Modem Design in the Era of 5G and Beyond: The Need for a Formal Approach

Robert Wittig, Andrés Goens, Christian Menard, Emil Matus, Gerhard P. Fettweis and Jeronimo Castrillon
Technische Universität Dresden
Email: {first.lastname}@tu-dresden.de

***Abstract*—In the era of 5G and beyond, adaptive workloads and the need for energy efficiency drive are becoming increasingly vital. Changes in parameters of the physical layer algorithm can cascade throughout the algorithm, requiring additional changes to keep a correct functionality within the timing bounds. These factors drive the process of designing systems for mobile communication towards reconfigurability. In this paper we analyze the trade-offs involved in changing algorithmic parameters and show how reconfigurable systems can be used to produce energy-efficient systems. We argue that we ought to resort to formal models to tame this reconfigurability and examine where existing formal models fall short.**

***Index Terms*—5G, modem design, base station, formal model**

I. Introduction

Designing custom systems for mobile communications is an elaborate process. It involves coordinating the communication between various subsystem elements, e.g. accelerators and control units, as well as the tracking of dependencies, while obeying requirements for latency and throughput. Even small changes in an algorithm often entail new dependencies, which in turn require profound adaptation of large parts of the system. This also makes exploiting data parallelism at the hardware level particularly difficult. Only a handful of companies can thus afford to offer highly integrated solutions that cover a complete standard, let alone multiple standards.

This design complexity will increase considerably in the near future. Every new iteration of a standard pushes the key performance indicators (KPIs) further. The 5G standard mandates a throughput beyond 1 Gbps to be reached with only a few milliseconds latency [1], see Fig. 1, and these requirements will tighten even further in upcoming standards. While this affects the peak performance, there is also a similar trend to expand the dynamic range of the workload as well. A voice call might only need a few kilo bytes per second with low latency, whereas a high-quality streaming service will require the full bandwidth, resulting in a dynamic range of up to seven orders of magnitude. Adding to the complexity of modem design, the increased parameter space makes it challenging to achieve high energy efficiency for every use case. Thus, we face two intertwined problems: Driven by the increased parameter space, modem design is getting more complex for the worst-case scenario, while the dynamic range of requirements makes it harder to be efficient as well.

Contemporary work harnesses the complexity with software defined radio (SDR) solutions [2], porting the functionality of a base station to general-purpose hardware [3]. Especially cloud radio access networks (C-RANs) enable the sharing of base station hardware through virtualization, increasing utilization [4]. Of course, these approaches cannot achieve the same energy-efficiency as custom hardware, expanding the energy footprint. Hence, we propose to complement SDR approaches with the use of runtime-reconfigurable hardware [5].

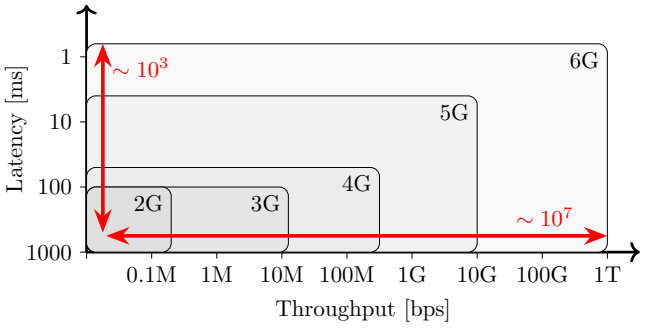

Fig. 1. Key performance indicators of mobile communication standards.

Originally, reconfigurable hardware in base stations enabled performances close to custom hardware, while allowing for rapid adaptation to new standards and algorithms. As technology progresses, field-programmable gate arrays (FPGA) can now be used for runtime system adaptation. In this paper, we argue that these means can be used to solve the energy efficiency challenge. We propose to exploit the dynamics between consecutive transmission time intervals (TTIs) for system adaptivity, according to the parameters dictated by the MAC layer, e.g., user count and resource block scheduling. By doing so, we eliminate the need for one flexible hardware configuration and instead use several highly tuned, smaller configurations. In consequence, energy efficiency can be raised in periods with low workload while preserving the ability to handle worst-case scenarios. Furthermore, our method can exploit even larger parameter spaces, accommodating scenarios that were not possible before.

