Low-Power Reconfigurable Computing for Biomedical Signal Processing

François Philipp¹ and Manfred Glesner¹.

ABSTRACT

This work introduces a low-power reconfigurable signal processing module for wearable body area networks. The architecture of the circuit is customized for the processing of sensor data used in wearable computing for biomedical applications such as EKG, EEG, or activity recognition. Hardware accelerators can be dynamically reconfigured to implement multiple signal processing tasks in a time-multiplexed manner. This approach allows reducing the size of the computing hardware while enabling energy-efficient operation. The functionality of the system is demonstrated for feature extractions in activity recognition applications with an average power consumption of 61µW.

Keywords: Biomedical signal processing, field programmable gate arrays, reconfigurable architectures

1. INTRODUCTION

1.1 Motivation

Surfing on the trend of personal informatics, the demand for low-cost wearable health monitoring systems is continuously increasing. These small electronic devices continuously measure diverse statistical characteristics of the patient condition (i.e. activity, heart rate, brain activity . . . ). Sensor nodes form a wireless body area network (WBAN) and regularly transfer their readings to a smart-phone, a personal computer or a remote server where the patient and his doctor can analyze the data and establish a diagnostic. The patient can thus react to these results and adapt his lifestyle to achieve healthier indicators. The large amount of recent projects initiated in this direction corroborates the increasing popularity of so-called eHealth systems. An overview of existing systems and evaluation features is given in [1].

In order to obtain the results of the sensor data analysis in real time, the system must be able to perform complex signal processing tasks online. If most of the existing solutions rely on the processing power of the smart-phone [1], we propose a distributed approach. Body-worn wireless sensor nodes have a very limited energy budget because of their small form factor which limits the battery size. As the recommended sampling rate of these sensors is relatively high (Table 1), wireless communication over protocol standards customized for such applications such as Bluetooth Smart or ANT has significant energy costs. A large amount of energy is wasted since sensor nodes must continuously stream data over the wireless link with a high rate. Furthermore, it exposes personal health data to the air, a vulnerable and unreliable communication medium. To enhance the security and the lifetime of the system, it is therefore preferable to minimize the utilization of wireless communication.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sampling Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrocardiography</td>
<td>100 – 500Hz</td>
</tr>
<tr>
<td>Electroencephalography</td>
<td>250 – 2000Hz</td>
</tr>
<tr>
<td>Electromyography</td>
<td>500 – 1000Hz</td>
</tr>
<tr>
<td>Pulse Oximetry</td>
<td>50 – 500Hz</td>
</tr>
<tr>
<td>Actigraphy with Inertial Sensor</td>
<td>20 – 100Hz</td>
</tr>
<tr>
<td>Respiratory Monitoring</td>
<td>100 – 200Hz</td>
</tr>
</tbody>
</table>

Table 1: Typical sampling frequency for selected biomedical monitoring methods used in WBANs

A simple solution to this problem would be to log the raw data into a non-volatile memory. The data is then transferred and processed offline when the device is connected by wire to the base station. This is for example applied by [2] for an activity recognition application. The authors of [3] showed that up to 58x energy can be saved by writing the data into a parallel NAND Flash memory instead of sending the data over the air. The authors showed that further reduction can be achieved by compressing the data or extracting features. However, the energy required by serial NOR Flash usually available on sensor nodes is up to two orders of magnitude higher, making logging also very expensive. In general, this approach is not suitable for applications where data has to be processed immediately to give real-time feedback to the user or generate alarms (for example the fall of an elderly person).

For both cases, reallocating the sensor data analysis on the wearable device and avoiding the transmission or logging of continuous data streams will improve the energy efficiency of the design. The main challenge of this approach is to provide a data processing infrastructure able to provide sufficient reli-
bility and performance while keeping the amount of necessary resources and the energy consumption low. For this purpose, we propose a low-power hardware solution based on dynamic reconfiguration. The frequent reuse of hardware resources to implement computationally intensive tasks will reduce the total size of the circuit while providing energy-efficient acceleration of sensor signal processing tasks. The availability of higher computational power will enable the implementation of algorithms with higher accuracy and reliability. The reconfigurability of the circuit makes it reusable for multiple applications or multiple sensor interfaces. The production costs are thus reduced since a single chip can be reused for wearable devices used in different contexts.

