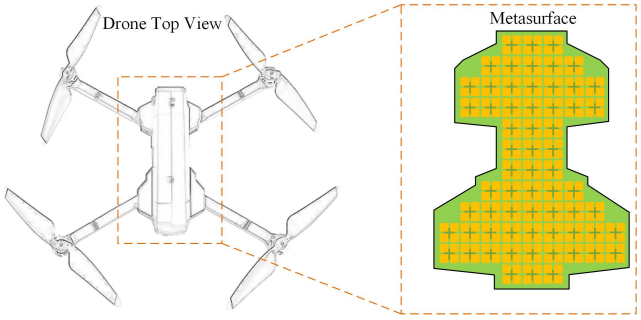

FIGURE 11. Realistic application of a programmable MSF utilizing the proposed integrated circuits. The MSF is constructed on a flexible substrate and placed on top of the drone's main body. The power and ground connections are shared with the drone's internal battery.

TABLE 5. Delay and consumption calculations for one reconfiguration cycle including addressing for various networks.

Network	Delay (ms)	Static Power (mW)	Energy (μJ)
1 ASIC	0.004	0.342	0.015
8 × 8 Manhat. (Fig. 7)	2.346	21.888	28.162
Hourglass (Fig. 9)	3.199	13.680	20.711
36 × 36 Manhat.	399.587	443.232	2632.619
Drone (Fig. 11)	2.320	21.888	28.073

TABLE 6. Delay and consumption calculations for one reconfiguration cycle without addressing for various networks.

Network	Delay (ms)	Static Power (mW)	Energy (μJ)
1 ASIC	0.002	0.342	0.008
8 × 8 Manhat. (Fig. 7)	1.173	21.888	14.085
Hourglass (Fig. 9)	1.600	13.680	10.358
36 × 36 Manhat.	199.798	443.232	1316.395
Drone (Fig. 11)	1.160	21.888	14.041

fairly simple. The power supply of the metasurface and the gateway can be shared with the drone's battery after voltage regulation, since the consumption of the metasurface and the gateway is relatively small. The network formed in this system utilizes the XY variant routing algorithm and the route-locking mechanism proposed in Section III and creates a path that can reach all nodes in the network grid. To set the RF impedance of all chips, the gateway has to send in the grid 64 configuration packets, one per chip. Each packet will move in the grid according to the XY variant algorithm, until it reaches its destination chip.

F. CALCULATED PERFORMANCE

The configuration delay and power consumption of all scenarios are presented in this part and include the number of 'hops' of each packet until it reaches the destination node. Furthermore, the calculations take into account the operations that have been performed in each chip, for example when a packet is send from one chip to the other, the sender chip

consumes 7.424 nJ of energy while the receiver chip consumes 21,186 nJ. The delay for sending a packet from one chip to another is 2.15 μs since both chips share the same time when exchanging packets. Table 5 shows the calculations for delay, static power consumption and energy consumption for one reconfiguration cycle including addressing for Scenarios 1, 3, 4 and 5. Scenario 2 is left out of the calculations. Similarly, Table 6 shows the same networks assuming that the ASICs are already addressed and only their payload is updated. From the current consumption perspective, the static current consumption is considered at the steady state, when all ASICs received address and have stored their payload. This way, the consumption of the MSF is calculated during static conditions, e.g. when optimized to absorb EM waves from a specific angle for a long amount of time. The calculations consider the number of 'hops' of every packet, the routing path to reach the destination ASIC and the operations that are enabled per ASIC. In the following calculations, acknowledgment packets are not considered. Also, there is only one gateway that sends packets in the network, however multiple gateways on multiple locations can be used to feed packets in the grid faster. Doing so will further exploit the advantages offered by pipelining the ASICs. The results from Table 5 and Table 6 show that the designed networks reconfigure metasurfaces in a matter of milliseconds consuming nanojoules or microjoules of energy. Once the network is reconfigured, the static current drawn by the ASICs is a few milliamperes, easily provided by FGPA devices and controllers. The 36 $\times$ 36 Manhattan network which is included in the calculations, can be considered as a typical tile having a size of 36 cm $\times$ 36 cm, with unit-cells of 10 mm $\times$ 10 mm for example. By including connectors at the edges of the tile, one can connect the ASICs at the edges with ASICs from other tiles expanding the network with tile-to-tile connections. The added ASICs dynamically receive addresses with the addressing algorithm and become part of the same network. This architecture achieves high scalability adaptations since every tile can use its own gateway to provide power to its ASICs.

