Microprocessors and Microsystems 33 (2009) 191–200

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

# Low-complexity FFT/IFFT IP hardware macrocells for OFDM and MIMO–OFDM CMOS transceivers

Sergio Saponara, Nicola E. L'Insalata, Luca Fanucci *

*Dept. of Information Engineering, University of Pisa, Via G. Caruso 16, I-56122 Pisa, Italy*

ARTICLE INFO

*Article history:*
Available online 16 December 2008

*Keywords:*
Application specific processors
Automatic generation of IP cores
OFDM communications
MIMO systems
FFT
CMOS VLSI design

ABSTRACT

The paper presents an automated environment for fast design space exploration and automatic generation of FFT/IFFT macrocells with minimum circuit and memory complexity within the numerical accuracy budget of the target application. The effectiveness of the tool is demonstrated through FPGA and CMOS implementations (90 nm, 65 nm and 45 nm technologies) of the baseband processing in embedded OFDM transceivers. Compared with state-of-art FFT/IFFT IP cores, the proposed work provides macrocells with lower circuit complexity while keeping the same system performance (throughput, transform size and accuracy) and is the first addressing the requirements of all OFDM standards including MIMO systems: 802.11 WLAN, 802.16 WMAN, Digital Audio and Video Broadcasting in terrestrial, handheld and hybrid satellite-scenarios, Ultra Wide Band, Broadband on Power Lines, xDSL.



## 1. Introduction

### 1.1. Automated generation of FFT/IFFT cores

To manage the high design complexity of Systems-on-Chip (SoC) a divide-and-conquer approach is typically adopted assembling pre-designed and pre-verified Intellectual Property (IP) hardware cores. This way design productivity is increased reducing time to market and saving design costs. The Fast Fourier Transform (FFT) and its inverse (IFFT) is the best candidate to be implemented as reusable IP macrocells since they are the most widespread operators for digital signal processing [1]. In most SoC designs the hardware/software partitioning maps FFT and IFFT tasks onto hardware macros. Indeed a software implementation of FFT/IFFT, although possible in real-time up to hundreds of MS/s with state-of-art DSPs [2,3], is power hungry. This is particularly critical for battery-powered applications, such as baseband processing of wireless transceivers, and for computation-intensive embedded signal processing systems where high power dissipation leads to heat removal and reliability issues.

At the state of the art several FFT/IFFT IP cores have been proposed [4–17] as parametric architectures customizable at synthesis time. Typical parameters are the FFT length and the bit-width of the I/O, the transform coefficients and the internal data path. However the above works miss the support of a design environment to find the optimal IP configuration for the given application. The IP core integrator is thus exposed to a wide design space involving multiple performances (transform length, accuracy, latency and throughput) and cost (area, leakage and dynamic power consumption) metrics. The design space is further enlarged since CMOS technologies offer standard-cells with multiple threshold voltages (multi-$V_t$), hence different trade-offs between power consumption and timing is possible. Finding, in such wide design space, the IP configuration with optimal trade-off among multiple cost functions is time consuming; this reduces the benefit of the IP design reuse approach.

To overcome the above issues in this paper we propose a FFT/IFFT scalable architecture template coupled with an automated design environment which allows at system level (i) fast design space exploration and (ii) IP macrocell configuration and database generation. As case studies to demonstrate the effectiveness of the design environment this paper presents the implementation, in CMOS and FPGA technologies, of several communication transceivers based on the multi-band orthogonal frequency division multiplexing (OFDM) technique. Multi input multi output MIMO–OFDM schemes are also considered. The performance and reusability of the generated FFT/IFFT IPs are assessed through logic synthesis results on FPGA devices, Virtex and Spartan families from Xilinx, and on CMOS standard-cells libraries from STMicroelectronics: 90 nm, 65 nm and 45 nm multi-$V_t$ CMOS with a supply voltage of 1.2 V, 1.1 V and 1 V, respectively.

### 1.2. State-of-art review

Other works in the state of the art targeted the problem of automatic generation of FFT/IFFT cores, particularly for embedded

* Corresponding author. Tel.: +390502217688; fax: +390502217522.
*E-mail addresses:* s.saponara@iet.unipi.it (S. Saponara), n.linsalata@iet.unipi.it (N.E. L'Insalata), l.fanucci@iet.unipi.it (L. Fanucci).

0141-9331/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.micpro.2008.12.001

OFDM systems [6–8,15,18–22]. However none of the above macrocells covers all the requirements coming from the analysis of the different OFDM standards. Particularly the implementation of MIMO schemes, used in Wireless LAN and MAN transceivers and of Broadband on Power Line (BPL) modems are not addressed. Most works [6–8,15,18,21,22] do not meet the high-throughput requirements of OFDM schemes such as Ultra Wide Band (UWB) [23]. Moreover several FFT/IFFT IP core generators in literature are based on fixed-point arithmetic [6–8,15,21,22]; other techniques exist, such as block floating point (BFP) and convergent block floating point (CBFP), which offer better trade-offs between circuit complexity and processing accuracy vs. conventional fixed- and floating point. Some works are not technology independent, e.g. [15,18,21] are specific designs for particular FPGA families and most of them, although presenting a characterization of the generated FFT/IFFT IPs on CMOS technologies, do not consider the impact of multi-$V_t$ libraries.

All the above state-of-art limitations are addressed in this work which is organized as follows: Section 2 briefly analyzes the baseband architecture of OFDM and MIMO–OFDM transceivers and the relevant computing requirements. Section 3 presents a configurable FFT/IFFT IP core and details some architectural solutions adopted to meet the required functional specifications while optimizing circuit complexity in terms of area, memory resources and power consumption. The design of the processor arithmetic is discussed in Section 4 which also shows, for the different OFDM standards, the optimal IP core configuration and database generation. The characterization of the generated FFT/IFFT processors in 45 nm, 65 nm and 90 nm multi-$V_t$ CMOS technologies is presented in Section 5, together with a comparison with the state of the art and a discussion on CMOS scaling trends. Section 6 presents the IP cores characterization in FPGA devices and a comparison of the achieved results with standard-cells CMOS ones and with state-of-art FPGA implementations. Conclusions are drawn in Section 7.

## 2. Baseband processing OFDM systems

### 2.1. OFDM architectures

The multi-carrier OFDM scheme has fostered the rise of several wireless and wired communication services. The broadcast delivery of video and audio contents is covered by Digital Video Broadcasting standards, in terrestrial (DVB–T), handheld (DVB–H) and satellite-hybrid (DVB–SH) scenarios [24], and by Digital Audio Broadcasting (DAB) [5]; wireless fast internet access in metropolitan scenarios is covered by 802.16-d and -e Wireless MAN (WMAN) standards, known, respectively, as fixed and mobile WiMAX [25]; wired fast internet access through the telephone line and the power line are enabled by Digital Subscriber Line (xDSL) [26,27] and BPL [28,29] technologies, respectively; 802.11-a and -n Wireless LAN (WLAN) deal with medium range indoor networking [30,31] while ECMA-368 UWB radio [23] is conceived for short range high data rate connectivity. Supported data rates amount to several Mbps, growing to hundreds of Mbps for MIMO WLAN and UWB.

With respect to single-carrier modulation OFDM offers enhanced robustness against cross-talk, fading channels and multipath effects [32]. In 802.16e, OFDM is also deployed as a multiuser access technology (OFDMA) where the carriers are clustered in subsets dynamically assigned to each user. All the aforementioned standards exploit a similar baseband processing scheme based on a FFT processor, in charge of multi-carrier symbol demodulation at the receiver (RX), plus an IFFT processor in charge of symbol modulation at the transmitter (TX). FFT and IFFT require roughly half of the total circuit complexity of the baseband processing in OFDM systems [5,33]. Fig. 1 details the baseband processing architecture for a WLAN transceiver [30].

