2016 11th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)

Dynamically Reconfigurable FFT Processor for Flexible OFDM Baseband Processing

Mário Lopes Ferreira*, Amin Barahimi† and João Canas Ferreira‡
INESC TEC*†‡ and Faculty of Engineering of the University of Porto*‡
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
Email: *mario.l.ferreira@inesctec.pt, †amin.barahimi@inesctec.pt, ‡jcf@fe.up.pt

***Abstract*—The Physical layer architectures for the next generation of wireless devices will be characterized by a high degree of flexibility for real-time adaptation to communication conditions variability. OFDM-based architectures are strong candidates for the Physical layer implementation in 5G systems and one of the most important baseband processing operations required by this waveform is the Fast Fourier Transform (FFT). This paper proposes a dynamically reconfigurable FFT processor supporting FFT sizes and throughputs required by the most widely used wireless standards. The FFT reconfiguration was achieved by means of FPGA-based Dynamic Partial Reconfiguration (DPR) techniques, which enables run-time FFT size adaptation according to communication requirements and better resource utilization. The impact of DPR in terms of reconfiguration time and power consumption overhead was evaluated. The obtained results encourage the exploitation of DPR techniques to implement reconfigurable hardware infrastructures for OFDM baseband processing engines.**

***Index Terms*—FFT, Dynamic Partial Reconfiguration (DPR), FPGA, OFDM, Cognitive Radio**

I. Introduction

The fifth generation of cellular communications (5G) will go far beyond its predecessors in aspects such as data rates, energy efficiency, spectrum efficiency and management. In order to achieve 5G requirements, Physical layer (PHY) architectures should be highly flexible and adaptable to allow a deeper exploitation of key enabling technologies for 5G, such as Cognitive Radio (CR) [1], which can improve spectrum utilization efficiency and provide support for a multitude of wireless standards [2]. There are several waveform candidates for future wireless communications, such as OFDM - the dominant waveform in 4G -, Filter Bank Multi-Carrier (FBMC) [3] or Generalized Frequency Division Multiplexing (GFDM) [4]. In particular, Andrews et al. [5] suggest that an OFDM-based PHY well-suited for 5G should allow some of its communication parameters to be flexibly adapted according to channel conditions instead of being fixed for the worst-case communication scenario. One of the communication parameters that can vary depending on the communication channel conditions and standard in use is the Fast Fourier Transform (FFT) block size. In fact, the FFT is one of the most relevant and complex baseband operations in wireless systems. Besides being the heart of an OFDM-based system, the FFT also plays an important role in GFDM receivers and FBMC transceiver implementations using fast convolution [6]. Additionally, FFTs can also be used in CR for spectrum sensing purposes [7]. The FFT size can be handled by considering an FFT processor supporting the largest possible size. However, from a resource usage perspective, this is clearly an inefficient approach. In contrast with this worst-case solution is the implementation of a flexible FFT processor with support for multiple FFT sizes, able to reconfigure itself and adapt the FFT size in response to run-time communication demands. By adjusting the operation according to the system's needs, resource usage efficiency improves, which can bring potential power consumption savings. Apart from flexibility concerns, future communication systems will also emphasize energy efficiency and cost [5]. Thus, the design of such an FFT processor is a considerable contribution in the context of hardware infrastructures for the next generation of wireless devices.

This paper presents research work targeting the implementation of a flexible, resource-efficient, power-aware and dynamically reconfigurable FFT processor for flexible OFDM baseband engines. The FFT processor supports the most used FFT sizes in wireless communications and has the capacity to dynamically modify the FFT size at run time. Moreover, the architecture can be extended to accommodate other sizes for future standards. Through resources reuse, the FFT processor can also handle the computation of double-stream FFT for the smallest supported size. This feature is useful to support multiple-stream operation modes present in some wireless protocols. FPGAs were chosen as the hardware platform for this work due to their good compromise between flexibility, throughput and power consumption. Furthermore, FPGA-based Dynamic Partial Reconfiguration (DPR) was the technique exploited to reconfigure the FFT processor. This work is part of the ongoing research effort described in [8] and the proposed FFT processor was implemented also considering its integration in a reconfigurable Non Contiguous-OFDM baseband processor.

The rest of the paper is structured as follows: the following section discusses some related work; Section III provides fundamental background about Dynamic Partial Reconfiguration; Section IV describes the implemented reconfigurable

This work is financed by the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project *POCI-01-0145-FEDER-006961*, by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project *EXCL/EEI-TEL/0067/2012* and through Ph.D. Grant *PD/BD/105860/2014*

FFT processor; Section V presents and discusses the results obtained using a fully-working prototype implementation; at last, Section VI presents some conclusions.