1.2 Related Work

The development of complete system-on-chip (SoC) solutions targeting biomedical signal processing was addressed in [4] and [5]. Chips integrating cores specialized for low-power wearable medical processing were introduced. Accelerators for multiple algorithms such as CORDIC, FFT, FIR filter and median filter were tightly coupled to a microprocessor core in [5] while the solution of [4] is limited to a specialized FFT core. In both cases, a reduction of two orders of magnitude in the energy consumption was observed when compared to an implementation on a CPU only. A similar approach is adopted for CoolBio [6], an ultra low-power chip targeting wearable heartbeat detection. The CoolBio is centered around the NXP CoolFlux baseband signal processor including a dual multiply-accumulate unit for accelerating biomedical signal processing tasks. The CoolFlux processor was compared with other state-of-the-art DSPs in [7] to estimate its suitability for biomedical monitoring, in particular EKG applications. The authors showed that Silicon Hive Pearl Ray very long instruction word (VLIW) processor achieves the best power-efficiency when running the EKG application. They also highlighted the importance of very low power consumption in sleep mode since the chip stays idle during most of the time.

Another reconfigurable processor for biomedical applications was proposed by Samsung in [8]. The architecture is based on an array of coarse-grained reconfigurable function units (CGRA) and a VLIW processor. The system supports several modes of operation to scale the performance and the energy consumption of the device dynamically. The authors illustrated the efficiency of their architecture with an EKG algorithm. The reconfigurable processor is however based on identical functional units. In this work, we propose an architecture with heterogeneous domain-specific functional units in order to accelerate application-specific arithmetic operations with higher efficiency.

The rest of this paper is organized as follows. Section 2 will describe the hardware architecture and the associated software in details. In section 3, we will take the example of activity recognition to illustrate the benefits of our approach and compare the performance with other state-of-the-art approaches. The last section will conclude the paper.

2. SYSTEM ARCHITECTURE

2.1 Reuse of Hardware Resources for Multiple Tasks

The flow chart depicted in Figure 1 depicts the different steps necessary to process data in an application for an eHealth device. This flow can be seamlessly applied for EKG, EEG, activity recognition or oximetry applications. Fig. 2 illustrates how we distribute these tasks between a reconfigurable hardware accelerator in charge of computationally intensive tasks and a RF Microcontroller Unit (MCU) for control-oriented tasks based on software.

The architecture proposed by Samsung in [8] implements all tasks using the same hardware resources. The processing engine switches in VLIW mode to handle control-oriented tasks while other tasks are run in the CGRA mode. We however consider that this solution has a very limited flexibility and that it is not suitable to implement complex programs such as the stack of a wireless communication protocol for example. A distinct separation can be drawn between the processing of the biomedical sensor data and the interaction with the rest of WBAN in terms of both data dependency and computational requirements. Therefore we physically separated these two parts in two different chips which are working in an independent and asynchronous manner. Data is exchanged only upon requests from the software running on the controlling unit. Two memories are used to store intermediate results during processing (volatile) and logs (non-volatile). This work focuses mainly on the optimization of the hardware accelerator for biomedical signal processing tasks.

Fig.1: Generic flow chart for an intelligent biomedical signal processing platform

The hardware accelerator is based on the CGRA described in [9]. The arithmetical operation or the
2.2 Customization of Processing Elements

In [5], a literature survey of biomedical signal processing algorithms showed that certain classes of arithmetical operations are dominating. For instance, FIR filters, FFT or template matching using sum of differences are typical operations for EKG QRS detection or heart sound processing.

In order to maximize the performance of the biomedical signal processing algorithms, we implemented the design with heterogeneous processing elements selected according to the profile realized in [5]. Each family of elements is customized for a specific type of operations which have a high occurrence in these algorithms. The architecture of the CGRA is depicted in Fig. 3 with three customized types of processing elements: a multiply-accumulate array, a CORDIC unit and a tree of arithmetical logical units (ALU). The multiply-accumulate unit will handle all algorithms based on this elementary operation such as a filtering, FFT or variance computation while the CORDIC can be configured to implement trigonometric and special operations such as square root, logarithm or division. The ALU tree is dedicated to other simple operators such as comparison or bit manipulation. A similar profiling has been realized in [10] for cryptographic and error correction algorithms. If required by the biomedical application, the coarse-grained reconfigurable elements specialized for these algorithms can be included in the architecture to extend the capability of the hardware accelerator.