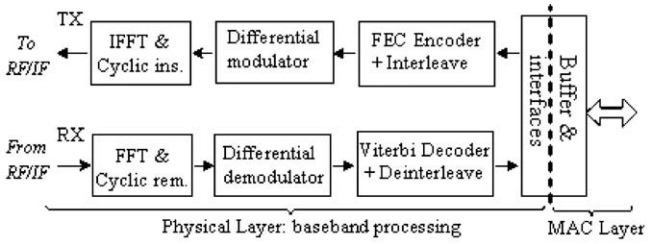


**Fig. 1.** WLAN OFDM architecture.

### 2.2. MIMO OFDM architecture

In 802.16 WMAN and 802.11n WLAN standards OFDM is used in conjunction with the MIMO technique which adopts multiple antennas at both RX and TX sides to exploit the spatial domain for spatial diversity and/or spatial multiplexing [14,33,34], see Fig. 2. Spatial multiplexing increases the capacity of a MIMO link by transmitting independent data streams in the same time slot and frequency band simultaneously from each TX antenna. Multiple data steams at the RX side are differentiated using channel information about each propagation path. As example the $2 \times 2$ MIMO WLAN in [34] and the $4 \times 4$ MIMO scheme in [33] achieve data rates of 120 and 192 Mbps, respectively. The gain vs. the 50–60 Mbps of a single input single output (SISO) WLAN is roughly a factor 2 for the $2 \times 2$ MIMO and 3.5 for the $4 \times 4$ MIMO scheme. In contrast to spatial multiplexing, spatial diversity increases the diversity order of a MIMO link to mitigate fading by coding a signal across space and time using special space-time code techniques such as the Alamouti code [34]. At the RX side the signal replicas are combined constructively to achieve a diversity gain. Fig. 3 shows the baseband processing architecture for a $2 \times 2$ WLAN OFDM systems [34]. Comparing the schemes of Fig. 3 and 1 it is clear that the number of FFT and IFFT processors to be integrated in the transceiver depends on the number of RX and TX paths. A MxM OFDM–MIMO system requires M FFT (and M IFFT) processors working in parallel thus increasing area by a factor M. Alternatively, a lower number of $P$ processors, with $P \in [1,M]$, can be used but the data throughput, and hence the clock frequency of the $P$ processors, should be increased by a factor M/P. Typical values for $M$ are 2 or 4.

In SISO and MIMO OFDM schemes, adopting time division duplexing, FFT and IFFT tasks can be merged in a single FFT/IFFT processor since the transceiver is working either in RX mode, demodulation by FFT, or in TX mode, modulation by IFFT. In transceivers adopting frequency division duplexing, featuring

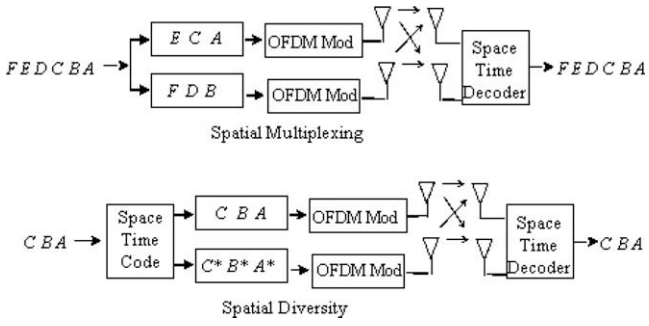


**Fig. 2.** Spatial diversity and spatial multiplexing in MIMO systems.

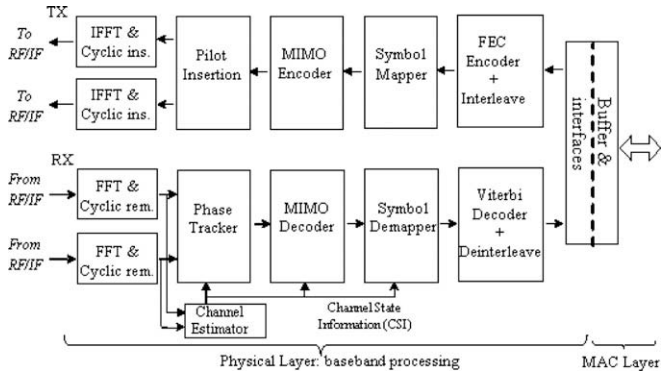


**Fig. 3.** WLAN 2 × 2 MIMO–OFDM architecture.

concurrent TX and RX paths, different processors are needed for FFT and IFFT.

### 2.3. OFDM and MIMO–OFDM processing requirements

The different OFDM standards considered in the paper (UWB, WLAN 802.11-a/n, WMAN 802.16-d/e, DVB–T/H/SH, DAB, xDSL, BPL) are characterized by different requirements for the FFT/IFFT processing in terms of I/O bit-width, up to 16 bits, transform length, up to 8192 points, and throughput, up to 528 MS/s. These values, summarized in Table 1 and Fig. 4, are derived from standards specifications and/or system level analysis in literature [5,24–31,35]. With respect to [20] this work includes the requirements of new systems: WLAN and WMAN MIMO applications, BPL modems and DVB–SH transceivers. The latter, using FFTs sized from 1024 to 8192 with throughput up to 8 MS/s, enable DVB services over hybrid satellite-terrestrial channels [36]. DVB services on pure satellite channels, covered as example by the DVB–S2 standard, adopt single-carrier schemes instead of OFDM.

**Table 1**
FFT requirements of OFDM and MxM MIMO OFDM standards.

| Standard | FFT length | I/O size, bits | Throughput, MS/s |
|---|---|---|---|
| DVB–T/H/SH | 1024–8192 | 8 | 9 |
| DAB | 256–4096 | 8 | 8.26 |
| xDSL | 256–4096 | 16 | 1–35 |
| 802.11a | 64 | 8 | 20 |
| 802.11n | 128 | 8 | 40 (xM) |
| 802.16-d/e | 128–2048 | 10 | 1.25–20 (xM) |
| UWB | 128 | 6 | 528 |
| BPL | 512,1024 | 16 | 30 |

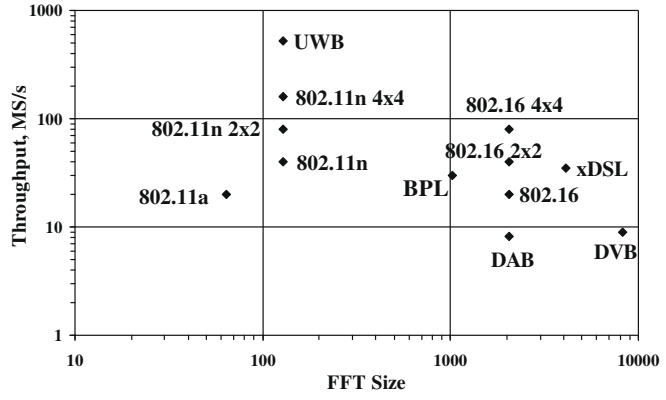


**Fig. 4.** FFT throughput/size for OFDM and MIMO–OFDM transceivers.

For standards with multiple modes, the one with highest size and throughput is reported in Fig. 4. For MxM MIMO–OFDM systems the throughput requirements are calculated considering the worst case of single FFT and IFFT processors, i.e. $P = 1$, running $M$ times faster than the basic SISO scheme. As example a 2 × 2 MIMO 802.11n scheme requires 1 FFT processor and 1 IFFT processor running at 80 MS/s instead of 40 MS/s. Alternatively $P = 2$ parallel processors for FFT and IFFT can be used running at the basic clock frequency but doubling the circuit area.

## 3. Low-complexity FFT/IFFT architectural template

This section briefly presents the pipelined mixed radix cascade architecture, on top of which the design environment for FFT/IFFT IP generation is built. Radix factorization is suitable for FFT/IFFT hardware implementation because of its high regularity and reduced number of operations. Given a radix factorization and the corresponding data-flow graph, the hardware architecture is defined by the degree of parallelism of the butterfly units. The IP architectural template, designed at register transfer level (RTL) as synthesizable VHDL, is based on a pipeline cascade of radix-4 (R4) butterfly stages. The pipeline cascade [4] allows for high length flexibility and is designed so that the whole processor can sustain a throughput of one complex sample per clock cycle. The use of R4 stages allows for higher output precision and a better trade-off between processing speed and area overhead vs. radix-2 (R2), as proved in [21,37]. As discussed in [38] higher radix factorizations (i.e. 8, 16 and 32) would require a butterfly with non-trivial multiplications, thus bearing an unacceptable increase of hardware complexity vs. R4. To support any power-of-two transform length, instead of power-of-four only, the last stage of the architectural template is a mixed R4/R2.