Runtime reconfiguration adds another layer of complexity to the modem design process. Several formal Models of Computation (MoC) have been proposed to tame the complexity of similar design processes. To overcome the complexity of 5G and beyond, the system has to reconfigure and adapt to dynamic workloads, controlling algorithmic changes for different KPIs, having tight control over latency and exploiting data parallelism, even at the hardware level. Well-defined formal models

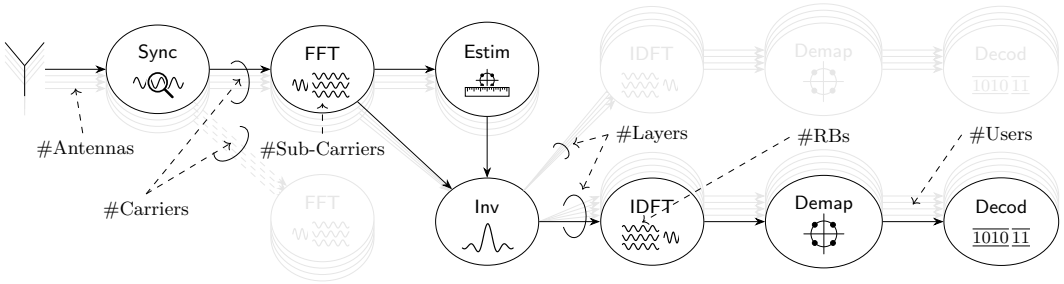

Fig. 2. Simplified model of a base station uplink modem with a set of parameters that can be adapted during runtime.

are essential for this. Widespread and established models, stemming from an era of less dynamic modems, are insufficient for this task. We argue that we require a formal approach that is inherently timed, deterministic, and especially, has well-defined semantics for adaptivity. To this end, we analyze how these properties allow us to better handle the emerging complexity and tackle the gap between the emerging reconfigurability of hardware and the error-prone modem design.

II. Physical Layer Design

The increasing parameter space offers both a challenge with regard to complexity and a chance for more energy-efficient designs. In this section we focus on the former, outlining the common practices in current modem design as well as the challenges we expect from emerging standards and technologies. The means to embrace the latter challenge of energy-efficient designs are subject to the subsequent sections.

The development of mobile communication systems is driven by trade-offs between different KPIs like data rate, latency, and user count. Standards define a range in which these KPIs can be adapted in the field. For example, a single user experiencing a high data rate may have their resources reduced when additional users join the system. The space of potential trade-offs is constrained by the algorithms used, the processing technologies, or both. With Moore's law intact, it becomes possible to increase the efficiency of an algorithm, e.g. used bandwidth, or to implement more sophisticated algorithms that improve performance. Additionally, the physical layer has hard real-time constraints [6], hence, it is usually built with custom-made hardware and FPGA solutions instead of general-purpose CPUs [7], [8]. Attempts to shift base station processing to the cloud [9] still exhibit too high communication overhead in the back-end.

Designing custom hardware is challenging, especially in the presence of real-time requirements. This is why almost no literature is available on complete base band design, except for software benchmarks [10], [11] and few simulation platforms [12], [13]. Nevertheless, the design process usually follows a top-down approach. Starting with high-level tools like Matlab or Simulink, every step of the processing chain (see Fig. 2) is implemented and tested. After validation, the individual processing steps are mapped to accelerators, and the latency and throughput requirements are validated. In this step, crucial information about the underlying algorithms is lost. As an example, consider the FFT block in Fig. 2. It is possible to implement it as a single, controllable IP block that meets the required timing in the worst case. Assuming the 5G standard and a bandwidth of $20\,\mathrm{MHz}$, that means 2048 samples must be processed every $66.7\,\mu\mathrm{s}$. However, it is also possible to split the computation into multiple slower FFT blocks. Each block can then process 2048 samples independently and in parallel, which allows for a linear increase in throughput that cannot be achieved by a single block. The splitting entails that the next processing step has to be adjusted as well, to get the input from different FFT blocks. The problem worsens in modern embedded systems, where the various accelerators are connected with a network-on-chip that introduces a non-deterministic communication overhead. Data can arrive unordered from different FFTs which can in turn lead to non-deterministic behavior. Opting for the single IP solution avoids these problems, but also surrenders the potential benefits of parallel processing. Alternatively, a deterministic formal approach could keep this information and avoid this problem. However, the model should also encompass information about the precise timings to keep the real-time constraints.

To increase the complexity further, a change in one algorithm might entail a profound adaptation of the whole system. Consider the transition from a cyclic prefix channel estimation to a pilot-based algorithm. Instead of using every symbol to sound the channel condition, only pre-defined symbols have to be fed into the estimator. This means that new synchronization signals between the accelerators are needed. The change might also impact subsequent blocks as estimations are only available upon receiving the pilots, which might also affect the system's total real-time capability. Suitable formal methods should thus allow us to reason about these algorithmic changes, and how they affect different timings within the algorithm.

