Custom versus Cell-Based ASIC Design for Many-Channel Correlators

Custom versus Cell-Based ASIC Design for Many-Channel Correlators

Erik Ryman[†‡], Christoffer Fougstedt[†], Lars Svensson[†], and Per Larsson-Edefors[†]
[†]Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden
[‡]Omnisys Instruments AB, Västra Frölunda, Sweden

***Abstract*—While ASICs are efficient in terms of area utilization, performance, and power dissipation, ASIC design requires significant development resources. We compare two approaches to implementing ASIC correlators for interferometric imagers and spectrometers: The first approach, custom design, gives very high performance and area utilization, but is complex and time consuming. The second approach, cell-based design, reduces design time, but leads to lower performance and area utilization. In our evaluation, we consider two different correlator architectures: Autocorrelators for spectrometry, and cross-correlators for synthetic aperture imaging. Based on both 65-nm CMOS and 28-nm FD-SOI process technologies, our results show that for implementations for a limited number of channels, the cell-based approach may prove useful since it offers relatively short development time while still providing acceptable area utilization and performance. For larger designs, however, the area overhead of cell-based design becomes a major concern, especially for autocorrelator architectures.**

I. Introduction

Signal processing is a vital part of interferometric spectrometry and imaging systems. To make these applications truly useful it is, however, imperative to design signal processing circuits that are able to handle an increasing channel count at very high speed. In addition, while also important for ground-based observations, it is critical to consider power dissipation constraints when dealing with space-borne systems.

In order to maximize area utilization, performance, and power efficiency, signal processing systems are often implemented as application-specific integrated circuits (ASICs). Field-programmable gate arrays (FPGAs) are competitive from the point of view of their rapid development flow, but they cannot compete with ASICs as far as area and power efficiency and, ultimately, the capacity to handle many channels. Digital ASICs can be designed using two different approaches; custom or cell-based design. While the former, thanks to its flexibility in crafting layouts at the transistor level, leads to maximal performance and area efficiency, it requires extensive development resources. In contrast, cell-based design, which relies on predesigned logic gates and therefore has less degrees of design freedom, offers shorter development time. The downside of cell-based designs, however, is less efficient usage of chip area, lower performance and higher power dissipation [1], [2], which impacts, e.g., the number of signal processing elements that can be implemented on one chip.

In this paper, we will consider two important applications in the fields of interferometric spectrometry and imaging, and how they can be effectively implemented in ASIC-based signal processing systems. In essence, we wish to answer the following question: How do we best design and implement future autocorrelator and cross-correlator ASICs given an increasing need for channel capacity?

II. ASIC Design Approaches

Digital ASIC design involves the creation of high-density layouts of transistors and wires. Today, digital ASIC often is synonymous with cell-based design approaches, where the ASIC designer is supported by a library of standard cells pre-designed at an IC foundry. Here, ASIC design entails developing register-transfer level (RTL) code using hardware description languages such as VHDL or Verilog, synthesizing this RTL code to gate netlists that are made up by a selection of the provided standard cells, and finally performing physical design where the cells are placed and routed. Since this design approach to a large degree is supported by design software tools, cell-based ASIC implementation is a rational approach which saves time and reduces risks over custom design.

While the custom approach is the main paradigm for analog design, it is also used when area utilization, performance and power efficiency are critical to digital ASICs. In a custom design approach, the designer develops logic circuits at the transistor level; first as a transistor schematic, then as a layout. The booming complexity of digital ASICs has, however, largely prohibited custom approaches; unless some kind of circuit hierarchy can be enforced, the design time increases exponentially with complexity. The correlator architectures that we consider here are highly regular, and more importantly, as we increase the number of channels on the chip, there is a possibility to maintain the regularity. This can be used in a custom-design approach, where logic cells and wire routes are customized for the application and where the intrinsic correlator hierarchy allows for cells (including wiring) to be instantiated for different channel counts[1].

So while correlator architectures may be well suited to custom design approaches, there is still this issue of design time. The bottom-up approach of custom design puts a serious strain on ASIC development resources and this begs the question if cell-based approaches can offer an intermediate solution that gives high enough "electrical" efficiency while reducing development costs.

[1]While this resembles the concept of memory compilers, correlator architectures do not display as regular organization as, e.g., SRAM arrays.