Fig. 5 illustrates the top-level architecture which features all the parameters reported in Table 2. Fig. 5 includes the data path of the FFT/IFFT cascade architecture and a wrapper between the local custom bus interface and the main system bus. In Fig. 5 the multiplexers at each input stage enable the FFT/IFFT length to be

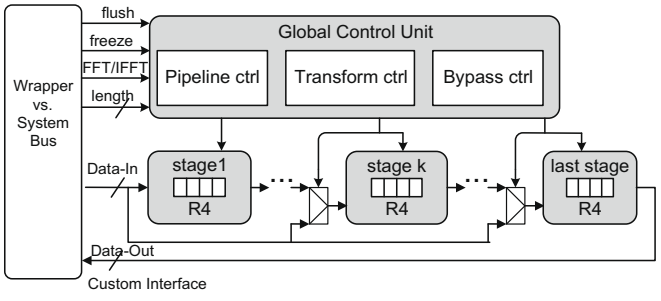


**Fig. 5.** Architecture top view.

**Table 2**
Architecture parameters.

| Symbol | Parameter | Configurable range (OFDM case studies) |
|---|---|---|
| $N_{max}$ | Maximum transform length | [64,8192] |
| $S$ | Total number of stages | [3,7] |
| $k$ | Number of R4 stages | [3,6] |
| $m$ | Number of R2 stages | [0,1] |
| IOWL | I/O bit-width | [6,16] |
| TWL | Twiddle coefficients bit-width | <32 |
| SWL | Internal data path bit-width | <32 |
| Wrap | Wrapper type | 0: custom bus, 1: AXI, 2: AHB, 3: STbus |

configured at run-time by selecting the number of stages in the cascade through the *length* signal. The *flush* and *freeze* signals control the internal pipeline, which can be frozen or flushed to match the input traffic rate or in case of run-time re-configuration of FFT/IFFT mode and/or transform length. The FFT/IFFT signal is used to switch between FFT and IFFT computation. Fig. 6 sketches a generic R4 stage which includes: (i) the butterfly/multiplier unit working on complex data; (ii) a data sequencing unit which uses small RAM banks for data reordering according to the algorithm control flow in [4] (thanks to these RAM banks input buffers can be removed since buffering capabilities are embedded in the data path); (iii) a ROM unit containing pre-computed twiddle factors. Exploiting the symmetry of the unit circle in the complex plane, all the twiddle factors can be generated starting from a kernel ROM reduced by a factor of 8 plus a simple circuit to exchanges the real and imaginary parts of the coefficients and/or to complement their signs. Eqs. (1) and (2), expressed as functions of the architecture parameters in Table 2, summarize the RAM and ROM size in bits for the $j$-th stage in the cascade.

The proposed FFT/IFFT cores can be integrated in complex SoCs where a high speed bus is typically used to connect high-rate IPs. To easy such integration the VHDL architecture template allows the selection of the interface towards the main system bus in Fig. 5, beside the native custom interface, a wrapper can be used to ensure compliancy with STbus, widely used in SoC by STMicroelectronics, or AXI and AHB AMBA (Advanced Microcontroller Bus Architecture) specifications which are de-facto standards in embedded systems [53]. The type of system bus interface can be selected through the wrap parameters, see Table 2, before IP synthesis. The hardware overhead of the wrappers amounts to few K gates. To easy the comparison with state-of-art FFT/IFFT IPs in the rest of the paper the synthesis results will refer to the native custom bus without extra wrapper.

$$\mathrm{RAM}\ (j) = \begin{cases} 7 \cdot 2\ \mathrm{IOWL} \cdot \frac{N_{\max}}{4 \cdot 2^m} & j = 1 \\ 7 \cdot 2\ \mathrm{SWL} \cdot \frac{N_{\max}}{4^j} & j \in [2, S-1] \\ 2\ \mathrm{SWL} & j = S \end{cases} \quad (1)$$

$$\mathrm{ROM}\ (j) = 2\ \mathrm{TWL} \cdot \frac{N_{\max}}{8 \cdot 4^{j-1}} \quad j \in [1, S-1] \quad (2)$$

## 4. Processor arithmetic design, IP core configuration and database generation

### 4.1. Arithmetic design

Given the specifications of a target application the IP configuration requires a trade-off between the numerical accuracy, depending on IOWL, TWL and SWL, and the hardware overhead associated to large parameter sizing (note that SWL affects the size of adders, multipliers, switches and RAMs in the radix stages while TWL sets the coefficients size in ROMs and multipliers). To this aim our configurable IP core supports different arithmetic types: fixed-point, block floating point (BFP) and convergent BFP (CBFP). Floating point arithmetic is not supported since it implies large area and power consumption overheads when implementing large transform sizes [37,39] such those of the target applications in Table 1.

Fixed point arithmetic does not require any kind of control logic thus simplifying the data path circuit design. As drawback, to avoid overflows during FFT/IFFT computation, the SWL value must grow by 2 bits for each R4 stage and one bit for the R2 stage. Considering the xDSL requirements as example, in a 4096 samples transform with 16 bits inputs the SWL should vary from 18 bits (first stage output) to 28 bits (last stage) to preserve a 16 bits output data accuracy. Instead of keeping IOWL at the minimum required and increasing SWL to avoid overflow, some fixed-point IP cores in literature

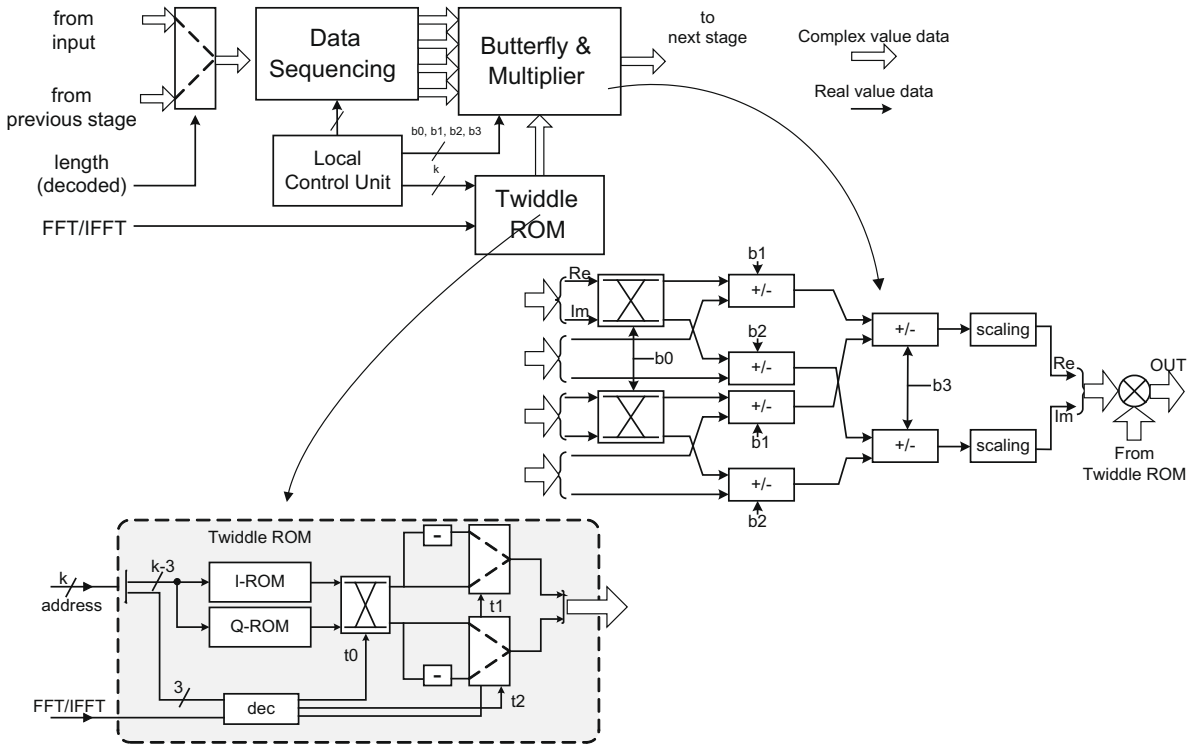


**Fig. 6.** Architecture of the R4 stage.

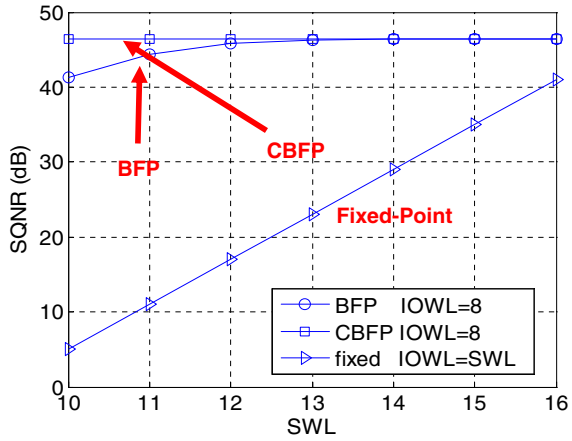


**Fig. 7.** SQNR vs. SWL, unquantized TWL. DVB–T 8192 mode. Fixed point performances are taken from [8].

[8,15] set IOWL = SWL = constant and pick up an intermediate value between the two extremes. Differently from fixed-point, BFP and CBFP associate an exponent to the data so that a single SWL is used for the whole architecture. In BFP the output data at each stage are scaled down through a right shift of 2 bits in R4 stages and 1 bit in the R2 stage.