II. Related Work

This section discusses some related work on FFT processors for OFDM baseband engines. In [9], Boopal et al. present a reconfigurable Single-Delay Feedback (SDF) FFT architecture for WiMAX systems. It supports several power-of-two FFT sizes and can handle multiple streams by interleaving the input data. The FFT reconfiguration is based on multiplexing modules for different sizes, so little hardware is reused. Wang et al. [10] propose a run-time reconfigurable FFT/IFFT architecture for 3GPP-LTE/LTE Advanced systems, by employing a mixed-radix SDF architecture to support non-powers-of-two FFT sizes. Like in [9], DPR is not explored and the run-time reconfiguration is implemented through bypass structures which connect the pipeline elements necessary for computing an FFT with the desired size. Dynamic Partial Reconfiguration (DPR) techniques are explored by Venilla et al. [11] to implement a reconfigurable FFT processor employing an SDF architecture on a Xilinx Virtex-II Pro board. Multiplexer-based and DPR-based approaches are compared and it is claimed that the later approach presented resource savings around 41% for slices and 32% for LUTs. However, this FFT processor supports only powers-of-two sizes and the impact of DPR in terms of power consumption overhead was not evaluated.

From an overview perspective, the majority of FFT processors designed for OFDM systems are reconfigurable via multiplexer-based schemes, which is not efficient regarding resource utilization. Moreover, in most cases, only powers-of-two FFT sizes are supported and this may not be sufficient to support recent and future wireless standards. The FFT processor proposed in this work brings together the support for non-powers-of-two FFT sizes with the exploration of DPR techniques and the evaluation of their impact in terms of reconfiguration times and power consumption.

III. FPGA Dynamic Partial Reconfiguration

SRAM-based FPGAs are intrinsically reconfigurable: the device will perform a new functionality, once a new bitstream is loaded into it. The traditional way to alter the functionality implemented on an FPGA consists of turning it off, downloading a new configuration onto it and rebooting the system. The reconfiguration process is independent of the application [12] and is designated by *Static Reconfiguration*. Alternatively, the reconfiguration process can be part of the application, such that it occurs on-the-fly during execution time, without turning off the FPGA, in response to a certain system functional requirement. This process is known as *Dynamic Reconfiguration*. Quite often, it is only necessary to reconfigure some particular regions of the system during run-time, keeping the remaining areas unaltered - *Dynamic Partial Reconfiguration* (DPR). In DPR, portions of the programmable logic - PL - are designated to be reconfigured during run-time: *Reconfigurable Partitions* (RPs). The functionality of the RPs can be changed while the static parts of the PL continue to operate [13], making the system highly flexible and adaptable. For a given RP functionality, a partial bitstream is generated. As opposed to complete bitstreams, the size of partial bitstreams depends on the reconfiguration effort (reconfiguration area or differences intended to be introduced in the design). In [13], several interesting DPR features other than flexibility are pointed out: feature wealth, system upgradability, area reduction and resource savings. Nevertheless, DPR imposes some design challenges, as the designer should be aware of the reconfiguration time overhead, in order to not violate the time requirements of the whole system. The reduction of reconfiguration times will also mitigate the reconfiguration power overhead. In turn, a small DPR power overhead combined with area reduction and resources savings can lead to better overall power efficiency.

IV. Implementation

The majority of FFT sizes required by the most used wireless communication standards are powers-of-two, but 3GPP-LTE requires also FFT sizes which are are not. For instance, the 3GPP-LTE operation mode for 15MHz requires a 1536-point FFT. The implemented FFT processor addresses the most frequent FFT sizes in 3G/4G wireless standards (64, 128, 256, 512, 1024, 1536 and 2048), and in order to support these FFT sizes a Cooley-Tukey Mixed-Radix-$2^2/2/3$ algorithm was employed. The Cooley-Tukey algorithm [14] was chosen, since it offers a systematic procedure to compute the FFT for any factorization on the FFT size - N -, with a reasonable computational complexity. Its Mixed-Radix algorithm variant is exhaustively exposed in [15]. As this FFT processor is to be applied in a mobile telecommunications environment, where continuous data streams need to be processed and the FFT requirements may change, it is convenient that the FFT processor has a scalable architecture supporting the ever increasing throughputs of wireless communications. Thus, a pipelined architecture was chosen for the reconfigurable FFT processor, in detriment of a memory-based architecture, whose mode of operation is iterative and does not exploit the hardware's potential for parallel execution. From several pipelined architectures presented in literature, Single Delay Feedback (SDF) was adopted because it is simple to implement, has low memory requirements and offers satisfactory throughputs. In [16] details about hardware processing structures to compute FFTs are provided.