III. Correlator ASIC Architectures

Cross-correlation is an operation performed between two signals, f and g. The signals are multiplied and summed, with a different time delay, n, applied to one of them, according to

$$(f \star g)[n] \overset{\text{def}}{=} \sum_{m=-\infty}^{\infty} f^*[m]\, g[m+n] \qquad (1)$$

where f^* denotes the complex conjugate of f. While autocorrelation of a signal is the cross-correlation with itself ($f = g$), the implementations have significant differences at an architectural level, which is due to the different usages of the correlator operation. Autocorrelation is used for spectrometers in the field of remote sensing [3]. Here a high spectral resolution is of interest and this is achieved by implementing many delay steps, or lags, in the correlation function.

Cross-correlation is extensively used for signal processing in aperture-synthesis-based radio astronomy and is currently being considered also for remote sensing applications [4], [5] as well as security scanning applications [6], [7]. In these applications, the main driver for processing power is imaging resolution. Instead of implementing many lags, the cross-correlation is kept simple, usually without lags altogether ($n = 0$). However, the cross-correlation has to be calculated for a very large number of different signal pairs or baselines.

Previously we have reported on a custom cross-correlator ASIC [8], but we have now also implemented an autocorrelator, based on experience from both the cross-correlator and earlier autocorrelation spectrometer development [9]. The correlators investigated share a number of features such as buffered readout, serial interface, dynamic multipliers, and similar integrators. In this section we will first describe the two ASIC architectures and then delve deeper into differences and similarities at circuit level.

A. Cross-Correlator

Cross-correlation, as used in aperture synthesis, has to be performed pair-wise for a large number of baselines from an array of receivers. Thus, the routing of a large number of signals in a cross-coupled network is a major challenge for these designs. The cross-correlator previously reported [8] uses a routing scheme shown in Fig. 1. Here, the correlator clock and data are routed together; one clock for each input channel. The channels are split into two separate paths; one going straight and one diagonally. At each intersection, a cross-correlator block including a synchronizer is placed. The two clocks are synchronized using a C-element, while the data is synchronized with this combined clock. The fabricated ASIC has 96 single-bit inputs and can operate as either a 96-channel 2-level or as a 48-channel 3-level cross-correlator.

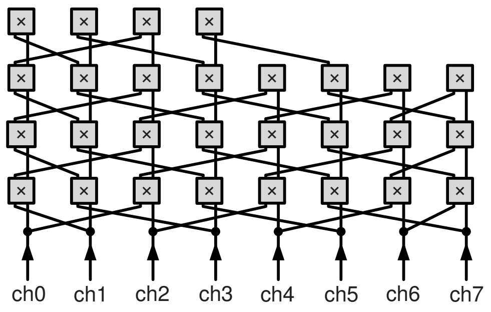


Fig. 1. Routing architecture of a 8-channel full-custom cross-correlator.

B. Autocorrelator

The autocorrelator implementations investigated in this paper all implement a time-division factor of four (TDM4). This means that the sampler operates at a sampling frequency four times the correlator clock and that each signal sampled is presented on four parallel inputs. The reason for employing this time division is to increase the available bandwidth. Each input is also represented as in-phase I and quadrature Q, with 3-level (2-bit) precision. Thus, the total number of 1-bit data inputs is 16. The inputs are then split into two paths where one is progressively time-delayed. For each lag, see Fig. 2, the $I \times I_\Delta$, $I \times Q_\Delta$, $Q \times I_\Delta$, and $Q \times Q_\Delta$ products are calculated four times, i.e., one for each time division. The four time-divided multiplications are then merged and accumulated. Note that the width of the data routing lines is two bits, representing the 3-level samples. The data of the delay path are finally delayed by an additional sample clock cycle (one 4th of a correlator clock) by reordering the four signals, and by delaying one of them by one correlator clock cycle. Between each lag, synchronization of data and clock is also performed but this is not shown in the figure.