In BFP the exponent field is implicit, and hence does not require extra control logic in the data path, because the number of total shifts is known a-priori and is related to the number of stages in the architecture. CBFP arithmetic implements adaptive scaling of data depending on their amplitude; the latter is estimated adding in the data path the low-complex magnitude estimator we originally proposed in [40]. To keep track of scaling operations CBFP extend data representation with an exponent field. By scaling data representation BFP and CBFP can achieve the same performance of fixed-point arithmetic but with much lower SWL and TWL values and hence lower circuit area, power consumption and cost. As example Fig. 7 compares, for the DVB–T 8192 mode the performance of BFP, CBFP and fixed-point arithmetic as a function of SWL when considering unquantized twiddle factors. It is clear from Fig. 7 that BFP and CBFP approaches outperform fixed-point one even with small SWL values. Due to adaptive scaling CBFP offers higher precision than BFP but requires extra hardware for the magnitude number estimation.

### 4.2. Automatic arithmetic configuration

A custom software tool, based on Monte Carlo simulations, has been developed to profile the different arithmetic types in terms of signal to quantization noise ratio (SQNR) or mean square error (MSE). Both open and closed-loop analysis are available. Open loop analysis allows measuring the performance of a specific arithmetic configuration. In this phase the tool can generate input test stimuli of different length according to deterministic (e.g. sinusoid, square-wave, ramp..,) or random (with uniform or Gaussian statistics) laws or a combination of them. For random stimuli, sequences with a desired peak to average ratio (PAR) can be generated too. In closed-loop analysis, see Fig. 8, the IP core integrator specifies the FFT/IFFT length, N, and a SQNR budget, i.e. the desired bit-true IOWL. The tool, using uniformly distributed random inputs (in a dynamic range depending on IOWL), evaluates the optimal values of SWL and TWL to achieve the desired output precision minimizing data path and memory sizes.

As golden reference model an FFT/IFFT 64 bits floating point processor is considered. The SWL and TWL configuration process

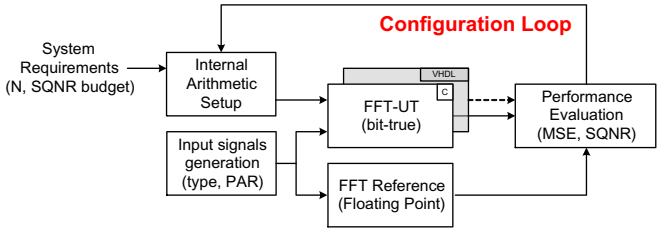


**Fig. 8.** Closed-loop arithmetic process sizing.

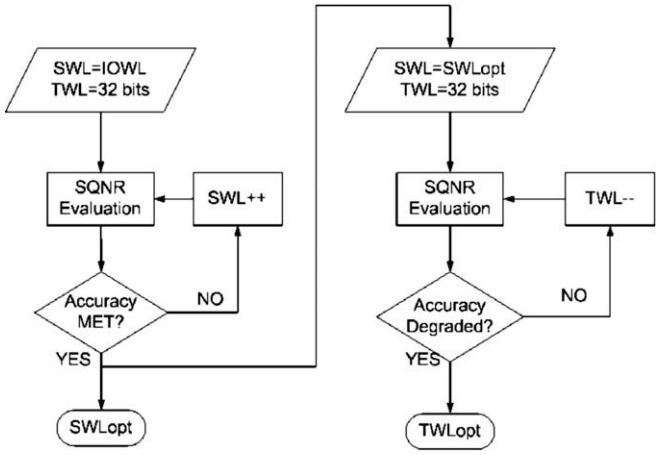


**Fig. 9.** SWL and TWL parameters setup.

is illustrated in Fig. 9. It is repeated for all the three arithmetic types. In Fig. 9 the SWL is determined first, as it affects the processor complexity (adders, multipliers, RAM) much more than TWL. Among the three arithmetic types the one which allows for the minimal SWL value is selected. In case of similar SWL value, between BFP and CBFP the first is preferred to avoid the hardware overhead of complex number magnitude estimation. The speed of the automatic configuration process mainly depends on the transform size N (fixed by the target application) and on the number of iterations used to evaluate the SQNR. The latter also determines the accuracy of the configuration process; hence a trade-off has to be found between speed and accuracy when configuring SWL and TWL. From experimental analysis we have verified that an optimal sizing of SWL and TWL is obtained using a number of iterations in the order of one thousand. A higher number of iterations are not useful since it increases the analysis time but leads to the same choice for the word lengths. To speed-up the exploration and IP configuration phase a bit-true C++ model has been associated to the VHDL template of Section 3. The C++ model structure matches the hardware architecture including the configurable parameters of each unit and the finite arithmetic effects. To assess the equivalence of the C++ bit-true and VHDL models in terms of IP configuration results, during the prototyping phase of the tool, we used both models in the loop of Fig. 8. Once validated, only the C++ model is used since its simulation time is one order of magnitude faster than the VHDL description. As example, the IP configuration time using a PC with an Intel Core2 CPU at 1.66 GHz, 1 GB RAM, Window XP-SP2 operating system, 50% CPU usage, is roughly 4 min with the C++ bit-true model for IOWL = 16 and N = 256; the time is reduced to 1 min when N = 64 and grows up to 1 h for N = 4096. After selecting the optimal architecture configuration the tool generates the IP macrocell database including the RTL VHDL code, testbenches and test vectors for verification and performance estimation.

**Table 3**
IP core closed-loop configuration for different OFDM standards.

| Standard | SWL | TWL | Type | Stages | SQNR (dB) |
|---|---|---|---|---|---|
| DVB–T/H/SH | 11 | 3 | BFP | 7 | 43.6 |
| DAB | 11 | 5 | BFP | 6 | 43 |
| xDSL | 18 | 12 | CBFP | 6 | 94.2 |
| 802.11-a/n | 11 | 6 | BFP | 4 | 44.7 |
| 802.16-d/e | 13 | 7 | BFP | 6 | 54.9 |
| UWB | 8 | 4 | BFP | 4 | 29.3 |
| BPL | 18 | 10 | CBFP | 5 | 93.7 |

### 4.3. OFDM FFT/IFFT IP core configuration

Applying the closed-loop flow to the OFDM standards in Table 1 and Fig. 4 the configurations of the FFT/IFFT IP cores in Table 3 are obtained (the IOWL and the length N are reported in Table 1). Analyzing the results it emerges that the CBFP arithmetic is suitable for large transform sizes and high IOWL accuracy, as in the case of xDSL or BPL requiring $N_{max} \geqslant 1024$ and SQNR $\geqslant$ 80 dB. As example for BPL modem, BFP arithmetic would require SWL = 20 to achieve the same SQNR that CBFP arithmetic achieves with SWL = 18. For other OFDM standards (DVB–T/H/SH, DAB, WLAN, WMAN, UWB) the required SQNR budget is such that BFP and CBFP reach the same performance with similar SWL and TWL values, thus BFP arithmetic is preferred to save the extra circuit complexity of the magnitude estimation unit.

As discussed in Section 2.2 adopting a MxM MIMO scheme, such as WLAN 802.11n or WMAN 802.16, requires, independently from the SQNR budget, the integration of $P$ processors running at a clock frequency M/P times faster than the basic SISO scheme. As a consequence the sizing of the arithmetic type and of the word lengths is the same for a given standard both in MIMO ($M = 2, 4$) or SISO ($M = 1$) configurations.

## 5. Sub-micron CMOS implementation results

### 5.1. Multi-$V_t$ CMOS technologies

This section presents the synthesis and characterization results of the IP cores configured as reported in Table 3. Three deep submicron technologies are considered: 90 nm, 65 nm and 45 nm CMOS libraries featuring multi-$V_t$ standard-cells, 7 metal layers, and scaled supply voltage $V_{DD}$ of 1.2 V, 1.1 V and 1 V, respectively. Synthesis and characterization activities are carried out with Synopsys Design Compiler CAD tool. RTL and gate-level simulations are carried out within the NC-Sim tool from Cadence.