VI. DISCUSSION

The ASIC architecture presented in this work establishes both the theoretical and practical limitations of using integrated circuits to enable programmability on MSF systems. The architecture is intentionally kept generic to provide the researchers and engineers with a multipurpose platform, which can be used for their ASIC-based programmable MSF enablers. Then, each group can optimize the system according to their needs. For example, some might be willing to explore the largest available tuning range by exploiting technology processes with better RF properties (e.g. Germanium). Others might be willing to implement intelligent networks with machine learning algorithms etc. However, the limitations presented in this work affect all these approaches and should be taken into account during the early stages of the design. The calculations derived from the transistor-level simulations already show a great improvement in both reconfiguration delay and power consumption, compared to the diode-based programmable MSF approaches. Furthermore, the 0.18 μm process technology used, is very reliable in terms of its models, which enhances the results derived in this work.

The architecture described in this work can be generalized for controlling various other applications besides metamaterials. For example, multi-chip board scale neural networks can be designed where the individual cells contain synaptic parameters rather that RF loading parameters. Or a scalable ISFET array system [43] for measuring ion concentration by having ISFET pixels [44] rather that RF loading parameters, and analog-to-digital converters instead of digital-to-analog converters. This architecture allows many systems to exploit the advantages of low static power, relatively low cost, low EM noise, high scalability, small size, and fast board-to-board communication systems which are offered by this architecture. The results gathered in this paper show that there are two limiting factors in this architecture that have potential to improve in later implementations. From the power consumption perspective, it is evident that the majority of the current drawn during static conditions is consumed at the DACs even though the resistor string architecture is considered low-power in general. The second limiting factor is the delay of receiving/transmitting a packet. The delay in this case is three order of magnitude (10^3) larger compared to the rest of the internal asynchronous functions. However, as mentioned during the description of the control circuit architecture, the internal blocks can be optimized at block level due to the adoption of the asynchronous digital circuit design. Thus, the reception and transmission circuitry can be optimized for higher speed and upgraded without affecting the rest of the building blocks.

VII. CONCLUSION

In this paper, the architecture of a custom ASIC for adaptive MSFs is presented. It is shown that a network can be formed on the back of MSFs through using this architecture, thus enabling a de-centralized programmability of each individual unit-cell. Circuit level simulations on a 0.18 μm process technology show that the ASICs exchange configuration packets within microseconds, consuming energy of the order of nanojoules. The ASIC implements a fault tolerant scheme, to increase its robustness. In order to increase its throughput, the architecture includes pipelining of the following tasks: packet reception, packet routing, payload storage and packet transmission. Furthermore, it can also generate acknowledgment packets to report payload reception at particular nodes of interest. The dynamic addressing mechanism used, makes the MSF system scalable by simply connecting tiles together with board-to-board connectors. The architecture is generic and any advances in the integrated technology can be adopted to improve the ASICs performance and decrease its size. Finally, the ASIC architecture overcomes all the specific challenges imposed by programable MSF systems, enabling software control of MSFs. The work presented here will lead

to the growth of a new generation of ASICs for board-to-board scalable networks for real-time impedance adaptation of MSFs. These ASICs will lead to improved performance on functionalities, such as EM absorption, anomalous reflection and beam control, that could be utilized in medical, telecommuting and radar applications. This work shows that ASICs are the future of programmable metasurface applications. As a direct follow-up to this work, the control-circuit and digital-to-analogue converters will be optimized for reconfiguring memristive devices instead of LEs, to set the complex impedance of each meta-atom. Also, other works include the optimization of the LEs to control metasurfaces for usage on satellite applications e.g. satellite signal receiver, and on telecommunication applications e.g. multi-angle anomalous reflection systems.