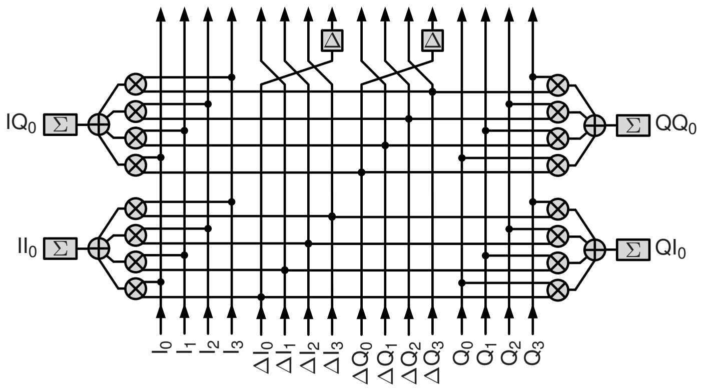


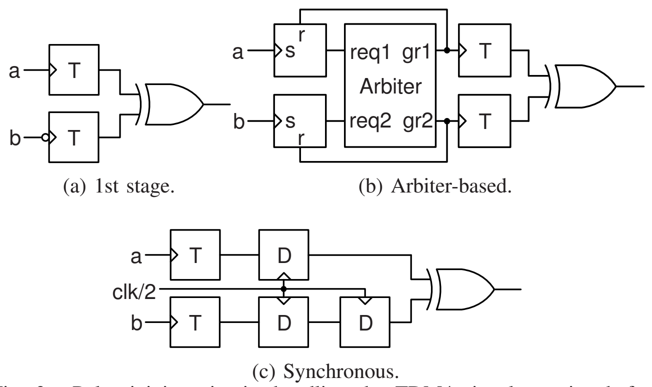
Fig. 2. A single lag of the TDM4 autocorrelator. Note that all paths are 2-bit/3-levels.

The autocorrelator may also be operated as a real-valued correlator, if needed, by replacing the quadrature phase sampling with an opposite phase sampling. By this method the number of "lags" in Fig. 2 is actually two, however, the number of resolved spectral channels and the bandwidth remain the same; two spectral channels are provided for each of these blocks. In Section VI we compare a range of autocorrelators based on the number of spectral channels they provide.

C. Full-Custom vs Cell-Based Circuits

For fair comparisons, we have, as far as the tools allow us, tried to keep features from the full-custom ASICs in the synthesized versions of the designs. All correlator implementations investigated have similar integrators, using chained toggle flip-flops, which are then buffered to a secondary storage for readout. This means the next integration can be performed simultaneously with the reading out of previous integration data. The readout is performed through a serial interface, using a secondary clock, operating at a lower frequency.

In the cell-based versions, the clocking of the input data routing is completely synchronous. Here, clock trees are synthesized for the correlator clock and readout clock regions. In the full-custom versions of both the cross-correlator and the autocorrelator, the correlator clock instead flows together with data along the datapaths, thus making the full-custom architectures row-wise synchronous and column-wise asynchronous.



(a) 1st stage. (b) Arbiter-based.

(c) Synchronous.

Fig. 3. Pulse joining circuits handling the TDM4 signal merging before integrators.

The dynamic multipliers used throughout all designs generate pulses of the same width as the correlator clock. In the autocorrelator, additional modifications had to be made in order to handle the data merging after multiplication, before integrators. The first merge is performed by using these pulses to clock a toggle flip-flop, or prescaler stage, as shown in Fig. 3a. By toggling half of the flop-flops on the positive and the other half on the negative edge, the two data signals can not change state at the same time, and so the first merge can be done by using a simple XOR gate. For the second merging the two remaining signals may have positive and negative edges simultaneously with each other, hence this stage cannot use the same simple circuit. In the full-custom case, a circuit employing an arbiter and SR-flip-flops is used, shown in Fig. 3b. Here, the pulse widths and delays through this circuit can be finely tuned. The synthesis tools used for cell-based design, however, were not able to deal with the asynchronous circuits correctly, hence, another approach was devised. Here, the correlator clock is halved in frequency and used to skew the signals before they enter the XOR gate, as in Fig. 3c. This means additional circuitry has to be driven by the correlator clock.

IV. Many-Channel Correlator Implementations

The cross-correlation products increase as $(n \cdot (n-1))/2$, where n is the number of inputs. This means a 96-channel cross-correlator has 4560 products. In addition to the correlation products, the fabricated 96-channel correlator [8] includes one monitor per input and one clock counter, but for simplicity reasons these monitors are not included in the cell-based counterpart. The monitors are each equal in size to a correlation product, thus, we consider the custom 96-channel correlator as a 4657-product correlator in the comparisons.

An autocorrelator design, featuring 8624 spectral channels, implemented in a 28-nm fully-depleted silicon-on-insulator (FD-SOI) process technology serves as the custom comparison case here. Similarly to the custom 96-channel cross-correlator, in addition to the spectral channels, the custom autocorrelator features two monitor blocks, not implemented in the cell-based counterparts of the autocorrelator. Each monitor block is equal in size to 8 spectral channels, thus, in the comparisons presented, we consider the custom autocorrelator as a 8640 channel unit.