At the top of the architecture, a memory layer handles the fetching of the data and configware for other layers. In the kernel, the data is processed by the processing elements configured by a generic controller. At the bottom, the processor interface reads the results and controls the global configuration of the core. The switch between feature extraction algorithms is handled automatically by the internal configuration controller.

The processing elements are included in the generic architecture with our graphical configuration software named (GECO) [11]. The output is a customized VHDL project which can be synthesized for the target technology (FPGA or digital application-specific integrated circuit (ASIC)). In a second step, the flow of the targeted biomedical signal processing application is mapped on the processing elements and dynamically loaded on the target platform where the configured architecture has been loaded. The different design steps leading to the execution of the application on a target platform are illustrated in Fig. 4.
3. CASE STUDY: ACTIVITY RECOGNITION FROM ACCELEROMETER DATA

3.1 Application description

Actigraphy embraces all methods for retracing the physical activities of a patient. The data can be used to estimate the amount of burned calories or periods of intense busyness in order to identify activity cycles (for example sleep cycles). The patient is wearing inertial sensors (usually an accelerometer coupled with a gyroscope) continuously measuring his movements. Specific features are extracted from the time series. Many works in the literature restrict the features set to mean and variance, mainly because of the limitations of the processing unit [2]. The utilization of additional features offers a larger space aiming to improve the quality of the classification. However, the computational complexity of the classification increases with the number of features. The accuracy will however start to decrease with a high number of features if the size of the training set is insufficient. In this case, the accuracy and the performance of the classification can be improved by selecting an optimal feature subset.

With our approach, we can dynamically modify the feature set according to context criteria such as location or time. Using this information, we can restrict the set of activity classes and reduce the search space. Each location or time is then associated to an optimal feature subset maximizing the accuracy of the classifier.

3.1.1 Feature Extraction

For our experiment, we selected an ADXL362 MEMS accelerometer from Analog Devices sampling at a frequency of 100 Hz. Accelerometer data is sampled in the range ±2g with a resolution of 12 bits on each axis. Previous to the feature extraction, spikes are removed by a short median filter and smoothed by a moving average filter. The gravity acceleration is then filtered by a high pass IIR filter of order 2 with a cutoff frequency of 0.3Hz. Features on each axis are computed using a sliding window. The window size as well as the overlap are dynamically configurable by setting a configuration register of the hardware accelerator.

The accelerometer integrates a 512-sample FIFO memory to limit the wake-up frequency of the sensor interface module. Power is thus saved by extending the sleep period of the hardware accelerator. Using this technique has a significant impact of the energy consumption impact at the system level. The authors of [8] showed on their system without sensor FIFO that the energy spent for collecting data is up to 25 times higher than the energy required to process it. Fig. 5 depicts the current consumption profile of the system.
hardware accelerator when the sensor FIFO is full. An interrupt will trigger the activation of the accelerator. After the restoration of the previous state, the data is transferred from the sensor. At maximum speed, this operation will take 1.6ms.

The data is then dispatched to the processing unit for feature extraction. Our processing core was programmed to compute the following features on each axis X, Y and Z of the accelerometer:
- **Time-domain features**: mean, min, max, range, variance, standard deviation, short time energy, zero-crossing rate, cross-correlation.
- **Frequency-domain features**: DC component, peak amplitude, peak frequency, total energy, spectral centroid.

In total, a vector of 42 features can be computed on each window. When unused, features are not computed to reduce the dynamic power consumption and the processing time.

### 3.1.2 Classification

A set of 16 gestures and activities associated with a location were selected for our test (e.g. tooth brushing and face washing are associated with the bathroom). As this example focuses on the feature extraction process, the accuracy and complexity of the classifier has not been taken into account. We only illustrate the usage of the selected features for the aforementioned application. Therefore, we used a simple binary decision tree to classify the readings in each location. The location awareness of the sensor is provided by the wireless interface. A localization service is implemented on top of the communication protocol and parametrizes the classifier accordingly.

Figure 6 shows the log for two activities related to two relevant features which allow to classify them easily. The accuracy of the classifier could reach 80% on our test data. Once the activity label has been determined, the information is passed to the CPU for communication or logging.

### 3.2 Task Mapping

The (GECO)$^2$ framework [11] is used to map the feature extraction algorithms on the architecture of the hardware accelerator (Fig. 3). Two additional MAC and ALU arrays can be added to process the data from each axis simultaneously but the unavailability of integrated multipliers in the target hardware costs a significant amount of resources. With three parallel MAC units, up to 75% of the cells of an IGLOO AGL1000 FPGA are occupied. This gives insufficient room for additional processing elements required for encryption or classification for example.

