EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

State of the art baseband DSP platforms for Software Defined Radio: A survey

Omer Anjum[1*], Tapani Ahonen[1], Fabio Garzia[1], Jari Nurmi[1], Claudio Brunelli[2] and Heikki Berg[2]

Abstract

Software Defined Radio (SDR) is an innovative approach which is becoming a more and more promising technology for future mobile handsets. Several proposals in the field of embedded systems have been introduced by different universities and industries to support SDR applications. This article presents an overview of current platforms and analyzes the related architectural choices, the current issues in SDR, as well as potential future trends.

Keywords: Software Defined Radio, Pipeline Processors, RISC, VLIW architectures, Array and vector processors, SIMD, Adaptable architectures, Mobile processors, Heterogeneous systems

Introduction

Software Defined Radio (SDR) platforms and solutions are being actively pursued by both the industry and the academia. The purpose of SDR is to enable a programmable solution based on Digital Signal Processing (DSP) software running on a set of programmable processors and accelerators.

With the ever increasing user demands and resource consuming applications, particularly in Telecom Industry, pressure has been built up for developing not only new standards for communication but new architectures as well. The importance of wireless communication systems can be seen easily by the rapid increase in the number of its subscribers. It is not limited to cellular mobile communication like GSM, WCDMA, HSDPA or 3GPP LTE but it also includes other wireless standards such as WiMAX, Wireless LAN, DVB-H and DVB-T. This demand for seamless global coverage, wireless internet connectivity with additional capabilities like user controlled quality of service (QoS) have posed major challenges to keep the radio hardware and software from becoming obsolete, as new standards and techniques are developed in the future [1]. Wireless operators and manufacturers must respond to the changes and come up with new innovations in technology to upgrade or to fix any bugs discovered later.

The future trends of the evolution of standards can also be predicted easily. 2G (GSM, IS-95, D-AMPS, and PDC) systems opened the door for digital communication systems. Later on these systems were replaced by 3G (WCDMA/UMTS, HSDPA, HSUPA and CDMA-2000) technology, deployed in many parts of the world, ultimately going to be evolved as 3GPP LTE with higher data rates. The next is 4G which is further development to 3G, coping with the technological challenges more efficiently. As compared to 3G, data rates in 4G are much higher reaching up to 100 Mbits/s and even more. These higher data rates are in fact due to the use of VSF-OFCDM (variable spreading factor orthogonal frequency and code division multiplexing) and VSF-CDMA (variable spreading factor code division multiple access) as access schemes and also efficient concatenated (serial and parallel) error correction codes. To answer these big challenges of rapidly growing communication industry, we need a piece of reusable hardware that can work with different standards and protocols at different times to provide service providers and users most effective solution in terms of low cost, adaptability, high spectral efficiency, low latency and future needs. We need so much flexibility because with ever growing standards always changing the hardware causes huge costs and huge delays in the product development as well. This is the motivation behind the 'Software Defined Radio' (SDR [2]).

* Correspondence: omer.anjum@tut.fi
[1]Department of Computer Systems, Tampere University of Technology, P. O. Box 553, Tampere, 33101, Finland
Full list of author information is available at the end of the article

One of the biggest challenges in SDR solutions consist of achieving giga operations per second (GOPS) in the baseband processing, while at the same time keeping the power budget limited to a few hundreds milliwatts. In this article, we will just discuss the baseband processing solutions. The issues related to the digital transformation of the RF chain will not be considered.

Digital baseband technologies

Most of the very high data rate broadcast applications today are based on multi-carrier techniques. The basic principle relies on the fact that high data rate stream is divided into multiple low rate data sub-streams. Each of these sub-streams are modulated on different sub-carriers, which are all orthogonal to each other [3]. The main advantage of multi-carrier transmission is its reduced signal processing complexity by equalization in frequency domain and efficiency in frequency selective fading channels. Orthogonal frequency division multiplexing (OFDM) proposed in [4] has been widely adopted as a very efficient multi-carrier digital modulation scheme to realize such systems. In this article, we look at some of the SDR enabling solutions proposed today in perspective of the specifications mentioned in Table 1. The claims need to be closely looked at in order to identify or to suggest a new solution to enable SDRs. One fact important to mention here is that there is generally no agreed benchmark set in industry and academia as far as SDR is concerned, which can be used to evaluate and make a straight comparison for a certain implementation by each party. One vendor implements WCDMA turbo decoder, the other LDPC decoder, the third LTE initial synchronization and so on. There is no common input language for the SDR platforms, we would need to agree on the algorithms and allow implementations with different languages and intrinsics.

The major algorithms in an OFDM receiver chain to be processed by the baseband processor are related to channel coding, modulation, synchronization, channel estimation and equalization blocks. Now these tasks are briefly discussed here in order to understand their basic processing requirements.

Channel coding

Error correcting codes have a major role in channel coding. These codes generate some redundant information based on the actual message. This redundant information is exploited by the decoder in order to recover the actual message from the transmitted data corrupted by the channel. Today most of the OFDM systems deploy Convolutional Codes, Turbo Codes and LDPC (low-density parity-check) as forward error correcting algorithms. They imply substantially complex routing logic, memory and latency costs and perhaps the most computationally intensive part of the receiver baseband processing [5]. These channel decoding algorithms are different in nature as compared to other algorithms in a receiver chain which are very regular in data flow such as FFT, correlation, filtering etc. In channel decoding algorithms instead of actual computations data-transfer and storage schemes are the main contributors of power consumption and thus the efficiency matrices based on GOPs are no more valid [6].

Modulation

OFDM baseband symbol is generated by modulating N complex data samples using IFFT with N subcarriers. FFT/IFFT is perhaps one of the most area and power consuming block in OFDM transceiver design [7]. Cooley-Tukey algorithm is the most widely used for calculating FFT. In this particular algorithm, the total number of complex additions and complex multiplications required for radix-2 are $N^{*}\log_2(N)$ and $(N/2)^{*}\log_2(N)$, respectively [8], where 'N' is the number of points. The primary computational unit in FFT is the butterfly in which complex data elements are multiplied with a set of corresponding twiddle factors 'W_N^{nk}' the results of which are then added and subtracted [8]. The complexity of the butterfly depends strictly on the 'radix' of the algorithm. Hardware solutions for FFT usually implement higher radix algorithms like radix-4 and radix-8 due to the reduced number of computations but at the cost of increased complexity of the algorithm. Until now several architectures have been proposed like pipelined architecture, memory-based architecture, cache memory and array architecture. Hardware requirements for each

Table 1 Specifications for the standards considered in this article using OFDM as modulation technique [7]

	DVB-T		802.11 a/g	WiMAX	3GPP-LTE E-UTRA					
Carrier frequency (GHz)	0.4-0.8		2.5, 5.8	2-11	2					
Bandwidth (MHz)	6, 7, 8		20	1.5-28	1.25	2.5	5	15	15	20
FFT size	8192	2048	64	256	128	256	512	1536	1536	2048
Used subcarriers	6817	1705	52	200	76	151	301	901	901	1201
FFT period (μs)	896	224	3.2	8 (2 MHz channel)	66.7					
Constellation	QPSK, 16QAM, 64QAM		BPSK, QPSK, 16QAM, 64QAM	BPSK, QPSK, 16QAM, 64QAM	QPSK, 16QAM, 64QAM					
Maximum data rate (bps)	31.67 M (8 MHz channel)		54 M	104.7 M (28 MHz channel)	>100 M (20 MHz channel)					
Power requirement	Power consumption for baseband processing in a mobile handset must be within 1 W [44]									

of these architectures are different in terms of memory accesses, number of multipliers, number of adders, clock cycles etc. It is the designer that should make a compromise considering the specifications and available resources.

Synchronization

In order to correctly demodulate the received OFDM signal, the transmitter and receiver must be synchronized in terms of carrier frequency, carrier phase, sampling clock frequency and symbol timing. In case of any mismatch in carrier and clock synchronization, the performance of the system is severely deteriorated due to the presence of ISI (inter symbol interference) and ICI (inter channel interference). In OFDM, the designer can choose time or frequency domain for synchronization depending upon the system resources, performance, application requirements etc. In OFDM symbols, there is repetition in the received signal in the form of cyclic prefix or preambles of identical period which is usually exploited for synchronization. The basic kernel of the synchronization algorithm is cross-correlation or auto-correlation independent from the choice of algorithm. Either it is coarse and fine symbol timing estimation or it is carrier frequency offset estimation. IFFT can also be used in frequency domain synchronization if long latency is not a problem. In practice, linear-phase FIR matched filter banks are also adopted as a choice to implement correlation structures. In addition, frequency-domain and time-domain interpolators are used for the compensation of carrier frequency and sampling clock offset. They are usually realized as linear phase digital filters. In SCO (sampling clock offset) compensation, continuously updating the filter coefficients in real time may consume more hardware resources and even more when the number of taps required are increased [9,10].

Channel estimation and equalization

In order to correctly demodulate the OFDM symbol, it is very important to make a good estimate of the response of the channel and equalize the distortions caused to the transmitted signal. OFDM based communication systems often make use of the reference signal named as preamble or pilot for channel estimation [10]. Depending on the channel characteristics (low/high frequency-dispersive channel, low/high Doppler channel or low/high frequency selective channel), there are different pilot configurations to equalize each subcarrier in OFDM based systems [11]. In block type pilot symbols, pattern channel estimation is based on different estimators like minimum mean square error (MMSE), Low-Rank Approximation, LS (least square) estimator and reduced-order ML (Maximum Likelihood) estimator. MMSE and Low-Rank Approximation regard the channel as stationary random vector. Therefore, the prior knowledge of channel like the auto-covariance matrix and operating SNR is required which further increases the complexity. In MMSE, matrix inversion is required for each symbol [7]. In Comb-type pilot symbols pattern, we have time-domain windowing and frequency-domain interpolation. Time domain approaches need additional blocks for IDFT and FFT, which further increases the complexity of the system. Channel estimation based on grid-type pilot symbols pattern involves 2D MMSE interpolation, which has a very high complexity and thus avoided in practical OFDM systems [7]. In adaptive channel estimation, normalize-least-mean-square algorithm is the simplest to be implemented in hardware. RLS (recursive least square) and Kalman-filtering approaches are computation intensive. Adaptive filters are only suitable when normalize Doppler frequency is below 0.01 [7].

Overview of existing SDR solutions

Several alternative solutions to enable SDR proposed by industry and academia are considered in this section. For instance, in [12] the authors suggest that there are mainly two enabling directions for SDR that could be followed: the first one based on reconfigurable hardware, the second one consists of DSP-centered and accelerator-assisted architectures. The second approach guarantees high flexibility, but also suffers from problems related to high power consumption. To reduce the power consumption, such a platform should feature multiple DSPs running at a relatively low clock frequency. In the next section, we will analyze different solutions proposed to enable SDR based on the two approaches mentioned above (Figure 1).