The proposed system was implemented on a Xilinx Virtex-7 FPGA board (device: XC7VX485T-2FFG1761C) running at 100 MHz. The arithmetic operations involved in FFT calculation are performed with 16-bit fixed-point values for real and imaginary parts. The FFT input stream is fed in natural order and the FFT output results are also produced in natural order. A general overview of the system architecture is presented in Figure 1.

The resources needed to implement the FFT reconfigurable pipeline were divided in six Reconfigurable Partitions (RPs) embedded in an AXI4-Stream [17] IP core. Through DPR

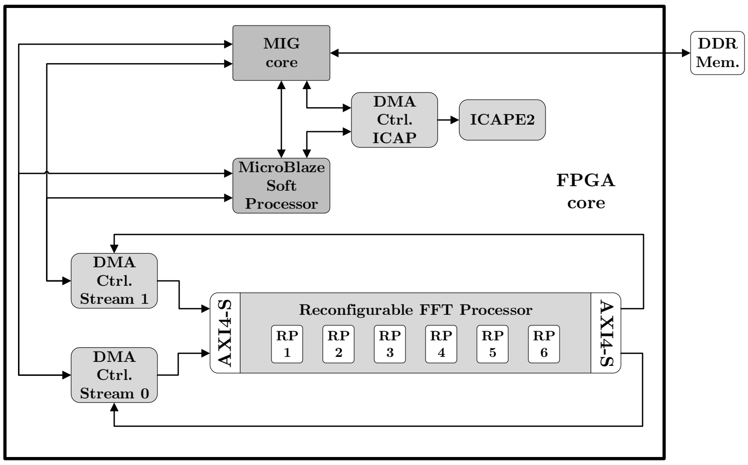

Figure 1: High Level System Architecture

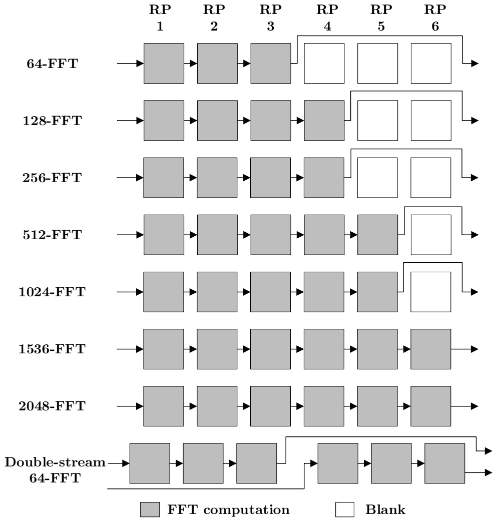

Figure 2: RPs role in each FFT configuration

and according to the FFT configuration to be used, a set of partial bitstreams defining the functionality of each RP is sent to the FPGA configuration port. For larger FFT sizes, as 1536 and 2048, all six RPs are needed for FFT processing, whereas for smaller FFT sizes some RPs are left *blank* (without any implemented logic) and can be reused. Several wireless standards have operation modes supporting multiple data streams (e.g.: IEEE 802.11g for 54 Mbps data rates can support two data streams, each of them requiring a 64-FFT operation). As the computation of one 64-FFT only requires three RPs, it is possible to use the remaining three RPs to perform another 64-FFT. This allows the parallel processing of a double data stream through resource reuse and hardware parallelism exploitation. In total, there are eight different FFT configurations: one for each addressed FFT size and another one for double-stream 64-FFT. Figure 2 explains the way RPs are used in each FFT configuration. The interconnections between the RPs belong to the system static part and are controlled using multiplexing schemes. Information regarding allocated resources and partial bitstream sizes for each RP is presented in Table I. The system infrastructure around the reconfigurable FFT pipeline - static part - consumes the following resources: 37054 (12.20% of available) slice LUTs, 29535 (4.86% of available) slice Registers, 168 (16.31% of available) BRAMs and 1 (0.04% of available) DSP block.