Another source of complexity is the increased parameter space. In addition to parameters like user count, used MIMO scheme and carrier aggregation, the sub-carrier spacing is also flexible in 5G systems. As a direct consequence, the real-time requirements have to adapt to the changing transmission time interval. Increasing the parameter space directly correlates to the dynamic range of possible workloads. A modem is required to run efficiently at full capacity, while it should also save energy in periods with a low workload. With the advent of 5G, workloads range at the order of seven orders of magnitude [14], and are expected to increase in future standards.

III. Leveraging Adaptivity in 5G and Beyond

With runtime reconfiguration we can utilize the increased parameter space of 5G. While researchers have focused on

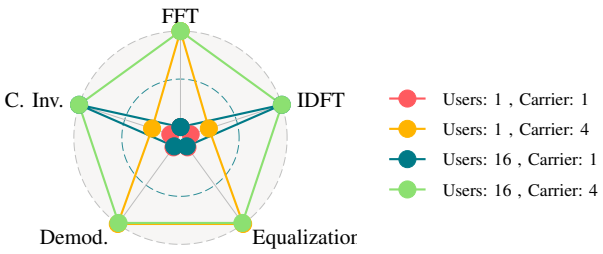

Fig. 3. Resource trade-off for different parameter configurations.

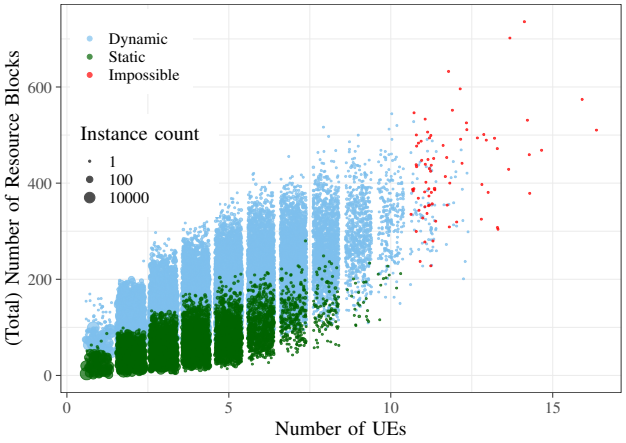

Fig. 4. Possible configurations in a resource-constrained LTE environment. The number of UEs are depicted with a meaningless random jitter for visibility.

reconfiguration in the 5G front-haul [15], we strive to replace the monolithic design of baseband modems in favor of multiple smaller, highly tuned designs. This offers an opportunity for considerable energy savings. Ever further, reconfiguration allows us to expand the operating space of a base station.

A base station typically has a fixed amount of FPGA hardware used to implement custom accelerators (e.g., for FFT, channel estimation and demodulation). For a given modem design it is then crucial to utilize the hardware efficiently to enable processing for a wide range of workloads. In consequence, designers must find the optimal count of every accelerator to deploy in the design. That is, we assume a fixed set of parameters and design a static system, capable of operating solely within the bounds of this worst-case parameter space. Thus, finding the optimal parameters requires a careful trade-off to define a large parameter space with the limited resources offered by the platform.

Different modem parameters have opposing processing requirements as can be seen in Fig. 3. The data stems from our C model of a base station uplink modem where we counted the basic mathematically operations in each processing step. It was originally designed for cloud RAN deployment and supports fine grained data parallelism. Shown are two constellations that differ only in user count and the number of aggregated carriers. The latter requires higher processing for FFT, demodulation, and equalization. In contrast, a high user count puts more load on the inverse discrete Fourier transform (IDFT) and the channel inversion. To support both constellations the maximum count for each accelerator has to be chosen. This poses two problems. First, if there is not enough hardware to support the combination of both constellations, it is necessary to scale down thereby reducing the supported parameter space. Second, the parts of the design that are not utilized by a constellation might still drain more energy than required.

To relieve these limitations, we propose to use runtime reconfigurations in base stations. To illustrate the impact on the design space, we used LTE traces to extrapolate information about the dynamicity of the demands for 5G and beyond. These traffic traces, collected over a 5 hour period spread over 15 days, feature real data with over 1.2 million Radio Network Temporary Identifiers (RNTIs) from 24 different base stations [16].