ACKNOWLEDGMENT

The authors thank their colleagues at the Holistic Electronic Research Laboratory (HERL), University of Cyprus, for constructive discussions regarding the system architecture. They are also grateful to their colleagues at the Computer Science Department, University of Cyprus, for the meetings and discussions regarding the networks.

REFERENCES

[1] N. Zilberman, P. M. Watts, C. Rotsos, and A. W. Moore, "Reconfigurable network systems and software-defined networking," *Proc. IEEE*, vol. 103, no. 7, pp. 1102–1124, Jul. 2015, doi: 10.1109/JPROC.2015.2435732.

[2] Y.-Z. Wang, Y.-P. Wang, Y.-C. Wu, and C.-H. Yang, "A 12.6 mW, 573–2901 kS/s reconfigurable processor for reconstruction of compressively sensed physiological signals," *IEEE J. Solid-State Circuits*, vol. 54, no. 10, pp. 2907–2916, Oct. 2019, doi: 10.1109/JSSC.2019.2933309.

[3] J. Wu, F. Li, Z. Chen, and X. Xiang, "A 3.89-GOPS/mW scalable recurrent neural network processor with improved efficiency on memory and computation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 12, pp. 2939–2943, Dec. 2019, doi: 10.1109/TVLSI.2019.2927375.

[4] H.-E. Kim, J.-S. Yoon, K.-D. Hwang, Y.-J. Kim, J.-S. Park, and L.-S. Kim, "A reconfigurable heterogeneous multimedia processor for IC-stacking on Si-interposer," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 589–604, Apr. 2012, doi: 10.1109/TCSVT.2011.2171209.

[5] S. Yin, P. Ouyang, S. Tang, F. Tu, X. Li, S. Zheng, T. Lu, J. Gu, L. Liu, and S. Wei, "A high energy efficient reconfigurable hybrid neural network processor for deep learning applications," *IEEE J. Solid-State Circuits*, vol. 53, no. 4, pp. 968–982, Apr. 2018, doi: 10.1109/JSSC.2017.2778281.

[6] B. Moons and M. Verhelst, "An energy-efficient precision-scalable ConvNet processor in 40-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 52, no. 4, pp. 903–914, Apr. 2017, doi: 10.1109/JSSC.2016.2636225.

[7] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 106–122, Feb. 2018, doi: 10.1109/TBCAS.2017.2759700.

[8] A. Psarras, J. Lee, P. Mattheakis, C. Nicopoulos, and G. Dimitrakopoulos, "A low-power network-on-chip architecture for tile-based chip multiprocessors," in *Proc. 26th Ed. Great Lakes Symp. (VLSI)*, May 2016, pp. 335–340, doi: 10.1145/2902961.2903010.

[9] H. Fang, A. Shrestha, D. Ma, and Q. Qiu, "Scalable NoC-based neuromorphic hardware learning and inference," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8, doi: 10.1109/IJCNN.2018.8489619.

[10] L. Zhang, "Transmission-reflection-integrated multifunctional coding metasurface for full-space controls of electromagnetic waves," *Adv. Funct. Mater.*, vol. 28, no. 33, pp. 1–9, 2018.

[11] X. Wan, Q. Zhang, T. Yi Chen, L. Zhang, W. Xu, H. Huang, C. K. Xiao, Q. Xiao, and T. Jun Cui, "Multichannel direct transmissions of near-field information," *Light, Sci. Appl.*, vol. 8, no. 1, pp. 1–8, Dec. 2019.

[12] L. Liang, M. Qi, J. Yang, X. Shen, J. Zhai, W. Xu, B. Jin, W. Liu, Y. Feng, C. Zhang, and H. Lu, "Anomalous terahertz reflection and scattering by flexible and conformal coding metamaterials," *Adv. Opt. Mater.*, vol. 3, no. 10, pp. 1374–1380, 2015.