Processor centered architectures

This section gives an overview of processor centered architectures, which is further categorized into DSP based and Many-Core platforms.

DSP-centered SDR solutions

This section provides an overview of some SDR solutions based on the DSPs with extra capabilities for exploiting the native data and instruction level parallelism of radio kernels. Some of these solutions are also assisted by accelerators. These solutions have been proposed during the last few years both by the industry and the academia.

LeoCore by CoreSonic

LeoCore [13] is an ASIP for radio baseband signal processing. This core is claimed to target cellular phones, laptop terminals, broadcast terminals, global positioning systems and embedded systems. The basic philosophy behind this architecture is first to identify the required baseband processing operations on algorithmic level of abstraction (such as Integer Data Filter, Correlation, Complex data filter, Decimators, Interpolators, FFT,

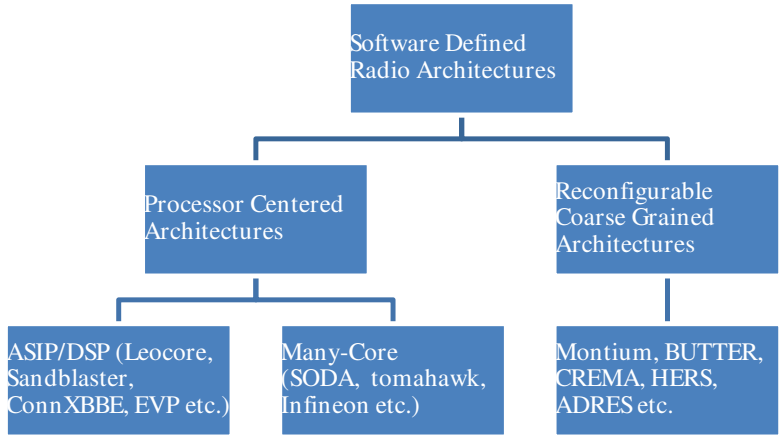


Figure 1 Categorization of SDR solutions.

DCT, Walsh Transform, Frequency domain filters, Matrix computations in time and frequency domain, Bit manipulations for forward error correction, Division, Square root, Waveform generator, Look Up Table logic, $1/x$), and then map them onto a suitable processing core such as a Single Instruction Multiple Data (SIMD) processor or an ASIC accelerator.

The abstracted information on algorithmic level for radio baseband processing reveals the fact that 90% of the time is consumed by the processes defined above. The basic optimization of the core is thus done to provide acceleration to 90% of the code.

Thus, depending on the nature of computations the LeoCore's architecture is divided into four processors optimized differently to handle different set of operations. These processors are categorized as: Digital Front End, Complex Data SIMD processor, Function accelerator, processor for control signals and miscellaneous functions (Figure 2).

The instruction set architecture is strictly covering only the required functions mentioned above and the flexibility beyond this domain of algorithms is avoided and it is not meant to run general purpose applications. There is a tradeoff between efficiency and flexibility at the instruction level. For example FFT N is a single instruction for N-step butterfly computing and cannot be used for other purposes. There are both accelerated instructions (task-level and vector instructions) and RISC instructions for simple arithmetic operations, data moving, program flow control and hardware/software configurations. The two main problems regarding optimization are data latency and power. The proposed solution to latency in this architecture is to use the task parallelization, scheduling and parallel data memory access [14]. To optimize power, they proposed to shut down the idling circuits and memory modules.

LeoCore is provided with Coresonic developer studio (CDS), a development platform including a cycle-true and bit-true simulator as well as assembler and debugger.

It is claimed that LeoCore [13] can handle all of the standards mentioned in Table 1. However, it appears that only DVB-T/H and WiMAX benchmarks were published in 2008. The system measurements found in the publications or on company's website are shown only for DVB-T/H [15]. It consumes 11 mm^2 in 0.12 μm CMOS process including 1.5 Mb of single port memory and 200 K gates logic. Peak power consumption is 70 mW@70 MHz for highest data rate of 31.67 Mb/s.

Sandblaster by SandBridge

SandBridge Technologies has offered a multicore multithreaded vector processor named 'Sandblaster' as a solution to SDR complying with the low power requirements. Sandblaster includes a combination of three units: instruction fetch and branch unit, an integer and load/store unit and a SIMD vector unit. Sandblaster 1.0 was targeted at implementing the physical layer of 3G wireless standards, with peak data rates of up to 15 Mbps. Later they proposed Sandblaster 2.0 to support 4G standards which was just an extension of version 1.0 that kept its philosophy. Vector registers connected to 64-bit data path were extended from 16 to 256-bit connected to 256-bit data path in version 2.0. In addition, the mask and accumulator registers expanded from 4 and 40 bits to 32 and 64 bits, respectively. In version 2.0 a SIMD operation can operate on 16 (short) or 8 (integer) values in parallel in contrast to 4 values in version 1.0 [16] (Figure 3).

Some of the key focuses are support for high-level programming language like C and compiler optimization for DSP. The need for compiler design in parallel with the DSP architecture design is particularly emphasized in their

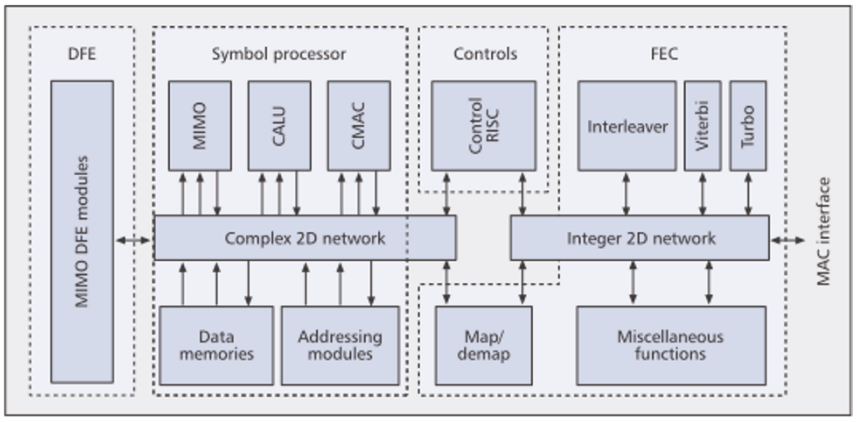


Figure 2 LeoCore Architecture [13].

design cycle for the whole system. The proposed compiler analyzes the C code and extracts the DSP operations itself. Compiler makes use of the data level parallelism in the C code and appropriately generates SIMD vector operation. Another important aspect is the Sandblaster's Token Triggered Threading (T^3) which features compound instructions, SIMD vector operations and greater flexibility in scheduling threads. Instructions issued from multiple threads are executed in parallel each cycle.

Several SDR Platforms, each using Sandblaster DSP core, have already been developed and tested by SandBridge technologies. For instance, SB3011 has four DSP cores running at minimum 600 MHz at 0.9 V each of which is 8-way multithreaded and can execute 32 independent instructions. It has already been tested for WiFi 802.11b, GPS, AM/FM radio, Analog NTSC Video TV, Bluetooth, GSM/GPRS, UMTS WCDMA, WiMax, CDMA and DVB-H [17]. Similarly SB3500 has three cores, each capable to handle SIMD instructions with four threads. This particular platform successfully targeted to handle LTE category 2 baseband processing [18]. The chip is fabricated on 65 nm and it is fully functional, providing nearly 30 GMACs at 600 MHz [16].

ConnX BBE by Tensilica

Tensilica has offered ConnX baseband engine, SIMD architecture, as a solution to SDR. It is claimed that it is

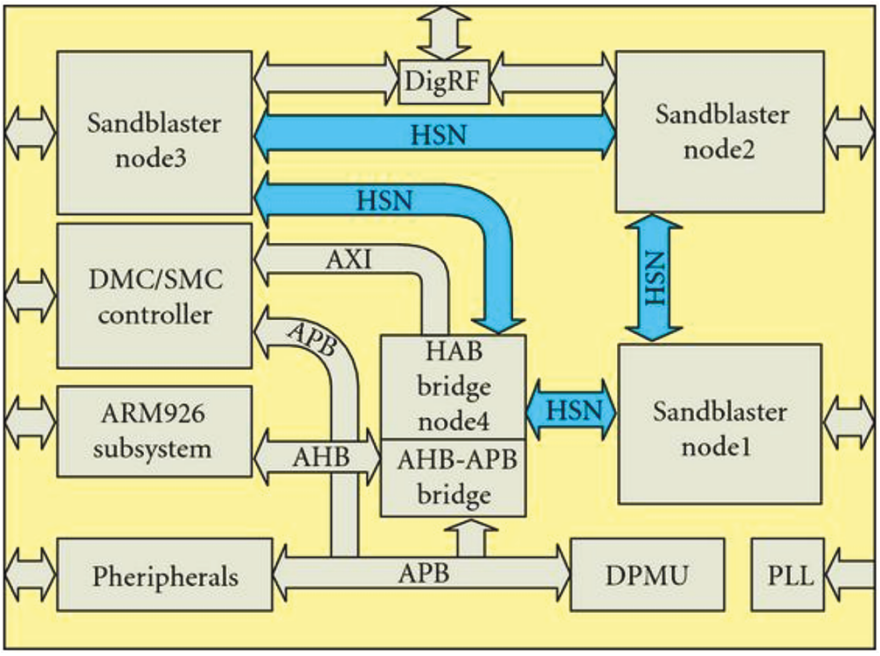


Figure 3 SandBridge's SB3500 SDR platform with three Sandblaster Cores [40].

an intermediate approach that do not use power consuming wider data paths at higher clock rates as scaled up conventional DSP and that has targeted only flexible functional blocks to enable SDR. This baseband-oriented DSP is a licensable processor core which uses Tensilica Xtensa template processor as a foundation. Different processor configurations according to the application requirements are generated using tools like Xtensa Processor Generator and Tensilica Instruction Extension. The configuration includes the choice of memory system, optional instructions and interfaces, custom instructions and I/O interfaces specified by Tensilica TIE language. There is a range of optimized instructions provided to meet the high throughput of DSP baseband operations like FFT, Complex multiplication, vector division, vector reciprocal, square root etc.

One important aspect is the vectorization analysis of an application program to efficiently exploit the inherent parallelism in DSP operations and restructure it accordingly. Developer can vectorize the program himself using ConnX BBE's data type and intrinsic function. In addition Xtensa C and C++ compiler can automatically do this vectorization with little or no human intervention (Figure 4).