The supply voltage scaling in submicron CMOS technologies is required to bound the dynamic power consumption which depends quadratically on $V_{DD}$. Also leakage power consumption, $P_{leak}$, has a non negligible contribution in today's SoC; $P_{leak}$ determines the stand-by energy cost of the device and this can be critical in battery-powered terminals. To address the increase of leakage power, submicron CMOS technologies provide standard-cells libraries with multiple threshold voltages ($V_t$). Indeed the higher $V_t$ the lower the leakage current $I_{leak}$ according to an exponential law, see Eq. (3) where SW is the sub-threshold swing and $I_o$ is a current proportional to the transistors width. On the other side, the higher $V_t$, the lower the $V_{DD}$–$V_t$ difference for a given supply voltage $V_{DD}$, hence the higher the time propagation delay (TPD) according to Eq. (4), [41,54].

$$P_{leak} = V_{DD} \cdot I_{leak} \approx V_{DD} \cdot I_0 \cdot 10^{-V_t/SW} \quad (3)$$

$$\mathrm{TPD} \propto \frac{V_{DD}}{(V_{DD} - V_t)^2} \quad (4)$$

Since in synchronous designs the maximum achievable frequency is determined by the time delay of the worst path, than synthesis tool can minimize leakage power consumption by substituting low-$V_t$ cells with high-$V_t$ ones (slower but with lower leakage) in non-critical paths, i.e. paths with positive timing slack with respect to the synthesis time delay constraint. The CMOS technologies used in this work provide cells with three $V_t$ values: high (HVT), standard (SVT) and low (LVT). In the different technology nodes the leakage power associated to HVT cells is about 1 order of magnitude smaller than SVT cells and two orders of magnitude smaller than LVT [42]. Due to the high leakage cost of the LVT cells their use is not suitable for battery-powered applications. Moreover as discussed in next Sections the speed performance of HVT and SVT cells is enough to sustain the throughput required by our target OFDM processing case studies; therefore a mix of HVT and SVT cells will be exploited. In terms of dynamic power HVT cells allow for a reduction of about 10–20% if compared with SVT cells.

### 5.2. Synthesis and characterization results

Fig. 10 shows the storage requirements, RAMs and ROMs, of the generated FFT/IFFT IP cores. For standards with multiple modes, the one with highest size is reported; for sake of clarity the WLAN label is used for 802.11-a/n, WMAN for 802.16-d/e and DVB for DVB–T/H/SH. RAM and ROM requirements in Fig. 10 are derived from Eqs. (1) and (2). The actual implementation of RAM banks, through SRAM macros, register files or Flip-flops depends on the actual size of each bank. SRAM macros are used for RAM banks above 8 K bits (e.g. first stage of DVB or xDSL FFT). Register files are chosen for RAM banks with size ranging from 1 to 8 K bits (e.g. third stage of DVB FFT). A flip-flop based implementation is used for RAM banks smaller than 1 K bits (e.g. WLAN and UWB memory banks, last stages of xDSL and DVB FFT cores).

Fig. 11 shows area and real-time clock frequency results for 45 nm 1 V CMOS standard-cells (underlined labels), for 65 nm 1.1 V CMOS standard-cells (strikethrough labels) and for 90 nm 1.2 V CMOS standard-cells (**bold** labels). The worst case operating condition of T = 125 °C is considered. The real-time clock frequencies are derived from throughput requirements, expressed as MS/s in Table 1, and considering that the pipelined cascade can process 1 sample per clock cycle.

As a consequence of the high architectural throughput for all considered technology libraries all clock frequency constraints can be met without resorting to the fast LVT cells, which suffer for high static power consumption. Clock frequencies for the DVB–T/H/SH, DAB, xDSL, BPL, WMAN and WLAN FFT/IFFT cores are below 50 MHz, thus only HVT cells are used. The only application requiring

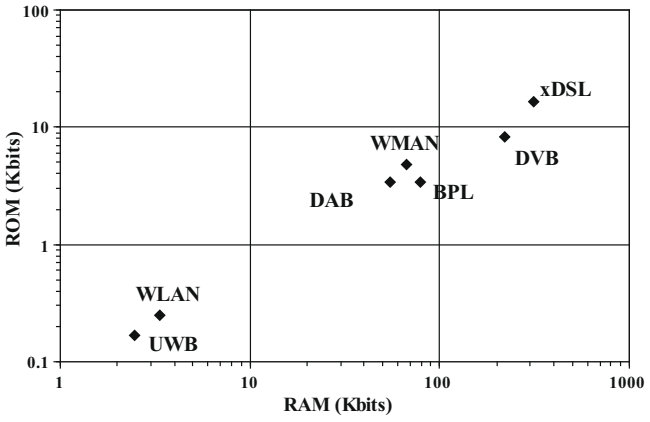


**Fig. 10.** ROM and RAM complexity for the synthesized IP cells.

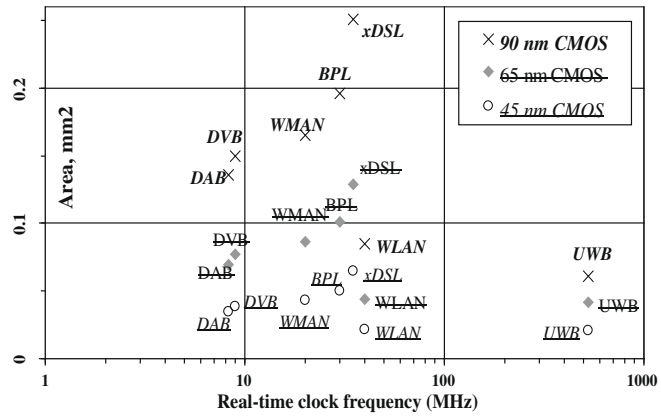


**Fig. 11.** Area and clock frequency results for the FFT/IFFT IP cores.

a high clock frequency, one order of magnitude higher than the other standards, is UWB; to meet the required 528 MHz a mix of HVT and SVT cells is required.

With reference to the above OFDM schemes and CMOS technologies, Fig. 12 sketches the leakage power consumption (when the transceiver is idle) and the total power consumption (when the processor is working); the latter includes both leakage and dynamic power contributions. In all technology nodes the highest silicon area is required by xDSL due to both high transform size, up to 4096, and processing accuracy, IOWL up to 16 bits, specifications. The highest power cost is required by UWB which, compared to other standards, has small transform size and accuracy specifications but requires a throughput one order of magnitude higher.

The results in Figs. 10–12 refer to SISO OFDM transceivers. Implementation results of MIMO OFDM schemes are discussed in Section 5.3.

Comparing the results of 90 nm, 65 nm and 45 nm CMOS standard-cells provides experimental evidence of scaling trends in deep submicron technologies. The area occupation can be reduced by a factor between 1.5 and 2 if scaling the same VHDL architecture from 90 nm to 65 nm CMOS technologies, or from 65 nm to 45 nm CMOS ones. In the same conditions the total power consumption can be reduced by a factor between 2 and 3. As far as circuit speed is concerned the gain due to technology scaling is lower. As example using a low-leakage (HVT cells only) 65 nm library an FFT/IFFT processor, in the different configurations, can reach a maximum clock frequency of about 250–300 MHz. Using a low-leakage 45 nm library the speed gain vs. 65 nm amounts to roughly 20%.

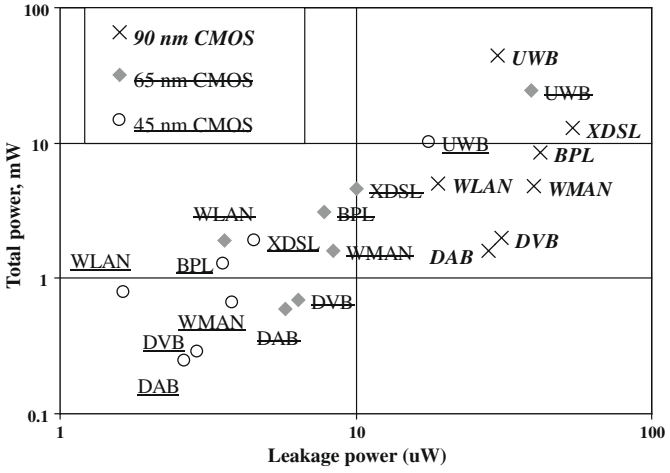


**Fig. 12.** Power consumption results for the FFT/IFFT IP cores.

For the OFDM case studies all the considered technology nodes allow for a real-time implementation. Particularly, in the 45 nm CMOS node all the FFT/IFFT processors can be implemented with power consumption of few mWs and an area below 0.1 mm$^2$.