[13] J. Zhao, Q. Cheng, J. Chen, M. Q. Qi, W. X. Jiang, and T. J. Cui, "A tunable metamaterial absorber using varactor diodes," *New J. Phys.*, vol. 15, no. 4, Apr. 2013, Art. no. 043049.

[14] K. Chen, Y. Feng, L. Cui, J. Zhao, T. Jiang, and B. Zhu, "Dynamic control of asymmetric electromagnetic wave transmission by active chiral metamaterial," *Sci. Rep.*, vol. 7, no. 1, pp. 1–10, Mar. 2017.

[15] X. Gao, W. L. Yang, H. F. Ma, Q. Cheng, X. H. Yu, and T. J. Cui, "A reconfigurable broadband polarization converter based on an active metasurface," *IEEE Trans. Antennas Propag.*, vol. 66, no. 11, pp. 6086–6095, Nov. 2018, doi: 10.1109/TAP.2018.2866636.

[16] D. S. Dong, "Terahertz broadband low-reflection metasurface by controlling phase distributions," *Adv. Opt. Mater.*, vol. 3, no. 10, pp. 1405–1410, 2015.

[17] G. Oliveri, D. H. Werner, and A. Massa, "Reconfigurable electromagnetics through metamaterials—A review," *Proc. IEEE*, vol. 103, no. 7, pp. 1034–1056, Jul. 2015, doi: 10.1109/JPROC.2015.2394292.

[18] T. Hand and S. Cummer, "Characterization of tunable metamaterial elements using MEMS switches," *IEEE Antennas Wireless Propag. Lett.*, vol. 6, pp. 401–404, 2007, doi: 10.1109/LAWP.2007.902807.

[19] S. R. Biswas, C. E. Gutiérrez, A. Nemilentsau, I.-H. Lee, S.-H. Oh, P. Avouris, and T. Low, "Tunable graphene metasurface reflectarray for cloaking, illusion, and focusing," *Phys. Rev. A, Gen. Phys.*, vol. 9, no. 3, Mar. 2018, Art. no. 034021, doi: 10.1103/PhysRevApplied.9.034021.

[20] S. E. Hosseininejad, K. Rouhi, M. Neshat, R. Faraji-Dana, A. Cabellos-Aparicio, S. Abadal, and E. Alarcón, "Reprogrammable graphene-based metasurface mirror with adaptive focal point for THz imaging," *Sci. Rep.*, vol. 9, no. 1, p. 2868, Dec. 2019, doi: 10.1038/s41598-019-39266-3.

[21] S. Savo, D. Shrekenhamer, and W. J. Padilla, "Liquid crystal metamaterial absorber spatial light modulator for THz applications," *Adv. Opt. Mater.*, vol. 2, no. 3, pp. 275–279, Mar. 2014, doi: 10.1002/adom.201300384.

[22] H. Yang, "A programmable metasurface with dynamic polarization, scattering and focusing control," *Sci. Rep.*, vol. 6, pp. 1–11, Oct. 2016.

[23] D. F. Sievenpiper, J. H. Schaffner, H. J. Song, R. Y. Loo, and G. Tangonan, "Two-dimensional beam steering using an electrically tunable impedance surface," *IEEE Trans. Antennas Propag.*, vol. 51, no. 10, pp. 2713–2722, Oct. 2003, doi: 10.1109/TAP.2003.817558.

[24] C. Mias and J. H. Yap, "A varactor-tunable high impedance surface with a resistive-lumped-element biasing grid," *IEEE Trans. Antennas Propag.*, vol. 55, no. 7, pp. 1955–1962, Jul. 2007, doi: 10.1109/TAP.2007.900228.