The reconfigurable FFT processor receives data from the DDR memory, computes the FFT and sends the results back to the DDR memory. If all these DDR read/write operations required the intervention of the MicroBlaze processor, the overall system performance would be deteriorated. In particular, the FFT throughput would be negatively affected. To enable the DDR memory access by the FFT processor without MicroBlaze control, Direct Memory Access (DMA) controllers for 32-bit data transactions are used. As our FFT processor can handle single-stream FFTs or double-stream 64-FFTs, two DMA controllers are employed, one for each data stream.

After the FPGA is powered on, a file system image containing a complete bitstream for the FPGA initial configuration and the partial bitstreams for each RP is loaded on the board flash memory. Upon FPGA complete configuration, the MicroBlaze copies all partial bitstreams to the DDR memory. The MicroBlaze is also be responsible for controlling the FFT dynamic reconfiguration. To perform DPR, partial bitstreams for the new FFT configuration need to be fetched from DDR memory and sent to the FPGA configuration memory. The access to the FPGA configuration memory is achieved through the Xilinx Internal Configuration Port (ICAPE2) primitive. To increase reconfiguration throughput and mitigate reconfiguration latency, another dedicated DMA controller for 32-bit data transactions is used to speed-up the partial bitstreams transfer from DDR to the ICAPE2. Xilinx AXI DMA IP cores were used to implement the DMA controllers for both DDR-FFT and DDR-ICAPE2 data transfers. The DMA cores are connected with a Xilinx MIG (Memory Interface Generator)

Table I: Resources per RP and partial bitstream sizes

	RP1	RP2	RP3	RP4	RP5	RP6
Slice LUTs	400 (0.13%)	2800 (0.92%)	2800 (0.92%)	1600 (0.53%)	1600 (0.53%)	2800 (0.92%)
Slice Registers	800 (0.13%)	5600 (0.92%)	5600 (0.97%)	3200 (0.53%)	3200 (0.53%)	5600 (0.92%)
BRAM	10 (0.97%)	10 (0.97%)	10 (0.97%)	10 (0.97%)	10 (0.97%)	10 (0.97%)
DSP	0 (0%)	40 (1.43%)	40 (1.43%)	20 (0.71%)	20 (0.71%)	40 (1.43%)
# RM variants	7	3	3	5	4	4
Partial Bitstream Max. size (KB)	67	201	183	123	125	211

core, which is responsible for the interface between the FPGA device and the DDR memory.

V. Results and Discussion

For all configurations, the FFT processor presented correct functional behaviour, as the output results produced were verified through comparison with MATLAB results. The FFT throughput was measured considering a scenario where a continuous flow of data is received by the FFT processor. For pipeline steady-state operation, the FFT processor presents a worst-case throughput of 88 MSamples/s. The overhead introduced by DDR read/write operations is the main cause cause for the difference between the observed throughput and the limit imposed by pipeline SDF architectures running at 100 MHz (100 MSamples/s). Still, the observed throughput is compatible with the majority of 3G and 4G wireless standards like 3GPP-LTE, WiMAX and IEEE 802.11. Moreover, the reconfigurable FFT core logic can run at higher clock frequencies if higher throughputs are required. The FFT configuration for double-stream 64-FFT is an example of resource utilization efficiency enabled by DPR techniques. However, in an OFDM transceiver, blank RPs could be reused not only for FFT purposes, but also for other baseband operations, such as cyclic prefix insertion/removal or M-PSK/M-QAM modulation/demodulation.

The latency introduced by DPR was also experimentally measured. The worst-case reconfiguration time observed was 2.3 ms and the reconfiguration throughput achieved is approximately 377 MiB/s, which is 94.25% of the limit imposed by the ICAPE2 interface - 400 MiB/s for 32-bit data transactions and $f_{clk} = 100\,\mathrm{MHz}$. The impact of DPR latency should be evaluated considering the scenarios where the system will operate. The implemented FFT processor will be integrated in an OFDM baseband engine that may operate in CR environments. Under these conditions, radio devices must adapt their internal operation, including the PHY configuration, to communicate without interference with licensed spectrum users. IEEE 802.22 [18] is a recent wireless standard intended for Wireless Regional Area Network (WRAN) communication using *white spaces* - frequencies allocated to licensed users, but not used in some time intervals or geographical locations - and it defines time parameters that protect licensed users from interferences. For example, radio devices communicating according to the IEEE 802.22 standard have 2 s to perform connection establishment on a channel - *channel setup time*. While using a channel, IEEE 802.22 devices must detect the presence of a licensed user in less than 2 s - *channel detect time* - and after that, all interfering communications have to cease in another period of 2 s - *channel move time*. These time parameters can be used as a reference for the reactivity requirements for reconfigurable hardware modules for CR devices. Although OFDM baseband processing comprises other modules, the FFT processor is one of the most computationally complex and demanding operations. Through the comparison with IEEE 802.22 time parameters, the latency introduced by DPR in the FFT processor is acceptable for the target application considered.