Figure 4 shows statistical data points extracted from the traces. Every point represents the workload of a base station at a particular set of subframes. We consider the relationship between the number of User Equipment units (UEs) and the total number of resource blocks required at those subframes. The size of each point represents how many subframes in the trace had those precise requirements in terms of UEs and total resource blocks.

This is a multi-dimensional design space, if we take into account the other parameters from Fig. 2, like the number of antennas, layers, or subcarriers. Still, even the two-dimensional figure illustrates the impact onto the design space. In a traditional setup, with a static implementation of the PHY layer, we would need to make worst-case assumptions about the resource requirements. By generating task-graphs for each configuration, we can estimate these requirements.

Let us assume that, given our limited resources, a static PHY layer implementation can support up to 10 UEs and 47 resource blocks per UE. These numbers are admittedly low, even for LTE. However, we choose this low threshold deliberately to elaborate on the benefit of dynamic reconfiguration. Out of the 3017424 considered subframes in the traces, slightly over half (1689447) could be processed with this static system. These design points are classified as “Static” and depicted as green points in Figure 4. Now, consider a dynamic system that can adapt to the current workload, e.g. by supporting fewer UEs but more resource blocks per UE, or vice-versa. We classify the design points that can only be supported by this dynamic system as “Dynamic”. Slightly less than half of the subframes in the trace (1327900) fall under this category. Finally, the remaining 77 points would need more resources to be supported, even with a dynamic system. We classify them as “Impossible” (red points). Overall, over 99.997% of the subframes observed could be implemented by a dynamic system, whereas a static one using roughly the same resources covers less than 56%.

TABLE I
SCALING FFT RESOURCES FOR DIFFERENT LENGTHS.

Transform Length	Area [LUTs / Registers]	Power [mW]
8192 (variable)	3573 / 5685	242
8192 (fix)	2915 / 4789	230
64 (fix)	1135 / 1811	60

Reconfiguration can also lead to energy savings over periods with smaller workloads. If only a low-traffic user is served by a base station, it is possible to scale down the design in the parameter to provide just enough resources. For example, instead of implementing an FFT accelerator that can handle different transform lengths, reconfiguration allows us to implement only the required length for the current transmission time interval. Table I illustrates how much resources can be saved. It shows implementation results when scaling the FFT for different lengths for the Xilinx FFT accelerator on a ZC706 evaluation board and 250 MHz

IV. FORMAL MODELS

The design of a runtime reconfigurable modem can both improve performance and energy efficiency. We have also discussed, however, how tight real-time constraints and algorithmic complexity already make modem design extremely difficult, even without reconfigurability. If we are to include dynamic reconfigurability in modems, we need a solid formal model to manage this complexity. In this section we consider the aspects a formal model well-suited for 5G and beyond should ideally have, and discuss existing models in terms of these. We explain why the different properties are desirable for modem design, how the advantages we have shown possible in this paper can be achieved, and discuss existing formal models that feature them.

a) Determinism: It can be argued that the success of modern hardware, in general, is in a large part due to its deterministic behavior. Deterministic execution semantics are extremely valuable in software design [17]. They allow for reproducible and testable behavior, which is essential for debugging and ensuring the modem behaves as intended. As explained in Section II, this is especially relevant with emerging Network-on-Chip technologies and to manage the cascading effects from algorithmic changes.

Determinism in the execution of software can be defined in different ways. Commonly it is understood to mean that the same input will produce the same output. However, thighter definitions of determinism can be taken to include a deterministic timing behavior: for multiple executions with the same input, a program will produce the same output and take the same amout of time. A large variety of formal models exist that ensure determinism. Functional programming, for example, is clearly deterministic by its very definition. Dataflow models [18], like Synchronous Data Flow (SDF) [19], Cyclo-Static Data Flow (CSDF) [20] and a multitude of related concepts provide determinism by modeling the system as a static network of *actors*, which perform the execution and have well-defined communication patterns. Mostly the dynamicity of these communication patterns distinguishes between different models. These actors can be seen as inspired by the actor model [21], which is not deterministic. Kahn Process Networks (KPNs) [22] are similar to these dataflow models. The communication in KPNs can be completely dynamic, but in a deterministic fashion from the input. Similarly, discrete event models, well-known in hardware design, encapsulate execution in actors and ensure determinism by having discrete time events at which the actors fire. Many iterations of these formal models were created to design software modems. In terms of determinism, all the models we have mentioned are appropriate for modem design and would provide benefits over ad-hoc approaches.

b) Timing Semantics: Modem design is a real-time problem, since processing has to comply to the hybrid automatic repeat request. Hence, a single frame has to be processed in three milli-seconds or less in 5G. Having timing be part of the semantics of the underlying formal model is thus essential for a formalized modem design. It allows reasoning about time and helps deal with issues where a change in a part of the algorithm spreads throughout the whole design.