[25] F. Liu, A. Pitilakis, M. S. Mirmoosa, O. Tsilipakos, X. Wang, A. C. Tasolamprou, S. Abadal, A. Cabellos-Aparicio, E. Alarcon, C. Liaskos, N. V. Kantartzis, M. Kafesaki, E. N. Economou, C. M. Soukoulis, and S. Tretyakov, "Programmable metasurfaces: State of the art and prospects," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5, doi: 10.1109/ISCAS.2018.8351817.

[26] S. Abadal, T.-J. Cui, T. Low, and J. Georgiou, "Programmable metamaterials for software-defined electromagnetic control: Circuits, systems, and architectures," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 10, no. 1, pp. 6–19, Mar. 2020, doi: 10.1109/JETCAS.2020.2976165.

[27] C. Liaskos, A. Tsioliaridou, A. Pitsillides, I. F. Akyildiz, N. V. Kantartzis, A. X. Lalas, X. Dimitropoulos, S. Ioannidis, M. Kafesaki, and C. M. Soukoulis, "Design and development of software defined metamaterials for nanonetworks," *IEEE Circuits Syst. Mag.*, vol. 15, no. 4, pp. 12–25, 4th Quart., 2015.

[28] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "Realizing wireless communication through software-defined HyperSurface environments," in *Proc. IEEE 19th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2018, pp. 14–15.

[29] C. Liaskos, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "Viewpoint: Using any surface to realize a new paradigm for wireless communications," *Commun. ACM*, vol. 61, no. 11, pp. 30–33, Oct. 2018.

[30] A. Pitilakis, O. Tsilipakos, F. Liu, K. M. Kossifos, A. C. Tasolamprou, D. H. Kwon, M. S. Mirmoosa, D. Manessis, N. V. Kantartzis, C. Liaskos, and M. A. Antoniades, "A multi-functional intelligent metasurface: Electromagnetic design accounting for fabrication aspects," *IEEE Trans. Antennas Propag.*, vol. 69, no. 3, pp. 1440–1454, Aug. 2020.