Layouts of two autocorrelators are shown in Fig. 4. Even though each employs 8640 spectral channels, they are very different in terms of area usage and implementation structure. To make routing visible, only metal layers 3-5 are displayed, since most of the datapath routing is done on these layers. While the full-custom layout is routed in a regular pattern with ten straight folds, the synthesis tools make the cell-based implementation display a more "organic" appearance.

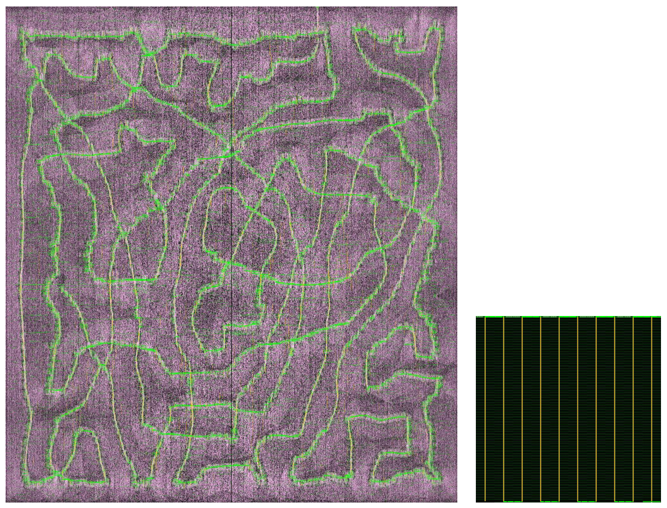

(a) Cell-based. (b) Full-custom.

Fig. 4. 8640-channel cell-based and full-custom 28-nm autocorrelator layouts (relative scaling shown).

The fixed channel count of a fabricated ASIC does not necessarily limit the channel count of a correlator system. Extending the number of channels, by utilizing multiple ASICs, does however effect system complexity. The impact, however, is different for autocorrelator and cross-correlator systems.

For an autocorrelator, there are basically two options for extending the frequency resolution: Parallelization or serialization, as illustrated in Fig. 5. Connecting autocorrelator ASICs in parallel means the frequency spectrum has to be divided into bands for each ASIC, using filter banks and power splitters. By instead connecting the ASICs in series, one can omit this extra circuitry, however, the ASIC will then require four times as many data pins, adding non-delayed signal outputs and delayed signal inputs and outputs.

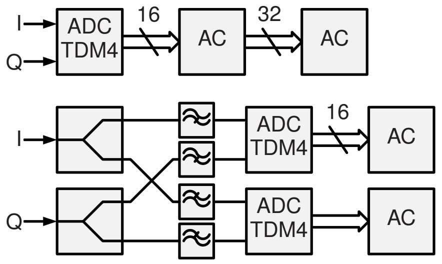


Fig. 5. System architectures for connecting multiple autocorrelators (AC) in series (top) or parallel using filter banks (bottom).

While the number of required autocorrelator ASICs grows linearly with required number of channels, the cross-correlator architecture is not as amenable to channel up-scaling. In fact, the required number of ASICs for the cross-correlator system, assuming all signal pairs are to be correlated, grows as $\lceil N/n \cdot 2\rceil \cdot \lceil N/n \cdot 2-1\rceil/2$, where N is the required number of channels for the system and n is the number of channels per ASIC, mirroring the arithmetic describing the number of correlation products within the ASIC. An example of such a scheme, building a 144-channel system out of three 96-channel ASICs, is shown in Fig. 6.

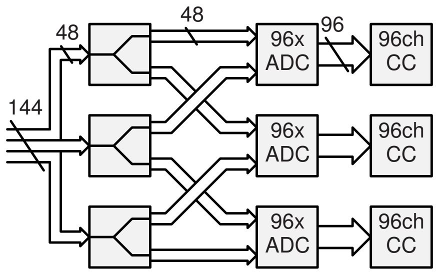


Fig. 6. System architectures for connecting multiple cross-correlators (CC).

With both these system architectures in mind, a clear case can be made for the advantage of incorporating a large number of channels in a single ASIC to minimize system complexity and power consumption.

V. ASIC Implementations

In our evaluations, we implement the correlators in both 65-nm CMOS and 28-nm FD-SOI process technologies. As previously reported [8], the 65-nm cross-correlator ASIC is fabricated using a mix of low threshold voltage (V_T) general purpose (LVTGP) transistors and high V_T low power (HVTLP) transistors. Input routing, clocks, multipliers, and first prescaler stages are implemented in LVTGP, while integrators and most readout logic use HVTLP. For the cell-based implementations, the same mix of transistor options were used in the 65-nm case, however, the tools heuristically handle which cells are used for what purpose, with the goal to minimize total area while meeting an overall timing constraint. The 28-nm FD-SOI process does not allow the same amount of flexibility for transistor selections, as these have to be separated into different regions. To avoid wasting chip area on region borders, the 28-nm implementations only use one transistor type throughout: Low threshold voltage (LVT) transistors.