Synchronous languages, like Lustre [23] and SIGNAL [24] have a well-defined notion of time. In these languages, the model of time is purely logical. In designing a modem, where timing semantics are crucial to coordinate the tight real-time demands of the application, the time model should have a direct connection to physical time. This is true of hardware description languages (VHDL, Verilog), which can be considered as implementations of the discrete events model. Even more so SystemC [25] or SpecC [26], where the semantics are more closely related. However, these models have been designed to specify hardware that is static in its features and the underlying model is based on a static networks of actors. More recently, the PTIDES model [27] and the related Reactors [28] establish these connections in an elegant fashion.

c) Adaptivity: We argued in favor of runtime system reconfiguration (cf. Section III). This should be supported by the underlying formal MoC with a well-defined model of adaptivity. To use different versions of the algorithms (cf. Fig. 4), the system has to have a well-defined method for adapting to the required workload without breaking its semantics. Models like SDF, CSDF and even KPN all rely on static networks. There are, however, extensions to these models with well-defined adaptivity semantics [29], [30].

In an unstructured, informal approach reconfiguration can be easily implemented in an ad-hoc fashion. However, the pitfal of ad-hoc reconfigurability is that it comes at a very high price: at the cost of deterimism and testability.

A. Formal models for 5G and beyond

We argue that a formal approach to wireless modem design for upcoming standards should have the above three properties. An exhaustive list of formal models of computation is far beyond the scope of this paper. A more in-depth classifications can be found in [31]. While many versions of the aforementioned formal models were created to design such modems, most

of them do not feature all these properties. AdaPNet [30] is a deterministic model with well-defined adaptivity semantics. While it lacks timing semantics, enhancing it with them would make a suitable model for modem design in the future. Very recently, a formal description [32] of the Reactors [28] model proposes a definition including transformations, which endows this model with the required adaptability. We believe this makes Reactors the best candidate among existing models for 5G and beyond.

V. Conclusions and Outlook

In this paper we outlined the challenges involved in current modem design processes. The need for energy efficiency and increasing parameter space in 5G and beyond both expand the complexity for manufacturers. We showed in our analysis how reconfiguration can be leveraged to mitigate the energy efficiency problem. To handle the increasing design complexity we should resort to a formal model that is deterministic, inherently timed and adaptable. This paper focused on the advantages of reconfiguration and the need for formal models in 5G and beyond. On future work we plan a concrete implementation using Reactors.

Acknowledgments

This work was funded in part by Dr. James Truchard. Additionally we would like to express our gratitude towards Jacob Kornerup, Aniruddha Shastri, and National Instruments for their time and ongoing support. In addition, this work was funded in part by the German Research Council (DFG) through the TraceSymm project CA 1602/4-1, the German Federal Ministry of Education and Research (BMBF) through the KMU-innovativ project EM-RAM (01IS17016D), and the Studienstiftung des deutschen Volkes.

References

[1] P. Schulz *et al.*, "Latency Critical IoT Applications in 5G: Perspective on the Design of Radio Interface and Network Architecture," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, feb 2017.

[2] C. Liang and F. R. Yu, "Wireless Network Virtualization: A Survey, Some Research Issues and Challenges," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1, 2015.

[3] N. Nikaein *et al.*, "Towards a Cloud-Native Radio Access Network," in *Advances in Mobile Cloud Computing and Big Data in the 5G Era*, 22nd ed., 2017.

[4] H. Wen, P. K. Tiwary, and T. Le-Ngoc, "Current trends and perspectives in wireless virtualization," *International Conference on Selected Topics in Mobile and Wireless Networking, MoWNeT*, 2013.

[5] China Mobile, "C-RAN: The Road Towards Green RAN," *China Mobile White Paper, ver 2.5*, vol. 5, pp. 15–16, 2011.

[6] M. Sauter, *From GSM to LTE-Advanced Pro and 5G*, 2017.

[7] Xilinx, "LTE base band targeted design platform product sheet," Tech. Rep., 2019.

[8] Nokia, "AirScale base station product sheet," Tech. Rep., 2019.

[9] A. Checko *et al.*, "Cloud RAN for Mobile Networks—A Technology Overview," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 405–426, 2015.