ConnX BBE's SIMD processor at 400 MHz (6.4×10^9 MAC operations per second) can do sixteen 18-bit multiplications, eight 20-bit additions or four 40-bit additions in parallel and also gives 13 GB per second data memory access bandwidth. It also accommodates three-way VLIW instructions with the first slot for Load/Store operation or Xtensa core instructions. The second slot is for real and complex multiply, FFT or any vector selected operation. The third slot uses the second Load/Store unit or is for arithmetic and logical operations. A wide range of instructions they have developed specializing the domain of operations particularly for SDR transceiver design.

The BBE when optimized for performance takes 1.1 mm^2 (430 K gates) in the TSMC 65LP process. For minimal area, the synthesis results in 230 K gates [19].

EVP (embedded vector processor) by NXP

NXP proposes a hardware architecture featuring a VLIW vector processor named EVP [20] targeted to support 3G standards. According to NXP the digital baseband processing for SDR can be split into three fundamental parts: filter, modem and codec. The filter stage should be as configurable as possible. The modem stage is the part that is most affected by different standards and implementations. For this reason, this stage should be kept programmable, thus flexible. The codec stage, instead, is made up of standard functions which remain similar among standards and are characterized by high processing requirements. Therefore, the codec stage does not benefit from programmability and is instead usually implemented in ASIC accelerators.

As mentioned in the previous chapter, data parallelism abounds in SDR applications. For this reason, using SIMD DSP processors appears like a natural choice. NXP adds to the SIMD capabilities also VLIW capabilities in the EVP processors, trying to provide a

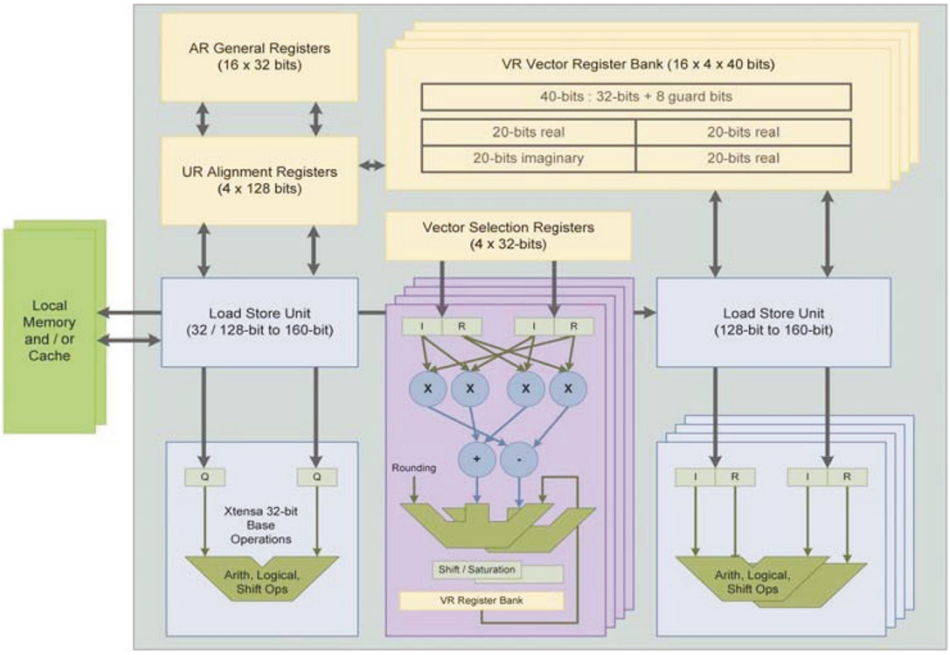


Figure 4 ConnX Baseband Engine [41].

comprehensive coverage of the parallelism available. VLIW capabilities help in accelerating several kernels, including rake receivers and FFT. VLIW parallelism is provided on the top of vector parallelism. The hardware supports also functionalities like zero-overhead looping, parallel address calculations and loop control, as well as intra-vector shuffling and arithmetic operations (very useful in FFT and Viterbi trellis construction). The EVP can handle 8-bit, 16-bit or even 32-bit data within the data vectors. The supported data types are integer and fixed point, supporting also complex numbers natively (28 or 216 bits). The vector size is 256 bits.

EVP has its own EVP-C compiler which includes extensions to support vector data types and intrinsics to support vector operations. Due to the lack of efficient vectorizing compilers available today, the compiled C code can be executed on the programmable host microprocessor which acts as system controller, while the intrinsics are converted into machine instructions for the vector processors, which act as number crunchers.

In a 90-nm CMOS process, the area of the EVP processor core is about 2 mm^2 (450 K Gates), runs at 300 MHz, and dissipates about 0.5 mW/MHz (considering only the core) and 1 mW/MHz (when considering also the memory system) (Figure 5).

NXP and Nokia proposed a real 'multi-radio computer' [21] as a result of a joint research project. Indeed, one of the major challenges of future SDR architectures consists of guaranteeing support for different radio protocols running concurrently. In particular, the Nokia-NXP SDR supports HSPA, DVB-T and WLAN active simultaneously on a shared hardware, as well as an SDR operating system which is able to schedule and support dynamic multi-radio operation.

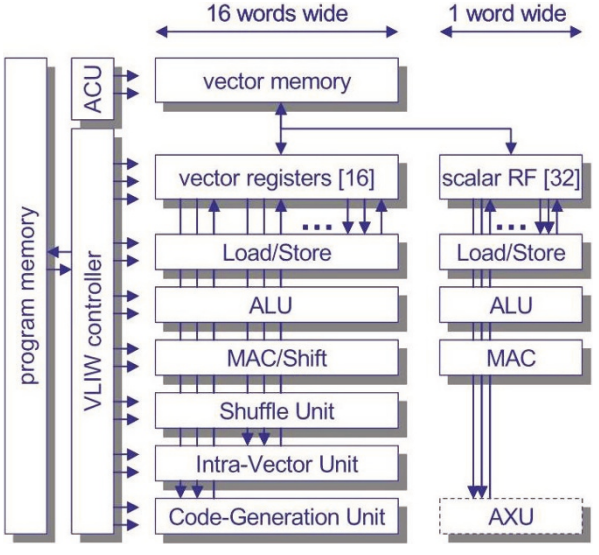


Figure 5 NXP's EVP architecture [42].

Many-Core SDR Platforms

This section provides an overview of some SDR platforms based on the idea of using multiple cores. The bigger tasks are broken into smaller ones and thus divided among the cores. Let us have a look on some of this kind of proposed solutions.

SODA (signal-processing on-demand architecture)

SODA takes the motivation for targeting mobile handsets aiming at reduction in power consumption to an acceptable level. The basic philosophy behind SODA architecture is based on dividing the whole processing domain between Data Processors and Control Processor. Data Processors are meant for computing computationally intensive DSP kernels like FFT, FEC kernels, Cell search and LPF. Control processor is meant to perform system operations and manages data processors through remote procedure calls and DMA operations. SODA is made up of four cores, a control processor and global scratchpad memory. These components are connected through a shared system bus. The cores contain dual pipelines which are able to support scalar and 32-wide SIMD operations. The arithmetic functional units are characterized by a 16-bit datapath, since 32-bit arithmetic was considered not necessary. Each core consists of a scalar unit and a vector (SIMD) unit (Figure 6).

An important aspect of this architecture is that it does not adopt multithreading approach, dividing the kernels into threads. Instead protocols are pipelined into kernels and statically assigned to one of the ultra-wide SIMD SODA processing elements. This is due to the fact which was observed during the design process of SODA that the inter-kernel communication throughput is very much lower than that of intra-kernel computational throughput. SODA here in fact discourages to have multithreading solution for a communication baseband processor design based on the observed fact. For inter algorithm data communication scratch pad memories are suggested in SODA platform. The scratchpad memories were proposed in streaming applications for multimedia processors like Imagine [22] and IBM Cell Processor [23] and later adopted by SODA to handle the streaming data between the algorithms.

SODA satisfies the throughput requirements of the 2 Mbps W-CDMA protocol (and of the 24 Mbps of the 802.11a protocol) running at 400 MHz. The area occupation is projected to be 6.7 mm^2. Results show that in a 180 nm technology, SODAs power consumption is 3 W, which is too much for current mobile phones constraints. It was also implemented on 90 and 65 nm technology, achieving power consumption of 450 and 250 mW, respectively [24].

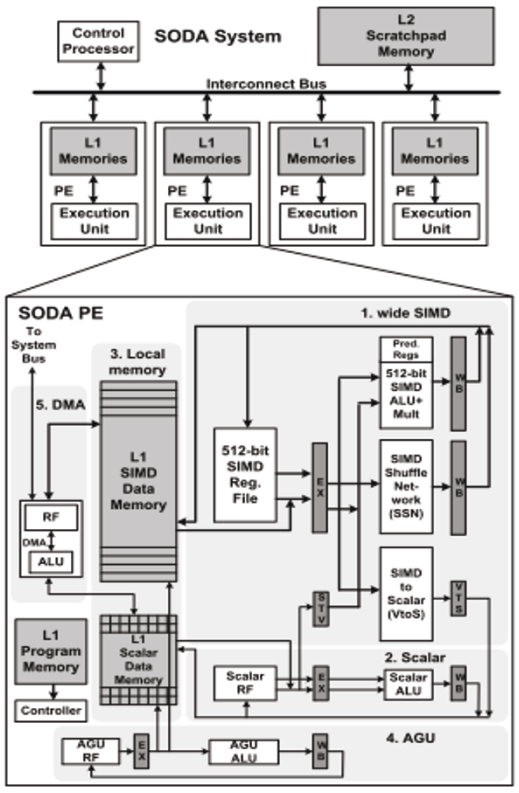


Figure 6 SODA multi-core DSP architecture [25].

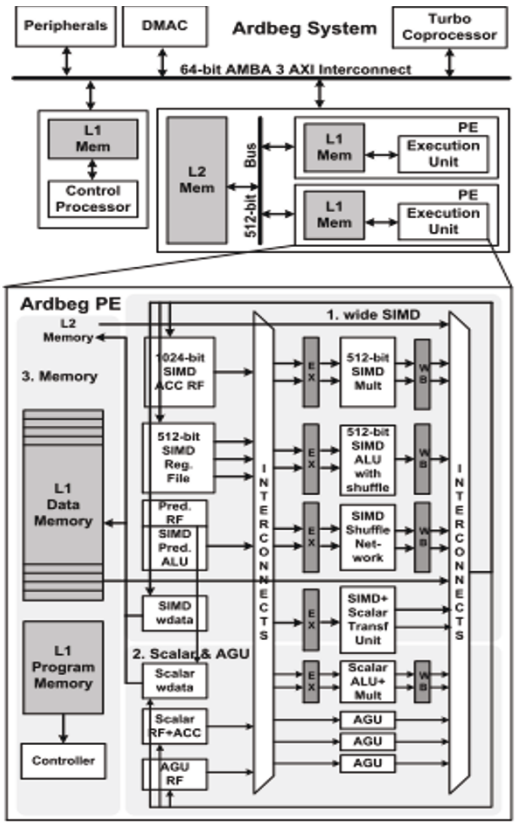


Figure 7 Ardbeg multi-core DSP architecture [25].