It is worth noting in Fig. 12 that, for UWB, the implementation in 90 nm CMOS has a lower leakage power than the implementation in 65 nm CMOS. The reason is that for UWB a mix of SVT and HVT cells is adopted; while at 90 nm less than 10% of SVT cells are required to meet the clock frequency constraint, in the 65 node the use of SVT cells is increased up to 57% resulting in higher leakage power cost.

### 5.3. MIMO OFDM synthesis results

Table 4 analyzes the CMOS implementation of $2 \times 2$ and $4 \times 4$ MIMO OFDM schemes adopted in WLAN and WMAN transceivers. Different configurations with a varying number $P$ of parallel FFT and IFFT processors and thus varying clock frequency are considered. Reported values prove that, for the considered real world cases, working with a single processor, $P = 1$, permits the minimization of both area and leakage power consumption with respect to a solution with higher parallelization degree, $P \geqslant 2$. In terms of total power cost the results in Table 4 show a weak dependence on the number of parallel processors. The reason why in sub-micron CMOS technologies for FFT/IFFT processing in MIMO OFDM schemes a solution with lower parallelization degree allows for a better trade-off is the following: for the considered $2 \times 2$ and $4 \times 4$ WLAN and WMAN cases the real-time clock frequency of a single processor solution, although higher than parallel configurations, is $\leqslant$160 MHz (worst case of $4 \times 4$ 802.11n with the 1 sample/clock cycle throughput of our architecture). Such value can be achieved using low-leakage HVT cells only.

As consequence real-time processing is achieved with minimal area cost and hence minimal leakage power. The dynamic power is roughly the same of more parallel solutions since, for fixed $V_{DD}$ values, the increase in frequency is counterbalanced by the decrease in circuit complexity. The results in Table 4 refer to 65 nm CMOS but similar considerations can be derived for 45 nm and 90 nm CMOS implementations.

### 5.4. Comparison with state-of-art CMOS IP cores

To assess the effectiveness of the automated IP configuration and generation tool vs. the state of the art. Tables 5 and 6 compare the obtained CMOS synthesis results to those of known FFT cores, published in recent literature. The comparison between IP cores based on different architectures and/or implemented in different technologies is not straightforward. For this reason, the implementations selected for the comparison are those with system level requirements (throughput and numerical accuracy) similar to those of our work. The focus is on FFT processors covering applications at two extremes: large size transform in DVB and high throughput transforms in UWB. The comparison includes an application specific instruction set processor (ASIP) [43], several macrocells specifically designed for DVB–T or UWB [6,13,44–48], and a macrocell obtained by an automatic IP generator [8].

For DVB the macrocell generated by the proposed tool features low-complexity while maintaining similar performance of other works: throughput of 9 MS/s, SQNR higher than 40 dB, programmable transform length between 1024 and 8192. Compared to custom designs our IP generator increases the design flexibility as the ASIP FFT in [43] but with a much lower circuit complexity.

For UWB our IP core has comparable performance and complexity to the custom architectures proposed in [44]; with respect to other known macrocells adopting multiple parallel data paths

**Table 4**
Synthesis results for MIMO FFT/IFFT cores, CMOS 65 nm 1.1 V.

| | Parallel processors | Fclk (MHz) | Area ($mm^2$) | Power | | |
|---|---|---|---|---|---|---|
| | | | | Leak (μW) | Dynamic (μW/MHz) | Total (mW) |
| *WLAN* | | | | | | |
| 2 × 2 | 1 | 80 | 0.0437 | 3.6 | 46.7 | 3.739 |
| 2 × 2 | 2 | 40 | 0.0874 | 7.2 | 93.4 | 3.743 |
| 4 × 4 | 1 | 160 | 0.0437 | 3.6 | 46.7 | 7.475 |
| 4 × 4 | 2 | 80 | 0.0874 | 7.2 | 93.4 | 7.479 |
| 4 × 4x | 4 | 40 | 0.1747 | 14.4 | 186.8 | 7.486 |
| *WMAN* | | | | | | |
| 2 × 2 | 1 | 80 | 0.0861 | 8.4 | 81.2 | 3.256 |
| 2 × 2 | 2 | 40 | 0.1722 | 16.8 | 162.4 | 3.265 |
| 4 × 4 | 1 | 160 | 0.0861 | 8.4 | 81.2 | 6.504 |
| 4 × 4 | 2 | 80 | 0.1722 | 16.8 | 162.4 | 6.513 |
| 4 × 4 | 4 | 40 | 0.3443 | 33.6 | 324.8 | 6.53 |

**Table 5**
Comparison vs. state-of-art DVB–T FFT cores (SQNR ⩾ 40 dB).

| IP | IP type | Arithmetic | Fclk (MHz) | K gates | RAM (bits) | ROM (bits) |
|---|---|---|---|---|---|---|
| [43] | ASIP | Fixed-point | 280 | 80 | 1.5 M | N.A. |
| [6] | Custom | Fixed-point | 16 | 139 | 206 K | 161 K |
| [8] | Generator | Fixed-point | 9 | 49 | 256 K | 300 K |
| [13] | Custom | BFP | 8 | 91 | N.A. | N.A. |
| [48] | Custom | Fixed-point | 9 | 55 | 18 K | N.A. |
| Our | Generator | BFP | 9 | 37 | 215 K | 8 K |

**Table 6**
Comparison vs state-of-art UWB-OFDM FFT cores.

| IP | SQNR dB | IP type | K gates | Memory (bits) | Arithmetic | Throughput |
|---|---|---|---|---|---|---|
| [46] | 33 | Custom | 130 | 0.6 K RAM 4 K ROM | Fixed-point | 4 data path at 450 MHz, 1.8 GS/s |
| [47] | 27 | Custom | 85 | 1 K ROM 1.25 K RAM | Fixed-point | 2 data path at 400 MHz 800 MS/s |
| [45] | 30 | Custom | 73 | N.A. | Fixed-point | 4 data path at 250 MHz, 1 GS/s |
| [44] | 20 | Custom | 15 | N.A. | Fixed-point | 1 data path at 528 MHz, 528 MS/s |
| [44] | 32.5 | Custom | 22.5 | N.A. | Fixed-point | 1 data path at 528 MHz, 528 MS/s |
| Our | 29 | Generator | 15.2 | 2.48 K RAM 0.17 K ROM | BFP | 1 data path at 528 MHz, 528 MS/s |

our approach leads to a much lower complexity but at the expense of a lower throughput.

## 6. FPGA implementation results

### 6.1. Synthesis and characterization results

The proposed macrocells, configured as in Section 4.3, have been implemented on FPGA devices. Different technologies are available for logic programming: antifuse, flash and SRAM. The latter is the most suited for consumer applications due to easier reprogrammability and hence reduced development costs and time to market. Moreover SRAM-based FPGAs have higher compatibility with conventional CMOS processes thus easing the integration of embedded FPGA cells in complex SoC to create a reconfigurable SoC platform. Table 7 shows the implementation results on two SRAM-based FPGA families, Xilinx Spartan 3 [49,50] and Virtex 4 [51], both realized in 90 nm 1.2 V CMOS technology. The RTL synthesis and the subsequent back end FPGA implementation phase have been carried out using Mentor Graphics Precision and Xilinx ISE CAD tools, respectively. Reported data are the % occupation of available slices, the used RAM blocks, the clock frequency and the power consumption in case of real-time processing of OFDM symbols.

The maximum achievable throughput on Spartan FPGA for the different configurations is roughly 80 MS/s. Hence for xDSL, WLAN 802.11-a/n, WMAN 802.16-d/e and DVB–T/H/SH applications the FFT/IFFT macrocells can be fitted on low cost Spartan FPGAs. Such devices are available also for high volume production with a cost of few dollars. The power consumption of the FPGA realizations is 115 mW for DAB, roughly 250 mW for the WLAN, WMAN and DVB standards and less than 400 mW for the wired xDSL one. Note that in Table 7 only one third of the device is occupied and hence the rest of the resources are available to build a complete OFDM communication system on FPGA. To this aim it is worth noting that in the FPGA DAB receiver proposed by [5] the FFT-based demodulation represents half of the total receiver complexity. As far as MIMO systems are concerned, the throughput requirements of 2 × 2 MIMO schemes (see Table 4) can be sustained by a single IP core on Spartan FPGA while 4 × 4 MIMO schemes require the implementation of multiple cores. Such parallelization is possible on Spartan 1000 since only one third of the available FPGA resources are occupied by the FFT IP.