[10] M. Själander *et al.*, "An LTE Uplink Receiver PHY benchmark and subframe-based power management," in *2012 IEEE International Symposium on Performance Analysis of Systems Software*, apr 2012, pp. 25–34.

[11] Q. Zheng *et al.*, "WiBench: An open source kernel suite for benchmarking wireless systems," in *2013 IEEE International Symposium on Workload Characterization (IISWC)*, 2013, pp. 123–132.

[12] V. Venkataramani *et al.*, "SPECTRUM: A Software Defined Predictable Many-core Architecture for LTE Baseband Processing," in *Proceedings of the 20th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems*, ser. LCTES 2019. New York, NY, USA: ACM, 2019, pp. 82–96.

[13] S. Haas *et al.*, "A Heterogeneous SDR MPSoC in 28 nm CMOS for Low-Latency Wireless Applications," in *DAC*. New York, New York, USA: ACM Press, 2017.

[14] G. P. Fettweis and E. Matús, "Scalable 5G MPSoC architecture," *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 613–618, 2017.

[15] D. Chitimalla *et al.*, "Reconfigurable and efficient fronthaul of 5G systems," in *2015 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS)*, dec 2015, pp. 1–5.

[16] N. Budhdev, M. C. Chan, and T. Mitra, "Isoran: Isolation and scaling for 5g ranvia user-level data plane virtualization," 2020.

[17] E. A. Lee, "The problem with threads," *Computer*, vol. 39, no. 5, pp. 33–42, 2006.

[18] J. B. Dennis, "First version data flow procedure language," MIT Laboratory for Computer Science, Report MAC TM61, 1974.

[19] E. A. Lee and D. G. Messerschmitt, "Synchronous data flow," *Proceedings of the IEEE*, vol. 75, no. 9, pp. 1235–1245, 1987.

[20] G. Bilsen *et al.*, "Static scheduling of multi-rate and cyclo-static DSP applications," in *Workshop on VLSI Signal Processing*. IEEE Press, 1994, Conference Proceedings.

[21] C. Hewitt, P. B. Bishop, and R. Steiger, "A universal modular ACTOR formalism for artificial intelligence," in *Proceedings of the 3rd International Joint Conference on Artificial Intelligence. Standford, CA, USA, August 20-23, 1973*, 1973, pp. 235–245.

[22] G. Kahn, "The semantics of a simple language for parallel programming," in *Proc. of the IFIP Congress 74*. North-Holland Publishing Co., 1974, Conference Proceedings, pp. 471–475.

[23] N. Halbwachs *et al.*, "The synchronous data flow programming language LUSTRE," *Proc. of the IEEE*, vol. 79, no. 9, pp. 1305–1319, 1991.

[24] A. Benveniste and P. Le Guernic, "Hybrid dynamical systems theory and the SIGNAL language," *IEEE Tr. on Automatic Control*, vol. 35, no. 5, pp. 525–546, 1990.

[25] S. Liao, S. Tjiang, and R. Gupta, "An efficient implementation of reactivity for modeling hardware in the Scenic design environment," in *Design Automation Conference*. ACM, 1997, Conference Proceedings.

[26] D. Gajski, *SpecC: Specification Language and Methodology*. Norwell, MA: Kluwer Academic Publishers, 2000.

[27] Y. Zhao, J. Liu, and E. A. Lee, "A programming model for time-synchronized distributed real-time systems," in *RTAS '07*, April 2007, pp. 259 – 268.

[28] M. Lohstroh *et al.*, "Actors revisited for time-critical systems," in *Proceedings of the 56th annual Design Automation Conference*, ser. DAC 2019. Las Vegas, NV, USA: ACM, Jun. 2019, p. 4pp.

[29] R. Khasanov, A. Goens, and J. Castrillon, "Implicit data-parallelism in kahn process networks: Bridging the macqueen gap," ser. PARMA-DITAM '18. New York, NY, USA: ACM, Jan. 2018, pp. 20–25.

[30] L. Schor *et al.*, "Adapnet: Adapting process networks in response to resource variations," ser. CASES'14. ACM, 2014, p. 22.

[31] C. Ptolemaeus, *System Design, Modeling, and Simulation Using Ptolemy II*. Berkeley, CA, USA: Ptolemy.org, 2012.

[32] M. Lohstroh *et al.*, "Reactors: A deterministic model for composable reactive systems," in *Proceedings of the 9th Workshop on Design, Modeling and Evaluation of Cyber Physical Systems (CyPhy 2019) and the Workshop on Embedded and Cyber-Physical Systems Education (WESE 2019)*, Oct. 2019, p. 26pp.