ARM Ardbeg

Ardbeg [25] is a commercial prototype based on revisiting SODA architecture (Figure 7). The main changes present in Ardbeg when compared to SODA consist of an optimized wide SIMD Design, its related VLIW Support, and algorithm specific hardware acceleration. Ardbeg is a multicore architecture, with one processor for control purposes and multiple Processing Elements for DSP operations. Ardbeg also features some special ASIC accelerator which is dedicated for specific algorithms like Turbo encoder/decoder, as well as operations like block floating point and fused permute and arithmetic operations. The memory hierarchy is conceived so that each PE has a local scratchpad memory and shares a global memory. Each of these memories is explicitly managed via DMA transfers between the local memories of the PEs, as well as to and from the global memory.

The evolution of SODA to Ardbeg implies making some design choices like keeping 32-lane 512-bit SIMD datapath for the DSPs (because they claim that it is the best SIMD design choice in 90 nm technology). Moreover, in creating Ardbeg they redesigned the internal SIMD shuffle network used to support vector permutation operations.

Ardbeg also introduces support for VLIW operations, enabling to issue two SIMD operations per clock cycle. Still, Ardbeg implements only a restricted version of VLIW operations: the aim is being able to support well common parallel operations present in SDR algorithms, while at the same time keeping the hardware relatively simple and thus less expensive. The development tools include the C-language support and even can take the C-language model from Matlab for compilation.

The Ardbeg system runs at 350 MHz in 90 nm technology, and dissipates approximately 500 mW. Ardbeg's efficiency is due to several factors. In particular, to a 2-way LIW execution of SIMD operations, together with ASIC coprocessors and a Banyan shuffle network. Still, according to [25], ASIC-based solutions are still much more power efficient than current SDR solutions.

Tomahawk MPSoC

Tomahawk is a heterogeneous single chip SDR platform. As many other solutions it also exploits instruction, data and task level parallelism. Its distinctive feature might be its CoreManager which is a dedicated run-time scheduler hardware unit (Figure 8). It consumes two Tensilica RISC processors to execute OS and control functions, Six Vector DSPs, an ASIP each for LDPC

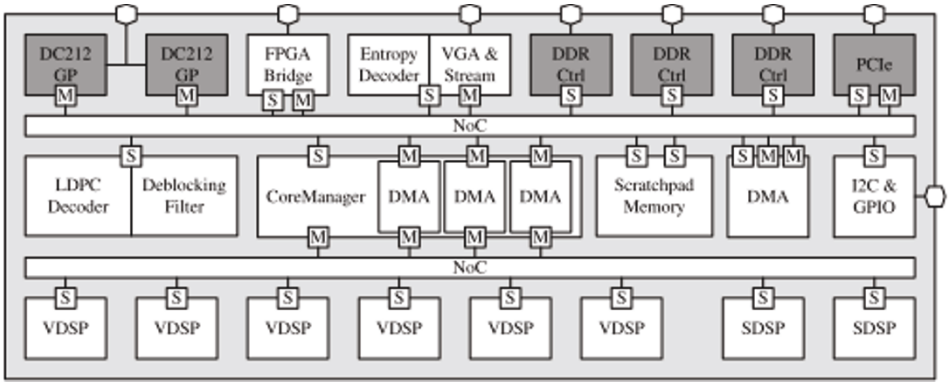


Figure 8 Tomahawk MPSoC architecture [43].

decoder, de-blocking filter and entropy decoder. Each of these units use data locality principle based on synchronous transfer architecture [26] for low power consumption.

Its programming model must be mentioned here as it is the key distinguishing factor from other solutions. The tasks are basically converted to task descriptions at compile time. These descriptions are continuously sent by the control unit to CoreManager with maximum queue length of 16 tasks. The spatial and temporal mapping of these tasks onto the PEs is then done automatically by the CoreManager. This programming model relaxes the programmer from time taking scheduling of the tasks thus decreasing the time of whole design cycle.

Tomahawk is claimed to have been tested for LTE and WiMax. Fabricated on 0.13 μm CMOS process it runs at 175 MHz with peak performance of 40 GOPS and with 1.5 W power dissipation which is too high for mobile units.

MuSIC by Infineon

One of the proposals by Infineon for SDR is the MuSIC-1 chip. MuSIC is included in a system powered by a programmable microprocessor few DSP processors, plus some ASIC accelerators. The DSPs have SIMD capabilities to exploit data parallelism. The SIMD cores are put together in a cluster, where each DSP is coupled with programmable processors for operations like filters or channel encoding and decoding. The number of SIMD cores can be increased or decreased according to the processing requirement.

Each of the SIMD cores cluster consists of four processing elements (PEs), and its working clock frequency is 300 MHz. These cores support advanced features such as saturating arithmetic and finite-field arithmetic. Moreover, it supports long instruction word (LIW) features for arithmetic operations, memory accesses and data exchange between the PEs (Figure 9).

MuSIC-1 chip was used for complete standards like WLAN and WCDMA, and according to [26] the related results showed how SDR baseband solutions for mobile phones are competitive with respect to power consumption and area in 65-nm CMOS. As specified in [26], MuSIC (multiple SIMD core) chip is Infineon's SDR prototype solution, originally designed in 90-nm CMOS

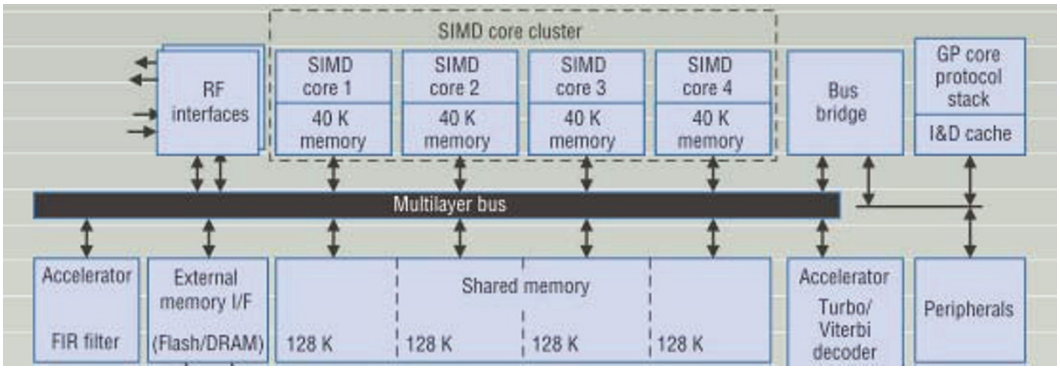


Figure 9 Infineon's MuSIC-1 chip's Baseband DSP with 4 SIMD cores [43].

technology, featuring 28 million transistors, 6 Mbits of SRAM, and six layers of wiring; its area occupation is 57 mm^2.

Reconfigurable architectures for SDR platforms

There have been numerous SDR solutions based on reconfigurable hardware. Some examples are: Montium, ADRES, HERS, Butter and CREMA.

Montium by recore systems

Recore Systems has offered coarse grained reconfigurable Montium technology as a solution to enable SDR. They define reconfigurable systems as the one in which hardware adapts the algorithm instead of algorithm adapts the hardware. Montium Tile Processor targets computational intensive kernels of 16-bit DSP domain. It can support both floating point and fixed point operations. It does not fetch instructions and resembles more like an ASIC instead of DSP avoiding von Neumann bottleneck. There are 10 global buses to provide the interconnect flexibility to be changed in even every clock cycle depending on the data flow. The other distinguishing feature of Montium is its multi-level ALU. Each ALU has two levels, one for general purpose computing and another for functions like FFT and Filtering. These levels can be bypassed according to the needs of the algorithm.

Montium's configuration overhead is less than 1 kb and takes less than 5 μs. It can be used as a single accelerator or as a part of heterogeneous MPSoC. It comes with its own design tools named as Montium Sensation Suite which has a Compiler, Simulator and Editor. Compiler uses its proprietary language called Montium Configuration Design Language (CDL) for reconfiguration.

There are some implementations of different communication standards done by Recore Systems. A flexible rake receiver can be implemented on a single Montium TP. Configuration size and time are 858 bytes and 4.29 μs. At run time number of fingers can be changed from 2 to 4 in 120 ns. HyperLAN/2 can be implemented on three Montium TPs. System can run fairly between the clock frequencies of 25 to 75 MHz. Configuration overhead is just 274 to 946 bytes. Viterbi decoder which can change its rate and decision depth depending on the application can be implemented on a single Montium TP. The initial reconfiguration requires 1376 bytes to be loaded in less than 7 μs at configuration clock frequency of 100 MHz [27]. The maximum FFT size that can be computed on one Montium TP is 1024 depending on the size of local memories. It takes around 5140 clock cycles or 51.4 μs at 100 MHz. In addition, the implementation of various DSP algorithms on Montium can be found in [28].

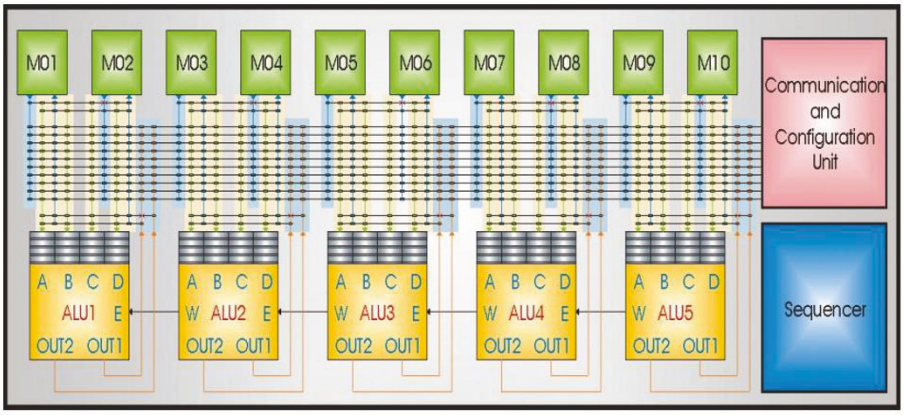
On 0.13 μm CMOS technology Montium covers 2 mm^2 with 10 kbs of SRAM. Its power consumption is 600 μW/MHz including memory access [29] (Figure 10).