The high throughput of the UWB scheme does not permit a real-time realization of a single core on a low cost FPGA family. A possible solution, similar to that discussed for FPGA devices in [52] and for the CMOS processors in [45,46], is using an array of four parallel cores. This way the required clock frequency is reduced from 528 MHz to roughly 132 MHz. To support a parallel array of 4 processors the logic resources of a more complex and faster Virtex 4 FPGA are needed. However the power budget at real-time for the four cores is 950 mW, not practical for mobile UWB applications. Moreover a Virtex 4 device has a much higher cost than Spartan

**Table 7**
FPGA results, for (de)-modulation in different OFDM standards.

| Standard | FPGA device | Slice (%) | Frequency (MHz) | Power (mW) | RAM (block) |
|---|---|---|---|---|---|
| DVB | Spartan 3-1000 | 30 | 9 | 230 | 48 |
| DAB | Spartan 3-1000 | 28 | 8.26 | 115 | 32 |
| xDSL | Spartan 3-1000 | 34 | 35 | 390 | 47 |
| WLAN | Spartan 3-500 | 35 | 40 | 264 | 24 |
| WMAN | Spartan 3-1000 | 29 | 20 | 278 | 32 |
| UWB | Virtex 4 LX40 | 32 | 132 | 950 | 120 |

devices and hence it is practical for fast prototyping but not for production.

Summarizing, for multi-carrier communication standards (xDSL, WMAN, WLAN, DVB, DAB) featuring moderate throughput an FPGA solution is feasible in terms of real –time processing and cost and has the advantage of reconfigurability vs. ASIC. The power consumption of the FPGA solutions, in the range of hundreds of mW, is more than one order of magnitude higher than the implementations in 90 nm, 65 nm and 45 nm CMOS technologies. The FPGA power consumption is too high for very low power mobile terminals such as smart phones but can be sustained for applications such as xDSL modem, multi-standard OFDM terminals plugged in desktop/notebook personal computers, home/car receivers of DVB and DAB contents, terminals for WLAN and WMAN access points in home, office or public buildings. In case of UWB, due to its high throughput requirements, an implementation on dedicated CMOS circuitry is mandatory.

### 6.2. Comparison with state-of-art FPGA implementations

As stated in Section 5.4 comparing IP cores based on different architectures and/or implemented in different FPGA devices is not straightforward. For a fair comparison we selected published works with similar system level requirements and targeting FPGA devices belonging to the same architecture and technology family.

As example to support a DAB compliant FFT demodulator our macrocell requires 28% of 1 Million gates SRAM FPGA while the macrocell in [5] requires 1/3 of a 6 Million gate SRAM FPGA. Our results are comparable to those in [21] and [22]. In [21] a fixed-point FFT processor is implemented on a Spartan 3 1500 FPGA; it supports transform sizes up to 4096 with a throughput of 30 MS/s. In [21] the 8192 DVB mode and the high throughput of MIMO WLAN and WMAN applications are not supported. In [22] a 12 bits fixed-point FFT processor is implemented in real-time for different standards (DVB–T, DAB, 802.11a) on a Spartan 3 FPGA. The required complexity is higher than our design since in [22] it amounts to 58% of a Spartan 1000 device.

## 7. Conclusions

This paper presented an FFT/IFFT scalable architecture template coupled with a computer assisted design environment. Based on a pipeline cascade of mixed radix-2/4 stages the architecture can be configured in terms of: transform length and type, machine arithmetic (fixed-point or BFP or CBFP), size of the I/O, of the twiddle coefficients and of the internal data path. An automatic configuration flow generates application specific processors with minimum circuit and memory complexity within the numerical accuracy budget of the target application. The effectiveness of the tool is demonstrated through FPGA (Spartan and Virtex families) and CMOS (90 nm, 65 nm and 45 nm multi-$V_t$ technologies) implementations of the baseband processing in embedded OFDM transceivers: 802.11 WLAN, 802.16 WMAN, Digital Audio and Video Broadcasting, Ultra Wide Band, Broadband on Power Lines, xDSL. Compared to state-of-art custom designs, considering similar performance target, our approach reduces the time needed for design space exploration and IP configuration while the implementation results in CMOS and FPGA technologies feature low-complexity. The proposed automatic IP generator is the first addressing the requirements of all OFDM and MIMO–OFDM standards while most of known works are specifically optimized for a subset of them. Comparing FPGA and standard-cells CMOS technologies, the former allows for real-time processing at moderate cost and power consumption (hundreds of mW) for all standards except UWB. The advantages of FPGA vs. ASIC are the reconfigurability for multi-standard support and the reduced development time and cost. Target applications for FPGA can be xDSL modem, multi-standard OFDM terminals plugged in desktop/notebook personal computers, home/car receivers of DVB/DAB contents, terminals for WLAN and WMAN access points in home, office or public buildings. On the other hand CMOS standard-cells allow for extremely low power consumption and area costs (few mWs and below 0.1 $mm^2$ in 45 nm CMOS) for all the standards, including real-time UWB transceivers.

## Acknowledgments

This work was supported by Italian PRIN 2005093524-002 and by the Network of Excellence NEWCOM++.

## References

[1] J. Cooley, J. Tukey, An algorithm for the machine calculation of complex Fourier series, Math. Comput. (19) (1965) 297–301.
[2] Atmel Corporation, DIOPSIS 740 dual-core DSP, 7001ADPS-09/04 edition, 2004.
[3] Texas Instruments, TMS320DM644x Processors Video Benchmarks, SPRT379A edition, 2006.
[4] G. Bi, E. Jones, A pipelined FFT processor for word sequential data, IEEE Trans. Acoust. Speech Signal Procss. 37 (12) (1988) 1982–1985.
[5] R. Cabral, S. Escarigo, H. Neto, H. Sarmento, Implementation of a DAB receiver with FPGA technology, in: Proceedings of the IEEE ICCE 2006, January 2006, pp. 397–398.
[6] Chua-Chin Wang, Jian-Ming Huang, Hsian-Chang Cheng, A 2 K/8 K mode small-area FFT processor for OFDM demodulation of DVB–T receivers, IEEE Trans. Consum. Electr. 51 (1) (2005) 28–32.
[7] A. Cortes, I. Velez, J. Sevillano, A. Irizar, An approach to simplify the design of IFFT/FFT cores for OFDM systems, IEEE Trans. Consum. Electr. 52 (1) (2006) 26–32.
[8] A. Cortes, I. Velez, J. Sevillano, A. Irizar, An FFT core for DVB–T/DVB–H receivers, in: Proceedings of the IEEE ICECS 2006, December 2006, pp. 102–105.
[9] J. Ding, J. McCanny, Y. Hu, Rapid design of application specific FFT cores, IEEE Trans. Acoust. Speech Signal Process. 47 (5) (1999) 1371–1381.
[10] L. Fanucci, M. Forliti, P. Terreni, FAST: FFT ASIC automated synthesis, Integrat. VLSI J. 33 (1-2) (2002) 23–37.
[11] M.J. Canet, F. Vicedo, V. Almenar, J. Valls, FPGA implementation of an IF transceiver for OFDM-BASED WLAN, in: Proceedings of the IEEE SIPS 2004, 2004, pp. 227–232.
[12] S. He, M. Torkelson, A new approach to pipeline FFT processor, in: Proceedings of the IEEE IPPS 1996, April 1996, pp. 766–770.
[13] X. Li, Z. Lai, J. Cui, A low power and small area FFT processor for OFDM demodulator, IEEE Trans. Consum. Electr. 53 (2) (2007) 274–277.
[14] Y. Lin, C. Lee, Design of an FFT/IFFT processor for MIMO OFDM systems, IEEE Trans. Circuits Syst. I 54 (4) (2007) 807–815.
[15] G. Nordin, P. Milder, J. Hoe, M. Puschel, Automatic generation of customized discrete Fourier transform IPs, in: Proceedings of the IEEE Design Automation Conference (DAC), June 2005, pp. 471–474.
[16] T. Sansaloni, A. Perez-Pascual, V. Torres, J. Valls, Efficient pipeline FFT processors for WLAN MIMO–OFDM systems, Electron. Lett. 41 (2005) 1043–1044.