The CORDIC unit includes a range extension block inspired by [13]. The square root function required by the standard deviation is then implemented using the CORDIC with the hyperbolic class in vectoring mode. The twiddles factors for the FFT are computed by the CORDIC with the circular class rotating mode. The magnitude of the spectrum is computed by using the circular class in vectoring mode.

The ALU tree comprises seven hierarchically connected ALUs. In this application, the ALUs are mainly used for comparison operations. The MAC unit is the kernel for all tasks based on accumulation. The several internal multipliers can be reconfigured to implement the butterfly operation needed in the FFT algorithm or to perform one or several multiply-accumulate operations in parallel.

Fig. 7 illustrates how the feature extraction tasks are mapped on the function units for the features extractions on a single axis. The pre-processing filters are not depicted on this figure. On the real system, each task is repeated for each axis. Tasks reusing the results from previous feature extraction algorithms are implement sequentially. For example, the standard deviation is based on the variance, which was computed using the mean and the short time energy. Tasks using different function units are implemented in parallel. The execution time of each task is given in number of processing cycles in relation with the size of the input window. This application illustrates well how the dynamic reconfiguration can be used to implement multiple tasks with the same hardware resources.
3.3 Performance Evaluation

3.3.1 Comparison with different target technologies

Metrics are given for the evaluation of the performance and energy consumption of the system in Table 2. The target IGLOO corresponds to the implementation on the FPGA described in Section 2.3 (AGL1000V5). The design was also tested and implemented on a Xilinx Spartan6 FPGA (XC6SLX9) and a TI MSP430F1611 CPU for comparison. This CPU can be found on commercial wireless sensor nodes like the Moteiv TmoteSky [14]. For a fair comparison with the IGLOO FPGAs which do not integrate multipliers, the design on the Spartan6 was synthesized with and without using the Xilinx embedded multiplier blocks. The energy consumption values were estimated using power profiling tools provided by the FPGA vendors and an identical stimulus. The execution time and the power consumption of the CPU were directly measured on the board. The feature extractions algorithms were written in the C language with an optimization for streaming processing. With this approach, features in time domain can be computed within a few clock cycles. However, FFT must be executed completely to extract features in frequency domain. For all targets, a window size of 512 samples and a 16 bits fixed-point word length is assumed. FPGA designs run with a 10 MHz clock while the CPU runs at 8 MHz.

Table 2: Performance comparison on different target platforms for the complete feature extraction process

<table>
<thead>
<tr>
<th>Platform</th>
<th>Area</th>
<th>Time</th>
<th>Dynamic Power</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGLOO</td>
<td>45%</td>
<td>591μs</td>
<td>8.11mW</td>
<td>4.8μJ</td>
</tr>
<tr>
<td>Sp. 6 DSP</td>
<td>39%</td>
<td>566μs</td>
<td>5.66mW</td>
<td>13.6μJ</td>
</tr>
<tr>
<td>Sp. 6 Logic</td>
<td>64%</td>
<td>118ms</td>
<td>8.6mW</td>
<td>15.4μJ</td>
</tr>
<tr>
<td>CPU</td>
<td>-</td>
<td>-</td>
<td>12mW</td>
<td>1.4mJ</td>
</tr>
</tbody>
</table>

These results first show that the proposed design has the lowest energy consumption. The cell occupancy of the design justifies the usage of a reconfigurable structure. A dedicated function unit for each feature or multiply-accumulate units for each axis would make the occupancy of the design exceed the capacity of the chip. The execution time are the same for all FPGA designs since the similar source code and identical clock frequency are used. A speedup of round 200x was measured in comparison to the CPU implementation. This large difference comes mainly from the FFT implementation which is long on the CPU. The energy consumption of the CPU is consequently between two and three orders of magnitude higher. The dynamic power consumption of the IGLOO and the Spartan6 without DSP block are comparable. Lower dynamic power consumption is achieved when using the embedded DSP blocks on the Xilinx chip. However these savings are counterbalanced by the large static power consumption (leakage) of the Xilinx chips. The power profiling tool reported 14 mW of static power consumption against a few μW for IGLOO. This difference has a large impact on the total energy consumption, which is about three times lower for the implementation on the IGLOO FPGA. This trend confirms the analysis realized in [15].