BUTTER and CREMA

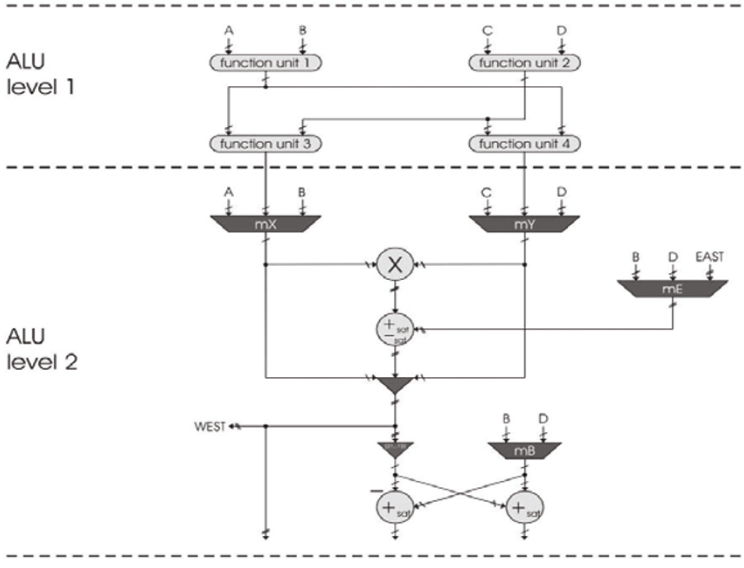
BUTTER is a coarse-grain reconfigurable array developed at Tampere University of Technology [30]. In this case, the demand of flexibility is satisfied by run-time reconfigurability, while the array structure provides the high data throughput needed by SDR applications. Its parametric template can gain any size of matrix but as a popular case currently BUTTER array is composed of a matrix of 4 × 8 processing elements, whose functionality and interconnections can be defined at run-time. Each processing element can perform different kind of arithmetic operations (integer and floating-point) between 8-, 16- and 32-bit values. Reconfiguration time varies between one clock cycle (in case that the context is already stored in the local configuration memories) and a few tens of cycles (if the context must be loaded from an external memory).

The array is meant to be used as a coprocessor in combination with a general-purpose processor core. In our platforms, BUTTER is coupled with an open-source processor core called COFFEE [31]. In the platform, COFFEE is meant to be used as a global controller, while the array performs data intensive computation. The exploitation of the large throughput of BUTTER is possible using two local data memories to store input operands and results. The adoption of a ping-pong mechanism allows the sequential processing of the data stream using different configuration contexts and without requiring additional data transfer to and from the system memory. Cell search algorithm from W-CDMA standard [32] as well as FFT [33] required for OFDM-based protocols have been both successfully mapped on the platform.

Lately, a new reconfigurable core has been designed as an evolution of BUTTER. The new core, called CREMA, introduces design-time adaptability that allows modeling the architecture of each PE according to the application requirements. This feature reduces the flexibility of a specific instantiation of CREMA, but produces better results in terms of operating frequency of the reconfigurable array in particular for an FPGA implementation of the IPs. Considering the synthesis on an Altera StratixII FPGA, we can see a significant difference in terms of area utilization between BUTTER and two different customized versions of CREMA. The two versions are customized for matrix multiplication algorithms. The first version supports only integer arithmetic, while the second version provides also a context for floating-point operations. After the synthesis, we noticed that the integer version of CREMA is 90% smaller than BUTTER. However, the adaptability guarantees a significant improvement also in case of floating-point computation, because it is still 80% smaller than BUTTER. This large difference can be explained considering the



a.



b.

Figure 10 Montium Processor Tile and Montium ALU[28]. **(a)** Montium Processor Tile [28]. **(b)** Montium ALU [28].

customization of the interconnection logic in CREMA implementation [23].

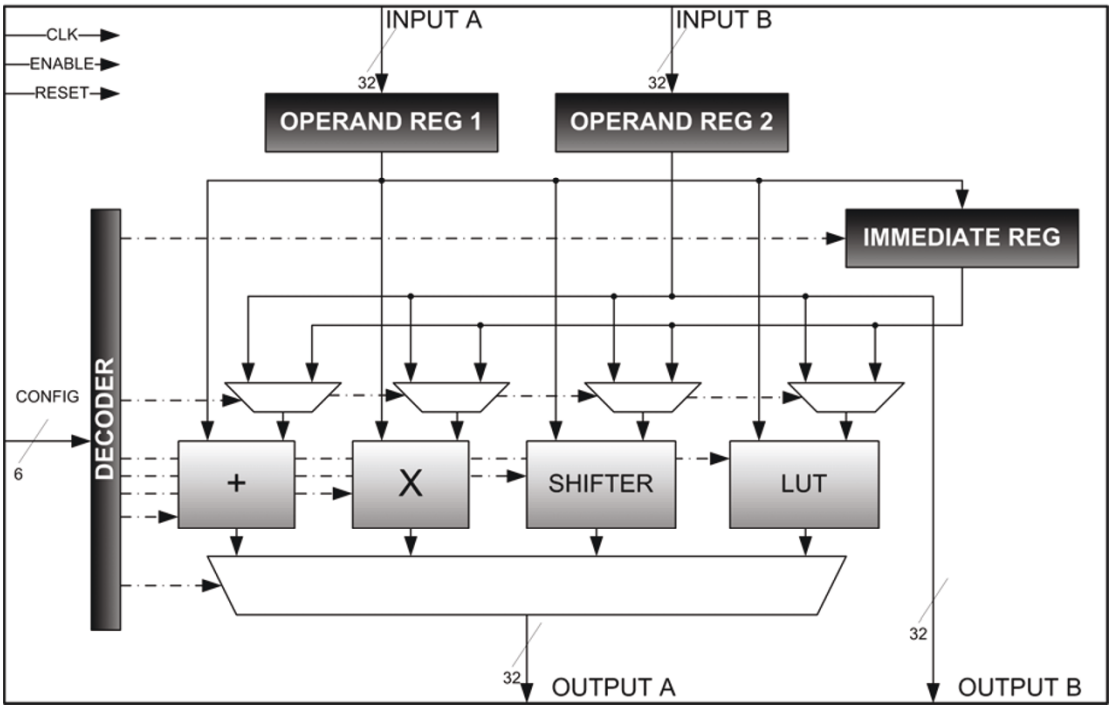
The latest research results regarding CREMA implementation show that the execution time for 64 and 1024-point FFT meets the timing constraints for 802.11 a/g (4 μs) and 3GPP-LTE (66.7 μs) using an optimized version of the CREMA implemented on a Stratix IV FPGA [9] (Figure 11).

HERS

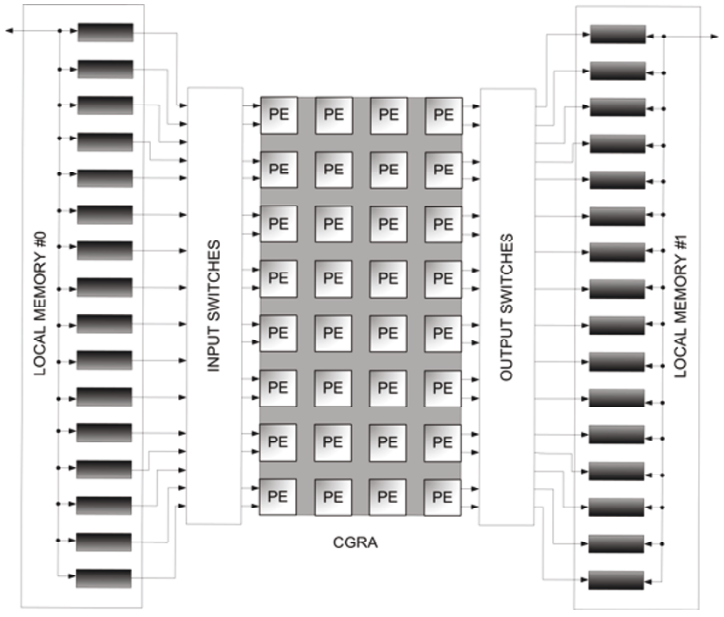
HERS [34] is a Heterogeneous Reconfigurable System aimed to serve as a platform for SDR. Its main idea is to divide the application among the reconfigurable engines (REs) based on the nature of computations. The RE further consists of a processor which comprises of a pool of homogeneous processing elements optimized to perform a class of wireless algorithms. To provide the inter-engine communication there is a high speed bus available. The REs do the time-multiplexing among the tasks associated to them. In the case of OFDM, the system comprises two REs: Modem Engine and Channel Coding Engine (RECFEC) [34] (Figure 12).

In a PE pool, the PEs are connected as a two-dimensional array of size 8 × 8. It can be seen from Figure 12b, that the four functional units in the PEs corresponding to the two engines are different in nature depending on the algorithm requirement.

The programming model is based on partitioning applications into sequential and parallel tasks [35]. The control code and the serial tasks are performed by the TinyRisc instructions and the parallel tasks are mapped



a.



b.

Figure 11 BUTTER processing elements(a) and BUTTER & Crema Architecture [32]. **(a)** Architecture of a processing element in BUTTER [32]. **(b)** Butter and Crema Architecture.

on the processing elements. In [34] is illustrated the implementation of W-LAN and DVB-T/H on HERS meeting the real time constraints at 250 MHz on 90 nm TMSC technology.

ADRES by IMEC

IMEC presents a hybrid CGA-SIMD SDR processor design based on ADRES/DRESC framework [36]. The core of the architecture consists of a Global Control