[17] S. Saponara, L. Fanucci, VLSI design investigation for low-cost, low-power FFT/IFFT processing in advanced VDSL transceivers, Microelectr. J. 34 (2) (2003) 133–148.
[18] Actel, CoreFFT Fast Fourier Transform generator v3.0. December 2005.
[19] N. L'Insalata, S. Saponara, L. Fanucci, P. Terreni. Automatic generation of low-complexity FFT/IFFT cores for multi-band OFDM systems, in: Proceedings of the Euromicro DSD 2007, August 2007, pp. 361–368.
[20] N. L'Insalata, S. Saponara, L. Fanucci, P. Terreni, Automatic synthesis of cost effective FFT/IFFT cores for VLSI OFDM systems, IEICE Trans. Electron. E-91C (4) (2008) 487–496.
[21] H.-G. Yeh, G. Truong, Speed and area analysis of memory based processors in a FPGA, in: Proceedings of the IEEE WTS, 2007, pp. 1–6.
[22] M. Petrov, M. Glesner, Optimal FFT architecture selection for OFDM receivers on FPGA, in: Proceedings of the IEEE ICFPT, 2005, pp. 313–314.
[23] High rate ultra wideband PHY and MAC standard, December 2005. Standard ECMA-368.
[24] M. Kornfeld, DVB–H – The emerging standard for mobile data communication, in: Proceedings of the IEEE International Symposium on Consumer Electronics, September 2004, pp. 193–198.
[25] B. Farahani, M. Ismail, WiMAX/WLAN radio receiver architecture for convergence in WMANS, in: Proceedings of the IEEE 48th Midwest Symposium on Circuits and Systems, August 2005, pp. 1621–1624.
[26] C. Del-Toso, M. Nava, A short overview of the VDSL system requirements, IEEE Commun. Mag. 40 (12) (2002) 82–90.
[27] J. Walko, Click here for VDSL2, IEE Commun. Eng. 3 (4) (2005) 9–12.
[28] P. Amirshahi, M. Navidpour, M. Kavehrad, Performance analysis of uncoded and coded OFDM broadband transmission over low voltage power-line channels with impulsive noise, IEEE Trans. Power Deliver. 21 (4) (2006) 1927–1934.
[29] M. Deinzer, M. Stoger. Integrated PLC-modem based on OFDM, in: Proceedings of the 3rd International Symposium on Powerline Communications and its Applications, March 1999, pp. 92–97.
[30] D. Skellern, A high-speed wireless LAN, IEEE Micro. 17 (1) (1997) 40–47.
[31] IEEE P802.11, Wireless LANs joint proposal: High throughput extension to the 802.11 standard: PHY, January 2006. IEEE 802.11-05/1102r4.
[32] J. Bingham, Multicarrier modulation for data transmission: an idea whose time has come, IEEE Commun. Mag. 28 (5) (1990) 5–14.
[33] S. Perels, D. Haene, P. Luethi, A. Burg, N. Felber,W. Fichtner, H. Bolcskei, ASIC implementation of a MIMO OFDM transceiver for 192 Mbps WLAN, in: Proceedings of the IEEE ESSCIRC 2005, 2005, pp. 215–218.
[34] Y. Jung, J. Kim, S. Lee, H. Yoon, J. Kim, Design and implementation of MIMO–OFDM baseband processor for high-speed wireless LANs, IEEE Trans. Circ. Syst. II 54 (7) (2007) 631–635.
[35] N. Weste, D. Skellern, VLSI for OFDM, IEEE Commun. Mag. 36 (10) (1998) 127–131.
[36] S. Rosati, S. Cioni, M. Neri, A. Vanelli-Coralli, G. Corazza, Joint symbol timing and carrier frequency recovery for DVB–SH system, in: Proceedings of the IEEE IWSSC 2007, 2007, pp. 79–83.
[37] S. Prakash, V. Rao, Fixed-point error analysis of radix-4 FFT, Signal Process. 3 (2) (1981) 123–133.
[38] P. Duhamel, M. Vetterli, Fast Fourier Transforms: a tutorial review and a state of the art, Signal Process. 19 (4) (1990) 259–299.
[39] C. Hui, T.J. Ding, J. McCanny, R. Woods, Error analysis of FFT architectures for digital video applications, in: Proceedings of the IEEE ICECS 1996, October 1996, pp. 820–823.
[40] L. Fanucci, M. Rovini, A low-complexity and high resolution algorithm for the magnitude approximation of complex numbers, IEICE Trans. Fund. E85-A (7) (2002) 651–654.
[41] L. Benini, G. De Micheli, E. Macii, Designing low power circuits: practical recipes, IEEE Circ. Syst. Mag. 1 (2001) 6–25.
[42] F. Boeuf et al., A conventional 45 nm CMOS low-cost platform for general purpose and low power applications, in: Proceedings of the IEEE IEDM 2004, 2004, pp. 425–248.
[43] J. Lee, J. Moon, K. Heo, M. Sunwoo, S. Oh, I. Kim, Implementation of application-specific DSP for OFDM systems, in: Proceedings of the IEEE ISCAS 2004, May 2004, pp. 665–668.
[44] A. Cortes, I. Velez, A. Irizar, J.F. Sevillano, Area efficient IFFT/FFT core for MB-OFDM UWB, Electron. Lett. 43 (11) (2007) 649–650.
[45] Sang-In Cho, Kyu-Min Kang, Sang-Sung Choi, Implementation of 128-point Fast Fourier Transforms processor for UWB systems, in: Proceedings of the IEEEIWCMC 2008, 2008, pp. 210–213.
[46] M. Shin, H. Lee, A high-speed four-parallel radix-2/4 FFT/IFFT processor for UWB applications, in: Proceedings of the IEEE ISCAS 2008, 2008, pp. 960–963.
[47] J. Lee, H. Lee, S.-I. Cho, S.-S. Choi, A high speed, low complexity radix-$2^4$ FFT processor for MB-OFDM UWB systems, in: Proceedings of the IEEE ISCAS 2006, May 2006, pp. 4719–4722.
[48] R.M. Jiang, An area efficient FFT architecture for OFDM digital video broadcasting, IEEE Trans. Consum. Electr. 53 (4) (2007) 1322–1326.
[49] Xilinx Inc, DS099 Spartan3 FPGA family complete data sheet, XC3S1000, 25 May 2007.
[50] Xilinx Inc, DS312 Spartan3E FPGA family: complete data sheet, XC3S500E, 29 May 2007.
[51] Xilinx Inc, DS112 Virtex-4 family overview, XC4VLX40, 23 January, 2007.
[52] R.S. Sherratt, O. Cadenas, N. Goswami, A low frequency FFT core implementation for multiband full-rate ultra-wideband (UWB) receivers, IEEE Trans. Consum. Electr. 51 (3) (2005) 798–802.
[53] M. Ruggiero, L. Benini, S. Medardoni, D. Bertozzi, Commmunications and I/O architecture for highly integrated MPSoC platforms, in: Artist2 Workshop on Excecution Platforms, May 2006.
[54] L. Fanucci, R. Saletti, S. Saponara, Parametrized and reusable VLSI macro cells for the low-power realization of 2-D discrete–cosine–transform, Microelectr. J. 32 (12) (2001) 1025–1045.

**Sergio Saponara** got the Laurea degree, cum laude, and the Ph.D. in Electronic Engineering from the University of Pisa in 1999 and 2003, respectively. In 2002 he was with IMEC, Leuven (B), as Marie Curie Research Fellow. Since 2001 he collaborates with Consorzio Pisa Ricerche in Pisa. He is senior researcher at University of Pisa in the field of electronic circuits and systems for telecom, multimedia, space and automotive applications. He holds the chair of electronic systems for automotive and automation at the Faculty of Engineering. He co-authored more than 90 scientific publications and 4 patents.

**Nicola E. L'Insalata** received the M.Sc. degree in Electronic Engineering and the Ph.D. degree in Information Engineering, both from University of Pisa, in 2004 and 2008, respectively. Currently he is with Synopsys. From 2005 to October 2008 he collaborated with the Microelectronic Systems Division of Consorzio Pisa Ricerche on several projects of industrial relevance in the fields of embedded systems. His research addresses design of digital macrocells with emphasis on design reuse methodologies.

**Luca Fanucci** got the Laurea and the Ph.D. degrees in Electronic Engineering from University of Pisa in 1992 and 1996, respectively. From 1992 to 1996, he was with ESA/ESTEC, Noordwijk (NL), as research fellow. From 1996 to 2004 he was a senior researcher of the CNR in Pisa. He is Professor of Microelectronics at the University of Pisa. His research interests include VLSI architectures for integrated circuits and systems. Prof. Fanucci co-authored more than 150 scientific publications and he holds more than 10 patents. He was program chair of IEEE Euromicro DSD 2008 and IEEE DATE Designer's Forum.