Another aspect favoring IGLOO FPGAs is their Flash * Freeze mode. In periods of inactivity (i.e. between two samples), the chip is switched in a deep sleep mode with a total power consumption as low as 50 μW [16]. While switching in low power mode with configuration retention, Spartan6 FPGAs still suffer from quiescent currents of several mw. As sampling rates are usually low for the target range of applications, the duty cycle of the embedded device is low. When considering the FIFO operation described in section 3.1.1 with a sampling frequency of 100 Hz, the duty cycle of the hardware accelerator drops to 0.14%. This results in an average power consumption of 61μW for the IGLOO FPGA. We therefore conclude that an IGLOO FPGA is the best technology choice.

3.3.2 Comparison with specialized biomedical signal processing chips

We now compare the results of our implementation (IGL.) with the biomedical signal processing chip developed in [5]. We consider the three benchmark tasks given in the Table 3.

Table 3: Performance comparison with the biomedical signal processing chip [5]

<table>
<thead>
<tr>
<th>Platform</th>
<th>32-FIR</th>
<th>512-FFT</th>
<th>sin(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain</td>
<td>21x</td>
<td>4x</td>
<td>5x</td>
</tr>
</tbody>
</table>

As one would expect, the energy consumption of the ASIC is lower as our approach using an IGLOO FPGA. However, the gain in energy does not create
a gap as large as the comparison with a CPU. This low gain must be balanced with the energy spent for the wireless communication as it will be shown in the next section (see Fig. 9). Our design is a preferable solution since it can be adapted to a larger range of signal processing tasks without significant performance losses. The size of the chip is nevertheless at the advantage of the ASIC (12 mm² [5]) which makes it easier to integrate on a board requiring a minimal form factor, as it is often the case in biomedical applications.

Table 4 reports the performance comparison with Samsung’s ultra-low-power reconfigurable processor (ULP-SRP) in high-performance (HP) and low-power (LP) mode [8] for the implementation of a 256-FFT algorithm.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>242µs</td>
<td>839.1µs</td>
<td>1446.1µs</td>
</tr>
<tr>
<td>Power</td>
<td>7.4mW</td>
<td>20.7mW</td>
<td>12.6mW</td>
</tr>
<tr>
<td>Energy</td>
<td>1.79µJ</td>
<td>17.4µJ</td>
<td>18.2µJ</td>
</tr>
</tbody>
</table>

Our solution provides up to 10 times lower energy consumption than Samsung’s chip. This benefit comes from improvements in both execution time and power consumption. Our architecture relies on specialized reconfigurable functional units working at a low frequency (10 MHz) while the architecture of [8] uses an homogeneous array of standard reconfigurable processing elements working at a higher frequency (100 MHz). As a consequence, the FFT needs a significantly lower number of clock cycles to execute on our design. When scaling at the same clock frequency, the ULP-SRP will consume much less power than our implementation on the IGLOO FPGA. However, the customization of the reconfigurable processing units largely compensates this difference by drastically reducing execution times. For example, the butterfly operation which is the kernel of the FFT algorithm, can be executed within four clock cycles on our reconfigurable multiply-accumulate array.

In [8] the authors demonstrate with an EKG application that the ULP-SRP outperforms CoolBio [6], another low-power chip dedicated to biomedical signal processing. The EKG application is based on the continuous wavelet transforming [6], an algorithm extensively using the multiply-accumulate operation since it consists of filter banks. As we demonstrated that our design was performing better than ULP-SRP on the FFT, another algorithm relying on this operation, we can infer that our design would achieve lower energy consumption than CoolBio for the EKG application.

The authors of [4] reached 100nJ for the energy consumption of the 256-point FFT. However, FFT is the only algorithm that can be accelerated by their chip and all other computationally-intensive tasks must be implemented in software.

3.4 Benefit of Local Signal Processing

The main benefits from processing the biomedical signals directly on the sensor platform are the savings in memory and communication energy. In activity monitoring applications, the accelerometer signals are constantly monitored. Logging the raw data in an external flash memory is the straightforward approach. But writing in non-volatile memories has high energy costs. Similarly, the data could be directly forwarded to a remote server by using a wireless interface. Here again, wireless communication has a high energy cost which prevents a constant usage for data transmission. Table 5 reports the energy costs of writing and reading 256 Bytes in the Flash Memory (ST S25M80) of the TmoteSky module [14], as well as sending the same amount of data over the wireless link. These metrics do not take the protocol overhead into account.