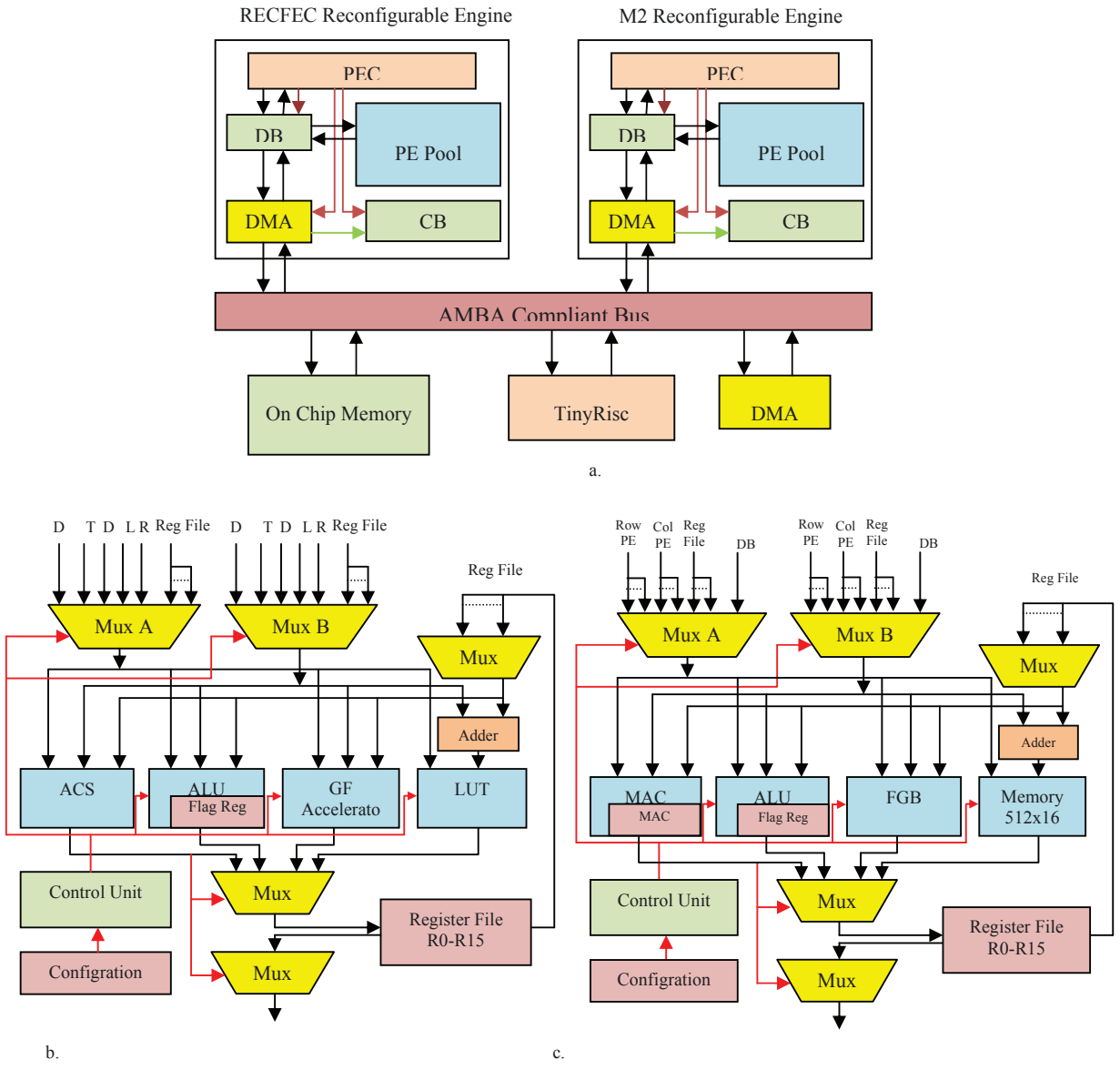


Figure 12 HERS model and Processing Element Architecture of RECFEC & Modem Engine [34]. **(a)** HERS model [34]. **(b)** Processing Element architecture of RECFEC [34]. **(c)** Processing Element architecture of Modem Engine [34].

Unit, three predicated VLIW Functional Units, a Predicate Register File and a Coarse Grain Array (CGA) module. The CGA module is made of 16 inter-connected units out of which 3 are directly connected to the predicate register files. The architecture can function as a VLIW processor or a true Coarse Grain Reconfigurable Architecture (CGRA) machine depending on the application requirements; the machine is able to switch on the fly at run time between those two modalities.

The architecture is fully programmable in C-Language compiled with DRESC framework [37]. To exploit 4-way SIMD capabilities some intrinsic functions are added in the C code. The DRESC framework is used to transparently compile a single C language source code to both the VLIW and the CGRA machines.

The processor, designed in TSMC 90G process according to a dual-VT standard-cells flow, achieves a clock frequency of 400 MHz in worst case conditions and consumes maximally 310 mW active and 25 mW leakage power (typical conditions) when delivering up to 25.6 GOPS (16-bit). The mapping of a 20 MHz 2 × 2 MIMO-OFDM transmit and receive baseband functionality is detailed as an application case study, achieving 100 Mbps+ throughput with an average consumption of 220 mW [36] (Figure 13).

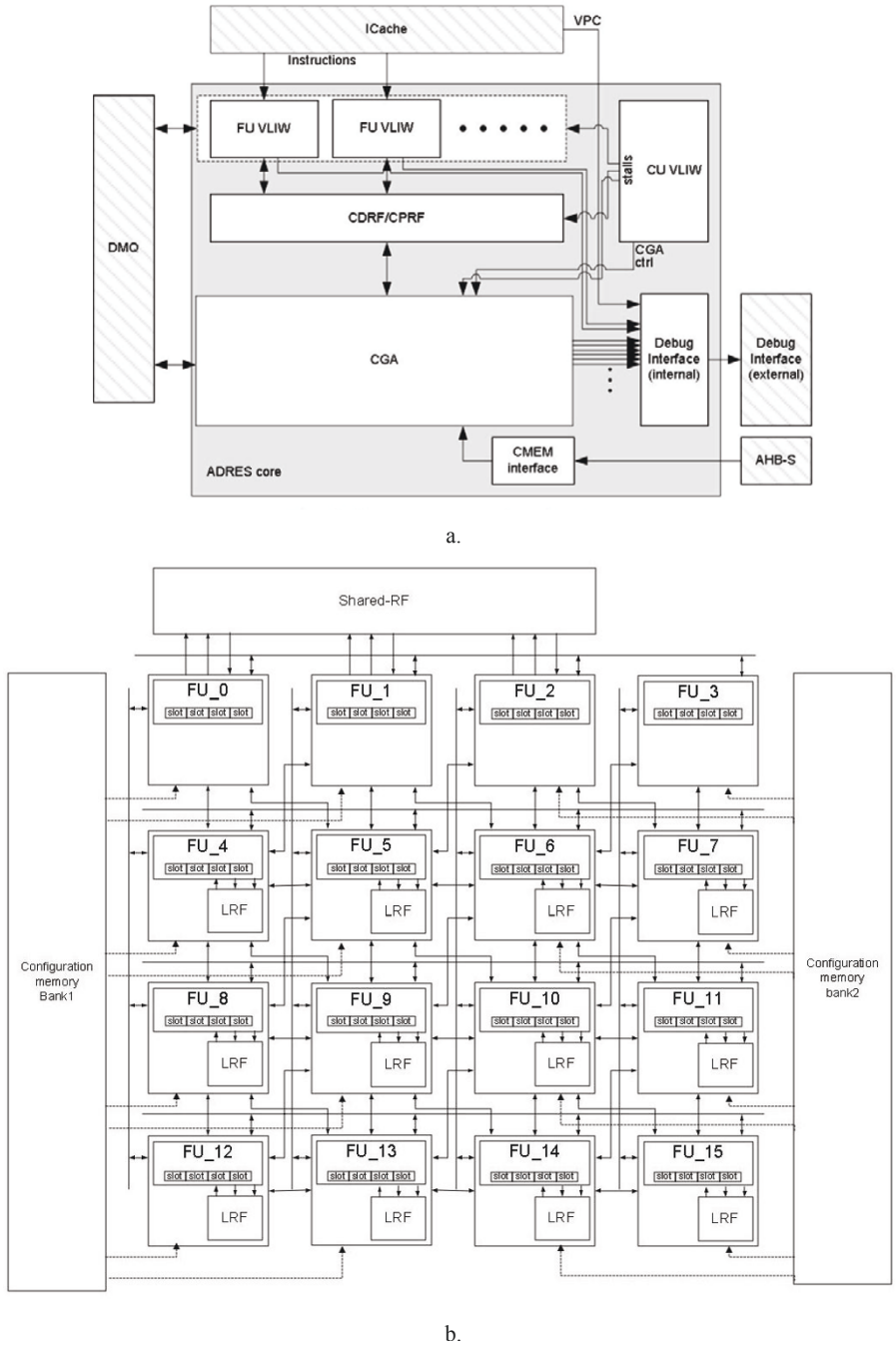


Figure 13 ADRES Processor Core Architecture and CGA Unit [36]. **(a)** ADRES Processor Core Architecture [36]. **(b)** CGA Unit [36].

Discussion

Based on the previous examples, we can now summarize the state of the art in SDR solutions and try to predict the main trends for the future without making a straight comparison of area, power and flexibility among different approaches because without a benchmark and based on available public figures it is not really possible as agreed in [38] as well. We still give some comparisons on the basis of programmability, flexibility and power in Tables 2, 3 and 4, respectively, at the end of the discussion which can provide clearer picture of the systems we have already seen in this article. It is apparent how approaches based on reconfigurable hardware come at the moment from the academic world, while proposals from the industry remain anchored to the DSP-based approach. This might be mainly due to the fact that

Table 2 Programmability

Architecture	SW/high level language	SW/ad-hoc language	SW/ assembly	Specific optimizations	Available support for high level language
LeoCore	✓		✓		Coresoninc developer studio
Sandblaster	✓			Auto-Vectorization, Token Triggered Threading	Sandbridge's optimized C compiler
ConnX BBE	✓	✓		Auto-Vectorization	Needs for TIE language
EVP	✓		✓		EVP-C compiler
SODA	✓		✓	Static compile-time scheduling and allocation, Compiler based task assignment to PEs	C compiler generated by OptimoDE's Framework, Matlab C-model supported
ARM Ardbeg	✓		✓	Static compile-time scheduling and allocation, Compiler based task assignment to PEs	C compiler generated by OptimoDE's Framework, Matlab C-model supported
Tomahawk	✓			dynamic Task scheduling and allocation	C compiler
Infineon	✓				MuSIC specific C compiler to support SIMD C Extensions
Montium	✓	✓			Montium specific C compiler
BUTTER & CREMA	✓	✓			FireTool to support C language Extensions
HERS	✓	✓			
ADRES	✓		✓		DRESC Framework

programmers and developers worldwide are used to the programming paradigm based on C language extended with assembly code (or intrinsics). Conversely, reconfigurable machines (due to their novelty and relatively recent appearance on the scene) are considered by the large public to be not that easy to program. For this reason, huge efforts are being spent to make reconfigurable hardware easier to use by third-party programmers. The promising results achieved so far, together with the high potential of such machines make them a very good candidate for the next future.

For the time being, systems are still likely to follow this paradigm: a programmable microprocessor acts as a system controller and is connected via a multilayer hierarchical bus to a series of subsystems hosting either ASIC components, ASIP processors [26] or VLIW DSP processors with SIMD capabilities.

In the near future, it is likely that we will witness an evolution of this paradigm consisting of adopting NoCs to interconnect an increasing number of subsystems, each hosting an increasing number of computation resources (in order to face increasing requirements of future radio applications: LTE, LTE-A and so forth). We cannot be certain about the applications that need to be supported in the coming future. The same chip should be used for as many years as possible to amortize the development costs, while standards keep evolving. If we consider only the standards mentioned in Table 1 they have been used for many years, and others (e.g., LTE) are likely to be there for a long time as named.