The synthesis tools optimize the designs for meeting a 1-GHz correlator and a 100-MHz readout clock target across an operating range between slow and fast corners, including variation in temperature, supply voltage and device spread. For the full-custom design, simulations were performed using Monte-Carlo-based methods for device spread at typical temperature and supply voltage for target frequencies of up to 3 and 4 GHz for the cross-correlator and autocorrelator, respectively.

VI. Evaluation

In this study we compare the power consumption and chip area implications of the two ASIC design approaches, for the two different correlator types. While other metrics, such as possible top speed or ASIC development time, are also of interest, they were not within the scope of the performed study. Area comparisons will not include pad frame, but only active logic regions. The cell-based designs are synthesized using Cadence Genus, and place and route is performed by Cadence Innovus, which also extracts wiring RC parasitics. Power estimation for the cell-based designs is performed by simulating the circuits, using Cadence Incisive, and then performing power estimation, using Cadence Genus, on the RC-extracted netlists.

A. Chip Area Usage

The cross-correlator implementations differ significantly in terms of chip area as shown in Fig. 7. Remarkably, the 65-nm full-custom implementation is almost identical in area to the 28-nm cell-based version, with the full-custom design being even slightly more compact. For all different channel-count versions evaluated, the area used for the cell-based approaches lies at about three times the 65-nm full-custom case.

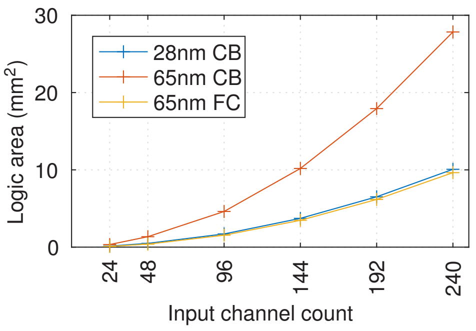


Fig. 7. Comparison of logic area between cell-based (CB) and full-custom (FC) cross-correlators.

For the autocorrelators, the area difference is even greater. Since the autocorrelator designs require much more logic than the cross-correlators, it does not make sense to consider these in a 65-nm process technology; hence, these are only evaluated in 28 nm. For the versions evaluated, the cell-based designs are about seven times larger than the full-custom designs as shown in Fig. 8.

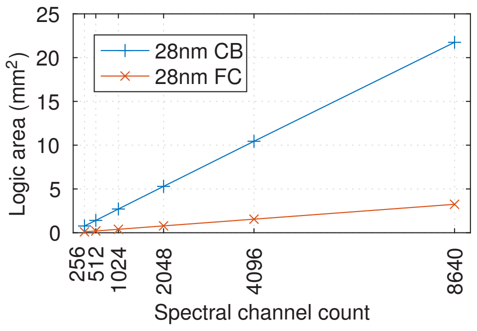


Fig. 8. Comparison of logic area between cell-based (CB) and full-custom (FC) autocorrelators.

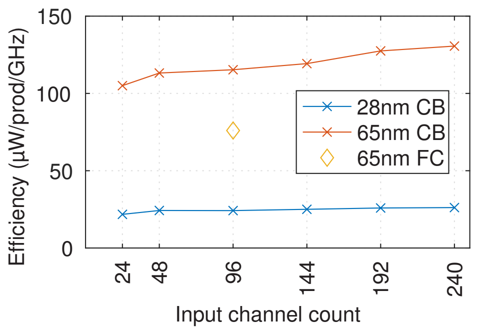


Fig. 9. Comparison of power efficiency between the cell-based (CB) and full-custom (FC) cross-correlators.

In summary, architectures with fewer across-chip interconnects (autocorrelator) are more suitable for full-custom implementation than those with a significant number of such interconnects (cross-correlator). A similar observation was made more than 15 years ago, using much less sophisticated ASIC design tools [10].