Table 5: Energy consumption for Flash and wireless communication operations

<table>
<thead>
<tr>
<th></th>
<th>Writing</th>
<th>Reading</th>
<th>Sending (dBm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>20mA</td>
<td>4mA</td>
<td>17.4mA</td>
</tr>
<tr>
<td>Time</td>
<td>0.8ms</td>
<td>0.1ms</td>
<td>8.2ms</td>
</tr>
<tr>
<td>Energy</td>
<td>43.2µJ</td>
<td>5.4µJ</td>
<td>356.7µJ</td>
</tr>
</tbody>
</table>

We consider a scenario where one hour of accelerometer data sampled at 100Hz has to be processed. We estimated the energy costs of recording data in the Flash memory for three solutions: (1) The raw data (2 bytes per sample on 3 axes), (2) the features vector computed on a sliding window (42 features) (3) The recognized activity or class label (encoded as a single byte).

**Fig.8:** Estimated energy costs for a one hour log at 100Hz sampling frequency

Saving the complete feature vector is only energy-efficient for large windows or low overlapping rate while logging the activity label has always a lower
ASIC and FPGA consume a little amount of the total energy, CPU has a more significant share energy cost. When including the energy spent for extracting the features with our co-processing unit, 80% of the total energy is required to write the feature vector in the Flash memory. Other sources such as the CPU operating system will add up to this value such that the feature extraction energy represents a small part of the total balance. Decreasing further this energy as it is done in [5] will then have a mitigated impact on the final energy-efficiency of the application. Using the reconfigurable architecture provides however a better flexibility while keeping the benefits of local signal processing.

4. CONCLUSION

We introduced an architecture based on reconfigurable hardware for the processing of biomedical sensor data in eHealth applications. The utilization of dynamic reconfiguration reduces the power consumption of the device while enabling hardware acceleration of algorithms improving the accuracy of the signal processing chain. We demonstrated the efficiency of our system for an activity recognition application and compared the performance with other implementation alternatives and technologies.

ACKNOWLEDGMENT

We would like to thank Jude Ngepi, Aude Gressier, Reda Mohamed Fathia and Surapong Pongyupipanich for their contributions in this work.

References


Francois Philipp and Manfred Glesner


Francois Philipp received in 2009 a double degree in the field of computer engineering from the Ecole Nationale Superieure de l’Electronique et de ses Applications (ENSEA), Cergy, France and from the Technische Universitat Darmstadt, Germany. He is now a Ph.D. candidate in the Microelectronic Systems Research Group at Technische Universitat Darmstadt. His research interests include signal processing and reconfigurable hardware applied to wireless sensor networks.

Manfred Glesner received the diploma degree and the Ph.D. degree from Universit at des Saarlandes, Saarbrucken, Germany, in 1969 and 1975, respectively. His doctoral research was based on the application of nonlinear optimization techniques in computer-aided design of electronic circuits. He received three Doctor Honoris Causa degrees from Tallinn Technical University, Tallinn, Estonia, in 1996, Poly-technical University of Bucharest, Bucharest, Romania, in 1997, and Mongolian Technical University, Ulan Bator, Mongolia, in 2006. Between 1969 and 1971, he has researched work in radar signal development for the Fraunhofer Institute in Werthoven/Bonn, Germany. From 1975 to 1981, he was a Lecturer in the areas of electronics and CAD with Saarland University. In 1981, he was appointed as an Associate Professor in electrical engineering with the Darmstadt University of Technology, Darmstadt, Germany, where, in 1989, he was appointed as a Full Professor for microelectronic system design. His current research interests include advanced design and CAD for micro- and nanoelectronic circuits, reconfigurable computing systems and architectures, organic circuit design, RFID design, mixed-signal circuit design, and process variations robust circuit design. With the EU-based TEMPUS initiative, he built up several microelectronic design centers in Eastern Europe. Between 1990 and 2006, he acted as a speaker of two DFG-funded graduate schools. Dr. Glesner is a member of several technical societies and he is active in organizing international conferences. Since 2003, he has been the vice-president of the German Information Technology Society (ITIS) in VDE and also a member of the DFG decision board for electronic semiconductors, components, and integrated systems. He was a recipient of the honor/decoration of “Palmes Academiques” in the order of Chevalier by the French Minister of National Education (Paris) for distinguished work in the field of education in 2007/2008.