Table 3 Flexibility

Architecture	Support GP applications	GP but optimized for radio kernels	Only Radio Kernels	Template-based design	Easily adaptable to new/updated radio standard
LeoCore			✓		
Sandblaster		✓			✓
ConnX BBE		✓			✓
EVP		✓			✓
SODA		✓			✓
ARM Ardbeg		✓			✓
Tomahawk		✓			✓
Infineon		✓			✓
Montium	✓				
BUTTER & CREMA	✓			✓	
HERS		✓			
ADRES	✓			✓	

Table 4 Power

Architecture	Technology	Details
LeoCore	0.12 µm CMOS	70 mW@70 MHz
Sandblaster	N/A	N/A
ConnX BBE	N/A	N/A
EVP	90 nm CMOS	1 mW/MHz
SODA	90 nm CMOS	450 mW@400 MHz
ARM Ardbeg		500 mw@350 MHz
Tomahawk	UMC 130 nm	940 mW@170 MHz
Infineon	65 nm CMOS	280 mW@300 MHz (DSP with 4 SIMD Cores)
Montium	0.13 µm CMOS	600 µW/MHz
BUTTER & CREMA	N/A	N/A
HERS	N/A	3.1 mW@250 MHz
ADRES	TSMC 90G	335 mW@400 MHz

Let us consider the CoreSonic's LeoCore. We may assume that no extra flexibility is needed to extend the fixed domain of functions required for radio baseband processing. Optimizing an architecture based on 90% code locality principle may help us to realize an SDR covering most of the used standards. As mentioned already no actual figures have been given for LTE test chip so far. Its optimized domain specific instruction set should lead to low program memory usage, low amount of memory accesses and low control overhead when compared to VLIW architectures. It can significantly reduce area and power consumption.

Sandbridge's efforts most importantly in addition to hardware are on the compiler sides to efficiently exploit the depth of parallelism inherent in DSP algorithms. However, compiler leaves how much room for further optimizations remains a question. The interesting aspect of such an approach is that all these standards are implemented in C language and no hardware accelerators are required. This is very efficient way when dealing with strict time constraints to launch a product in the market. However, when we think about the constraints for SDR such as power and area simply scaling up the DSPs to wider data paths and multiple cores seems not being truly a solution. Considering channel coding kernels such as FECare also very complex in their implementation. Running these algorithms without the support of any accelerators on scaled up DSPs might pose serious challenges of area and power.

Tensilica again adopts the same philosophy of exploiting data and instruction level parallelism using SIMD and 3-way VLIW architecture, respectively. Its distinct feature might be its multiple load/store units but most importantly their efforts in compiler design. Any C-code for DSP operations like FFT or filtering is restructured and vectorized to exploit the data and instruction level parallelism, inherent in DSP operations. This process might be automatic or manual.

NXP also opts for going toward SIMD architecture along with VLIW capabilities. Their division of whole receiver functions on the basis of needed flexibility in reconfiguration appeals to some extent. Because if somewhere in the system flexibility is not required so much and a less flexible hardware can be used across a range of different standards we can save power and area. However, as a whole architecture becomes less flexible which might become a serious drawback when it comes to SDR to be used over longer period of time.

SODA as well exploits the data level parallelism leveraging its SIMD architecture. One interesting thing they do is to avoid splitting DSP kernels into threads and instead schedule an entire kernel statically on a PE according to the algorithm data flow. This avoids heavy overhead of intra-kernel communication traffic. Another aspect which distinguishes SODA from other processor centered architectures is the use of scratchpad memory. It can be seen easily that most of the above platforms revolve around two planes: the control plane and data plane. DSP kernels are power hungry kernels and SIMD seems a natural choice for them. Considering error control algorithms, they do not hold the Data Level Parallelism as in DSP kernels like FFT. Using SIMD capable DSP might go for much power consumption in this domain of algorithms. Revisiting SODA architecture and adding a TURBO coprocessor leading to ARM Ardbeg is a step toward accepting the above mentioned fact.

Again Tomahawk and Infineon go for same SIMD approach assisted with some accelerators. In case of Tomahawk, the hardware unit named as CoreManager as mentioned earlier is its distinct feature. It can schedule at runtime 16 tasks in the pipeline and this scheduling load can be taken off from the compiler side. Tomahawk also realizes the different complex and computationally intensive nature of channel coding algorithms and deploys dedicated ASIPs instead of high performance general DSPs.

In fact, the main obstacle in using such complex and massively parallel DSP processors is the fact that the compilers available today cannot fully exploit the architecture and at the same time achieve efficient code for SDR. For this reason, today the applications running on the vector DSPs are still coded (or at least optimized) manually: applications can be written in C augmented with so called intrinsic, which are processor-specific complex instructions.

One of the biggest challenges in the future will be finding a good way of programming such complex machines, especially when considering a performance vs. portability trade-off. For instance, Sandbridge has a solid

programming model based on plain ANSI C and threads, thus offering real portability between systems supporting this programming model. On the other hand, C introduces artificial dependencies in the code which are not actually present in the algorithms described, thus such solution is not optimal under this respect. The other approaches discussed in this article are instead affected by lack of portability of the code between them: the DSP-based solutions are programmed via C extended with intrinsics, thus processor-specific assembly instructions. The CGRAs are instead programmed by means of architecture-specific bit-streams.

Considering CGRAs over other solutions their advantage can be their adaptability to algorithms with inherent data and instruction level parallelism to complex error correction algorithms. Their hardware adapts the algorithm and thus more closer to ASICs. Principally they must be efficient in terms of area and power. Their flexibility depends upon their grain. Some CGRAs may suffer from runtime reconfigurability as they are once optimized for a certain application they cannot be changed like ADRES and CREMA. The elements in these arrays might be basic arithmetic units or some abstraction at higher level or lower level. The programmer has to connect these elementary units in order to implement an algorithm like convolution, correlation, FFT or Viterbi algorithm. And at this point he needs to put real efforts in order to produce a final bit stream for an algorithm. We thus need a new programming model, open and shared between vendors, in order to guarantee total portability. Moreover, it should allow extracting all the parallelism contained in the applications, and be able to guarantee an easy and effective mapping on massively parallel MPSoCs. When mapping an application on such systems with numerous cores exploiting parallelism at all levels such as TLP (task level parallelism), ILP (instruction level parallelism). DLP (data level parallelism) and PLP (processor level parallelism) poses new challenges and questions that what could be atomized and what is at hands of the programmer. In order to facilitate programming huge systems probably industry needs to put more efforts in developing the tools and environments and may follow some standards to make the portability among different platforms feasible. In this case, an interesting study has been found in [38] about the mapping flow approaches followed by different research groups. The suggested mapping flow starts with inter-processor TLP toward intra-processor TLP which is basically parallelism among threads. Then comes the DLP and in the last it is ILP. Explorations for all these parallelism must be done carefully either by experienced programmer or by high level estimators [38,39]. More details about mapping flow on some individual platforms can be found in [38].

Multi-Processor System-on-Chip (MPSoC) seems one very promising and more radical evolution. In such a scenario, several clusters are connected via a NoC; such architecture is very interesting because it enables exploiting the application parallelism at all levels. MPSoCs can be either homogeneous or heterogeneous and can have centralized control or distributed control. Homogeneous MPSoCs exhibit high regularity due to the fact that they are obtained by replicating the same cluster, and can be seen as a large CGRA-like architecture. This approach has significant advantages from a silicon implementation point of view, but implies higher overhead in implementing heavy computation, thus requiring a very high amount of clusters when compared to heterogeneous MPSoCs. On the other hand, heterogeneous MPSoCs may pose more pressure on the interconnection network due to the necessity to move data to specialized computation engines across the system. When the control is centralized, a cluster acts as a 'master' of the system, coordinating the work of the rest of the system. This is a robust approach, but may pose issues from the scalability point of view. On the other hand, distributed control poses high challenges to the programmers about software development and verification. Heterogeneous solutions can provide sufficient flexibility with low-energy by mixing GP processors and carefully selected custom accelerators.

From cache point of view, general-purpose processors with caches are unpredictable and energy hungry due to the massive amount of speculation involved in both execution and data prefetching. This speculation is unnecessary in SDR and only increases with concurrency. When using deterministic prefetching strategies suitable for SDR, cache coherence is not a problem and segmented interconnects can be more readily exploited to achieve higher bandwidth of dataflow inside the chip. Snooping cache coherence protocols tend to be infeasible in segmented interconnects of which NoCs are one specific sub-group. However, this benefit of higher internal bandwidth might not be needed if the algorithm can be mapped so that thick datastreams are not needed between remote locations. Internal bandwith requirements might also be relaxed by low external data rate or chip I/O bottleneck. Thus, the benefit of segmented interconnects depends on the internal mapping of the algorithm and chip I/O capabilities among others.

Conclusion

This article provided an overview about current SDR platforms and solutions proposed by the academic world and by the industry. The SDR baseband solutions proposed so far represent just the first step toward a second generation of SDR platforms. Still, the current solutions are very important in a way that they show

SDR as viable approach, and already a reality under some constraints. Important challenges related to the need of huge processing power and to the need to limit energy consumption need to be solved before SDR can become a mainstream technology.

We can easily conclude that most of the solutions are DSP based. General purpose DSPs can give the highest level of flexibility but at some point they are impractical especially for hand held devices because of their huge area and power consumption. That is why industry realized that general purpose DSP based solutions must be assisted with accelerators or they must be extended with optimized instruction set. They can give enough flexibility for SDR with area and power savings. Reconfigurable arrays on the other hand can be very useful as function accelerators in SDRs. However, they can also serve to develop a whole SDR platform with multiple instantiations, each working differently connected to a network on chip.

Summarizing, we foresee that the short/mid-term future platform for SDR is a heterogeneous MPSoC with centralized control and up to tens of clusters. The master cluster may host an ASIP processor; some other clusters may host VLIW DSPs, or CGRAs, while others may host ASIC accelerators. In the mid/long-term future, the platform for SDR might be a distributed control MPSoC, either homogeneous or heterogeneous, with hundreds or even thousands of clusters, in order to support dynamically a series of demanding concurrent applications. There could be an entire set of reconfigurable machines ranging from fine grain custom FPGAs to CGRAs. Very importantly the obvious memory bottleneck in high-performance SDR computation platforms also needs to be tackled.

Abbreviations

CDL: Configuration Design Language; CDS: Coresonic developer studio; CGRA: Coarse Grain Reconfigurable Architecture; DSP: Digital Signal Processing; LIW: long instruction word; MMSE: minimum mean square error; MPSoC: Multi-Processor System-on-Chip; OFDM: orthogonal frequency division multiplexing; PEs: processing elements; QoS: quality of service; SDR: Software Defined Radio; SIMD: Single Instruction Multiple Data; SODA: signal-processing on-demand architecture.

Acknowledgements

This research was supported by Tampere University of Technology Finland & Graduate School of Electronics, Telecommunication & Automation (GETA) Finland.