B. Power Dissipation

The cross-correlator implementations differ in power dissipation, as shown in Fig. 9. We use $\mu\mathrm{W/prod/GHz}$ as a power-efficiency metric to compare the implementations. The increased drive strength required to meet the timing requirement (1 GHz) for larger designs means a slightly increasing trend with increasing channel count can be observed. For the full-custom implementation, only the measured efficiency when operating at 1.5 GHz from [8] is displayed. Notably, the 65-nm full-custom power efficiency lies about halfway between 65- and 28-nm cell-based figures.

For the autocorrelator, a similar comparison of efficiency can be made, however, here the numbers are based only on simulations. For the full-custom implementation, the power dissipation of one RC-extracted lag using typical corner is simulated. A comparison of power dissipation per lag and GHz shows that while the full-custom lag dissipates only 40 μW at 1 GHz, the cell-based versions all come in between 470 and 480 μW, which is more than a magnitude higher.

One reason for the much wider gap between efficiencies of full-custom vs cell-based in the autocorrelator as compared to the cross-correlator implementations is the rather costly clocking scheme implemented in the full-custom cross-correlator. A large part of the power is dissipated by routing a clock path for every input signal throughout the ASIC and by the synchronizing C-elements used. In the autocorrelator case, there is much less power spent on clock distribution.

A major contributor to the difference in power efficiency for the autocorrelators is the merging of the four time divided signals after multiplication. Since the full-custom correlator handles this with asynchronous logic, it dissipates much less power for performing this function as opposed to the synchronous version implemented for the cell-based design, where the high-speed correlator clock has to drive considerably more logic. Also, the requirement of synchronicity cannot be relaxed until after this merging.

VII. Conclusion

In our evaluation of two different ASIC design approaches for two different correlator architectures we found that autocorrelators, whose logic interconnections to a large extent are local, should be developed using a full-custom approach since this significantly increases the number of channels one ASIC can support. In the context of cross-correlators, while full-custom approaches offer less of an area and power dissipation advantage over cell-based approaches, these also significantly benefit from full-custom design. Our conclusion is that the full-custom approach to implementing correlators will be important for meeting the future demand of an increasing number of spectral channels or higher baseline count.

References

[1] D. G. Chinnery and K. Keutzer, "Closing the power gap between ASIC and custom: an ASIC perspective," in *Design Automation Conf.*, Jun. 2005, pp. 275–280.

[2] A. Chang and W. J. Dally, "Explaining the gap between ASIC and custom power: a custom perspective," in *Design Automation Conf.*, Jun. 2005, pp. 281–284.

[3] A. Emrich, S. Andersson, and M. Krus, "Spectrometers for (sub)mm radiometers," in *Joint 31st Int. Conf. on Infrared Millimeter Waves and 14th Int. Conf. on Teraherz Electronics*, Sep. 2006, pp. 314–314.

[4] A. Carlström, J. Christensen, J. Embretsén, A. Emrich, and P. D. Maagt, "A geostationary atmospheric sounder for now-casting and short-range weather forecasting," in *IEEE Antennas and Propagation Society Int. Symp.*, Jun. 2009.

[5] B. Lambrigtsen, A. Tanner, T. Gaier, P. Kangaslahti, and S. Brown, "Developing a GeoSTAR science mission," in *IEEE Int. Geoscience and Remote Sensing Symp.*, Jul. 2007, pp. 5232–36.

[6] N. A. Salmon, J. Beale, J. Parkinson, S. Hayward, P. Hall, R. Macpherson, R. Lewis, and A. Harvey, "Digital beam-forming for passive millimetre wave security imaging," in *European Conf. on Antennas and Propagation*, Nov. 2007, pp. 1–11.

[7] C. Zheng, X. Yao, H. Anyong, and J. Miao, "Initial results of a passive millimeter-wave imager used for concealed weapon detection BHU-2D-U," *Progress in Electromagnetics Research C*, vol. 43, pp. 151–163, 2013.

[8] E. Ryman, A. Emrich, L. Svensson, and P. Larsson-Edefors, "A 3-GHz reconfigurable 2/3-level 96/48-channel cross-correlator for synthetic aperture radiometry," in *European Solid State Circuits Conf.*, Sep. 2017, pp. 39–42.

[9] A. Emrich, S. Andersson, J. Dahlberg, M. Hjorth, and T. Kjellberg, "HIFAS: Wide-band spectrometer ASIC," in *Fourth Edition of the Microelectronics Presentation Days*, Mar. 2010.

[10] H. Eriksson, P. Larsson-Edefors, T. Henriksson, and C. Svensson, "Full-custom vs. standard-cell design flow - an adder case study," in *Asia and South Pacific Design Automation Conf.*, Jan. 2003, pp. 507–510.