[31] D. Kouzapas, C. Skitsas, T. Saeed, V. Soteriou, M. Lestas, A. Philippou, S. Abadal, C. Liaskos, L. Petrou, J. Georgiou, and A. Pitsillides, "Towards fault adaptive routing in metasurface controller networks," *J. Syst. Archit.*, vol. 106, Jun. 2020, Art. no. 101703.
[32] L. Petrou, P. Karousios, and J. Georgiou, "Asynchronous circuits as an enabler of scalable and programmable metasurfaces," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5, doi: 10.1109/ISCAS.2018.8351672.
[33] K. M. Kossifos, L. Petrou, G. Varnava, A. Pitilakis, O. Tsilipakos, F. Liu, P. Karousios, A. C. Tasolamprou, M. Seckel, D. Manessis, N. V. Kantartzis, D.-H. Kwon, M. A. Antoniades, and J. Georgiou, "Toward the realization of a programmable metasurface absorber enabled by custom integrated circuit technology," *IEEE Access*, vol. 8, pp. 92986–92998, 2020, doi: 10.1109/ACCESS.2020.2994469.
[34] R. J. van de Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters* (Springer International Series in Engineering and Computer Science), vol. 742, 2nd ed. Springer, 2010.
[35] M. Imai, T. Van Chu, K. Kise, and T. Yoneda, "The synchronous vs. Asynchronous NoC routers: An apple-to-apple comparison between synchronous and transition signaling asynchronous designs," in *Proc. 10th IEEE/ACM Int. Symp. Networks Chip (NOCS)*, Sep. 2016, pp. 1–8, doi: 10.1109/NOCS.2016.7579330.
[36] W. Jiang, D. Bertozzi, G. Miorandi, S. M. Nowick, W. Burleson, and G. Sadowski, "An asynchronous NoC router in a 14nm FinFET library: Comparison to an industrial synchronous counterpart," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2017, pp. 732–733, doi: 10.23919/DATE.2017.7927084.
[37] J. Sparsø, *Introduction to Asynchronous Circuit Design*. Lyngby, Denmark: DTU Compute, Technical Univ. Denmark, 2020.
[38] S. M. Nowick and M. Singh, "Asynchronous design—Part 1: Overview and recent advances," *IEEE Design Test*, vol. 32, no. 3, pp. 5–18, Jun. 2015, doi: 10.1109/MDAT.2015.2413759.
[39] S. M. Nowick and M. Singh, "Asynchronous design—Part 2: Systems and methodologies," *IEEE Design Test*, vol. 32, no. 3, pp. 19–28, Jun. 2015, doi: 10.1109/MDAT.2015.2413757.
[40] D. Kouzapas, C. Skitsas, T. Saeed, V. Soteriou, M. Lestas, A. Philippou, S. Abadal, C. Liaskos, L. Petrou, J. Georgiou, and A. Pitsillides, "Towards fault adaptive routing in metasurface controller networks," *J. Syst. Archit.*, vol. 106, Jun. 2020, Art. no. 101703, doi: 10.1016/j.sysarc.2019.101703.
[41] T. Saeed, C. Skitsas, D. Kouzapas, M. Lestas, V. Soteriou, A. Philippou, S. Abadal, C. Liaskos, L. Petrou, J. Georgiou, and A. Pitsillides, "Fault adaptive routing in metasurface controller networks," in *Proc. 11th Int. Workshop Netw. Chip Archit. (NoCArc)*, Oct. 2018, pp. 1–6, doi: 10.1109/NOCARC.2018.8541148.
[42] P. Kouvaros, D. Kouzapas, A. Philippou, J. Georgiou, L. Petrou, and A. Pitsillides, "Formal verification of a programmable hypersurface," in *Proc. Int. Workshop Formal Methods Ind. Crit. Syst.* in Lecture Notes in Computer Science, vol. 11119, 2018, pp. 83–97.
[43] N. Moser, J. Rodriguez-manzano, T. S. Lande, and P. Georgiou, "A scalable ISFET sensing and memory array with sensor auto-calibration for on-chip real-time DNA detection," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 2, pp. 390–401, Apr. 2018, doi: 10.1109/TBCAS.2017.2789161.
[44] N. Moser, L. Petrou, Y. Hu, and P. Georgiou, "An ISFET pixel with integrated trapped charge compensation using temperature feedback," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5, doi: 10.1109/ISCAS.2018.8351358.

LOUKAS PETROU (Member, IEEE) received the bachelor's degree in electrical engineering from the University of Cyprus, in 2016, and the master's degree in analogue and digital integrated circuit design from Imperial College London. He is currently pursuing the Ph.D. degree in electrical engineering with the University of Cyprus. His research interests include integrated circuit design and asynchronous digital circuits.

JULIUS GEORGIOU (Senior Member, IEEE) received the M.Eng. degree in electrical and electronic engineering and the Ph.D. degree from Imperial College London, in 1998 and 2003, respectively.

For two years, he worked as the Head of micropower design with a technology start-up company, Toumaz Technology. In 2004, he joined Johns Hopkins University as a Postdoctoral Fellow, before becoming a Faculty Member with the University of Cyprus (since 2005). He is currently an Associate Professor with the University of Cyprus. He is a member of the IEEE Circuits and Systems Society and the IEEE Circuits and Systems Society Analog Signal Processing Technical Committee. He was the IEEE Circuits and Systems Society Distinguished Lecturer, from 2016 to 2017. He was a recipient of the Best Paper Award from the IEEE ISCAS 2011 International Symposium and the IEEE BioDevices 2008 Conference. In 2016, he received the 2015 ONE Award from the President of the Republic of Cyprus for his research accomplishments. He is the Chair of the IEEE Biomedical and Life Science Circuits and Systems (BioCAS) Technical Committee. He served as the General Chair for the 2010 IEEE Biomedical Circuits and Systems Conference and the Action Chair for the EU COST Action ICT-1401 on "Memristors-Devices, Models, Circuits, Systems and Applications (MemoCIS)." He was an Associate Editor of the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS and the *Frontiers in Neuromorphic Engineering* journal.