Author details

[1]Department of Computer Systems, Tampere University of Technology, P. O. Box 553, Tampere, 33101, Finland [2]Nokia Research Center, Helsinki, 00180, Finland

Competing interests

The authors declare that they have no competing interests.

Received: 24 September 2010 Accepted: 6 June 2011
Published: 6 June 2011

References

1. JH Reed, *Software Defined Radio: A Modern Approach to Radio Engineering* (Prentice Hall, 2002, Upper Saddle River, New Jersey 07458)
2. J Mitola III, *Software Radio Architecture: Object Oriented Approaches to Wireless Systems Engineering* (John Wiley and Sons, 2000, Third Avenue, New York, NY 10158)
3. JAC Bingham, Multicarrier modulation for data transmission: an idea whose time has come. IEEE Commun Mag. **28**, 5–14 (1990)
4. M Alard, R Lassalle, Principles of modulation and channel coding for digital broadcasting for mobile receivers. Eur Broadcast Union Rev. **224**, 47–69 (1987)
5. F Berns, G Kreiselmaier, N Wehn, Channel decoder architecture for 3G mobile wireless terminals. in *IEEE Design, Automation and Test in Europe Conference and Exhibition*, vol. 3, 2004, pp. 192–197
6. F Kienle, N Wehn, H Meyr, On complexity, energy- and implementation-efficiency of channel decoders. in *IEEE, Transactions on Communications*, March 2010
7. T-D Chiueh, P-Y Tsai, *OFDM Baseband Receiver Design for Wireless Communications* (John Wiley and Sons (Asia), 2007, Clementi Loop, #02-01, Singapore 129809)
8. JW Cooley, JW Tukey, An algorithm for the machine computation of the complex Fourier series. Math Comput. **19**, 297–301 (1965)
9. J-S Park, B-K Kim, J-G Chung, KK Parhi, High-speed tunable fractional-delay allpass filter structure. in *Proceedings of 2003 IEEE International Symposium Circuits and Systems (ISCAS03)*, vol. 4, Bangkok, Thailand, May 2003, pp. 165–168
10. T-D Chiueh, P-Y Tsai, *OFDM Baseband Receiver Design for Wireless Communications* (John Wiley and Sons, 2007, Clementi Loop, #02-01, Singapore 129809)
11. M-H Hsieh, C-H Wei, Channel estimation for OFDM systems based on comb-type pilot arrangement in frequency selective fading channels. in *IEEE Transactions on Consumer Electronics*, vol. 44(1), pp. 217–225, February 1998
12. U Ramacher, Software-defined radio prospects for multistandard mobile phones. IEEE Comput. **40**(10), 62–69 (2007)
13. D Liu, A Nilsson, E Tell, D Wu, J Eilert, Bridging dream and reality: programmable baseband processors for software-defined radio. IEEE Commun Mag. **47**, 134–140 (2009)
14. D Liu, *Embedded DSP Processor Design, Application Specific Instruction Set Processors* (Morgan Kaufmann, 2008, Burlington, MA 01803, USA)
15. A Nilsson, E Tell, D Liu, An 11 mm^2, 70 mw fully programmable baseband processor for mobile WiMAX and DVB-T/H in 0.12 μm CMOS. Proceedings of ISSCC, San Francisco, CA, February 2008
16. M Moudgill, J Glossner, S Agrawal, G Nacer, The Sandblaster 2.0 Architecture and SB3500 Implementation. in Proceedings of the *Software Defined Radio Technical Forum (SDR Forum '08)*, Washington, DC, October 2008
17. J Glossner, D Iancu, M Moudgill, G Nacer, S Jinturkar, M Schulte, The sandbridge SB3011 SDR platform. in Proceedings of *Joint IST Workshop Mobile Future Symposium on Trends Communication (SympoTIC)*, June 2006, pp. ii–v
18. Z Tu, M Yu, D Iancu, M Moudgill, J Glossner, On the performance of 3GPP LTE baseband using SB3500. in *System-on-Chip, 2009. SOC 2009. International Symposium*, 5–7 October 2009, pp. 138–142
19. C Rowen, P Nuth, S Fiske, A DSP architecture optimized for wireless baseband. International Symposium on *System-on-Chip*, 5–7 October 2009, pp. 151–156
20. K Van Berkel, F Heinle, PE Patrick Meuwissen, Kees Moerman, Matthias Weiss, Vector processing as an enabler for software defined radio in handheld devices. EURASIP J Appl Signal Process. **2005**(16), 2613–2625 (2005)
21. K van Berkel, A Burchard, D van Kampen, P Kourzanov, O Moreira, A Piipponen, K Raiskila, S Slotte, M van Splunter, T Zetterman, Multi-radio scheduling and resource sharing on a software defined radio computing. in Proceedings of the *SDR 09 Technical Conference and Product Exposition*, 2009
22. JH Ahn, WJ Dally, B Khailany, UJ Kapasi, A Das, Evaluating the imagine stream architecture. in Proceedings of the *31st Annual International Symposium on Computer Architecture*, June 2004
23. PH Hofstee, All about the cell processor. in *IEEE Symposium on Low-Power and High-Speed Chips(COOL Chips VIII)*, April 2005.

24. Y Lin, H Lee, M Woh, Y Harel, S Mahlke, T Mudge, C Chakrabarti, K Flautner, Soda: a high-performance dsp architecture for software-defined radio. IEEE Micro **27**(1), 114–123 (2007)
25. M Woh Yuan Lin, S Seo, S Mahlke, T Mudge, C Chakrabarti, R Bruce, D Kershaw, A Reid, M Wilder, K Flautner, From SODA to scotch: the evolution of a wireless baseband processor. in Proceedings of *41st IEEE/ACM International Symposium on Microarchitecture (MICRO-41)*, Lake Como, Italy, 8–12 November 2008
26. G Cichon, P Robelly, H Seidel, E Matus, M Bronzel, G Fettweis, Synchronous transfer architecture (sta). in Proceedings of the *4th International Workshop on Systems, Architectures, Modeling, and Simulation (SAMOS'04)*, Samos, Greece, July 2004, pp. 126–130
27. GK Rauwerda, GJM Smit, CRW van Bentham, PM Heysters, Reconfigurable turbo/viterbi channel decoder in the coarse-grained montium architecture. in Proceedings of the *International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA'06)*, Las Vegas, Nevada, USA, June 2006
28. PM Heysters, GJM Smit, Mapping of DSP algorithms on the MONTIUM architecture. *Parallel and Distributed Processing Symposium, 2003. Proceedings International*, 22–26 April 2003
29. PM Heysters, Coarse-grained reconfigurable computing for power aware applications. in Proceedings of the *2006 International Conference on Engineering of Reconfigurable Systems Algorithms*, Las Vegas, Nevada, USA, June 26–29, 2006
30. C Brunelli, F Cinelli, J Nurmi, A vhdl model and implementation of a coarse-grain reconfigurable coprocessor for a risc core. in *Proceedings of the 2nd Conference on Ph.D. Research in MicroElectronics and Electronics (PRIME)*, pp. 229–232, June 2006
31. J Kylliäinen, T Ahonen, J Nurmi, General-purpose embedded processor cores—the COFFEE RISC example. in *Chapter 5, Processor Design: System-on-Chip Computing for ASICs and FPGAs*, edited by Nurmi J (Kluwer Academic Publishers/Springer Publishers, 2007), pp. 83–100
32. F Garzia, C Brunelli, C Giliberto, R Airoldi, J Nurmi, Implementation of W-CDMA cell search on a runtime reconfigurable coarse-grain array. in Proceedings of the *International Conference of Computer as a Tool (EUROCON '09)*, IEEE, 2009
33. F Garzia, R Airoldi, C Brunelli, C Giliberto, J Nurmi, Mapping of the FFT on a reconfigurable architecture targeted to SDR applications. in Proceedings of *International Symposium on System-on-Chip (SOC'09)*, IEEE, 2009, pp. 157–160
34. A Niktash, HT Parizi, N Bagherzadeh, Application of a heterogeneous reconfigurable architecture to OFDM wireless systems. in Proceedings of the *IEEE International Symposium on Circuits and Systems (ISCAS) 2007*, New Orleans, 27–30 May 2007, pp. 2586–2589
35. HT Parizi, A Niktash, AH Kamalizad, N Bagherzadeh, A reconfigurable architecture for wireless communication systems. in *Third International Conference on Information Technology: New Generations (ITNG 06)*, 2006, pp. 250–255
36. B Bougard, B De Sutter, S Rabou, D Novo, O Allam, S Dupont, L Van der Perre, A coarse-grained array based baseband processor for 100Mbps+ software defined radio. in Proceedings of the *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, Munich, Germany, 10–14 March 2008, pp. 716–721
37. B Mei, S Vernalde, D Verkest, H De Man, R Lauwereins, Exploiting loop-level parallelism on coarse-grained reconfigurable architectures using modulo scheduling. Comput Digital Tech, IEE Proc. **150**, 255–261 (2003)
38. M Palkovic, P Raghavan, M Li, A Dejonghe, L Van der Perre, F Catthoor, Multicore embedded systems for future SDR platforms: architecture and mapping flow overview. IEEE Signal Process Mag. **27**, 22–33 (2010)
39. Multicore Association, Multicore-Programming Practices Group (2009). http://www.multicore-association.org/workgroup/mpp.php
40. V Surducan, M Moudgill, G Nacer, E Surducan, P Balzola, J Glossner, S Stanley, M Yu, D Iancu, The sandblaster software-defined radio platform for mobile 4G wireless communications. Hindawi Publishing Corporation. Int J Digital Multimedia Broadcast. **2009**, Article ID 384507, 9pp
41. Tensilica: http://www.tensilica.com/uploads/pdf/connx_bbe.pdf Accessed date on 31.12.2010 and 31.05.2011
42. Ö Paker, K van Berkel, K Moerman, Hardware and software implementations of an MMSE equalizer for MIMO-OFDM based WLAN. in *IEEE Workshop on Signal Processing Systems Design and Implementation*, November 2005, pp. 1–6
43. T Limberg, M Winter, M Bimberg, MBS Tavares, H Ahlendorf, MG Fettweis, H Eisenreich, G Ellguth, A heterogeneous mpsoc with hardware supported dynamic task scheduling for software defined radio. In *Design Automation Conference 2009 (DAC'09)*.
44. CH van Berkel, Multi-core for mobile phones. in Proceedings of *DATE '09. Design, Automation. Test in Europe Conference. Exhibition*, 20–24 April 2009, pp.1260–1265

doi:10.1186/1687-1499-2011-5
Cite this article as: Anjum *et al.*: State of the art baseband DSP platforms for Software Defined Radio: A survey. *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:5.