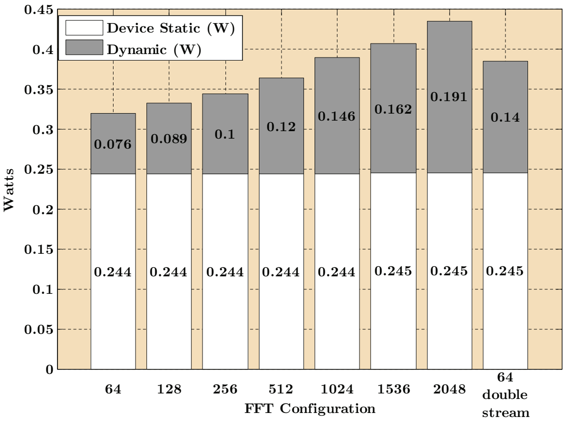

Figure 3: Power consumption estimations. FPGA device: XC7VX485T-2FFG1761C; Clock freq.: 100 MHz; Analysis tool: Vivado 2015.2; Post-Implementation power analysis with high confidence level; Activity derived from simulation files (.saif).

Before performing power measurements on the system, the power consumption of the FFT processor for every FFT configuration was estimated with the Vivado Power Analysis tool. The system used for estimation consisted only of the FFT reconfigurable pipeline. The obtained estimates are shown in Figure 3.

From these estimations, it is possible to observe that the power consumption variability is mostly related to dynamic power consumption, which in turn increases with the FFT size. Comparing the two configurations involving 64-FFTs, the dynamic power consumption for the double-stream case is nearly twice the dynamic power consumption for the single-stream case. This result is expected, as the double-stream FFT is built by replicating the same logic present in the single stream configuration.

Next, the power consumption of the whole system was measured by connecting a Texas Instruments (TI) USB Interface Adapter to the PMBus port on the VC707 board. The TI Fusion Digital Power Designer software was then used to monitor and log the power measurements. The parameter monitored was the output power of the power rail used to provide an operating voltage of 1 V to the XC7VX485T FPGA core. A limitation inherent to this measurement procedure is related to the fact that the time interval between two measurements provided by the TI Fusion Digital Power tool can reach 50 ms. So, the interval between two consecutive measurements can be bigger than the time interval in which one wants to perform measurements. To mitigate this limitation and increase the probability of obtaining measurements relative to the time intervals of interest, the experiment duration was extended.

Two operation regimes were considered: *Idle* (the FFT processor does not perform any FFT processing) and *Processing* (the FFT processor reads data from the DDR memory, performs FFT computation and sends the results back to DDR memory). The experimental setup for power measurements was the following: each FFT configuration remained for 10 min in the *Idle* and another 10 min in the *Processing* regime. The data fetched from DDR and sent to the FFT processor, as well as the the DMA buffer length register width (23-bit) used for DDR-FFT processor data transfers were the same for every FFT configuration. Thus, in the whole system, the only difference between FFT configurations was the logic implemented in the RPs. During all experiments, the FPGA temperature was kept at 34 °C. Figure 4 shows a plot of the power measured over time for the 2048-FFT configuration. Clearly, it is possible to identify two states in the power consumption that are relative to the two operation regimes defined: *Idle* for the first 10 min and Processing for the following 10 min. In the transition between the two operation regimes, a power increase is observed. This is due to node switching activity caused by FFT processing operations. A similar power behaviour was also observed for the other FFT configurations. Table II contains the average of output power obtained during *Idle* and *Processing* regimes, for the considered FFT configurations.

From the obtained results, one observes that the power consumption in both *Idle* and *Processing* regimes increase with the FFT size. This observation is more evident in the *Processing* regime and coincides with what is observed in the power estimations initially performed. A closer look to the behaviour in *Processing* regime, for the smallest and largest FFT sizes considered, one can observe that the difference in power consumption between 64-FFT and 2048-FFT (80 mW) is similar to the correspondent power estimations differences (115 mW). This suggests that the main variations in the system power consumption are mainly dependent on the FFT configuration in use. It also suggests that, in spite of measurement limitations previously mentioned, the experimental setup considered for power measurements produces fairly reasonable results.

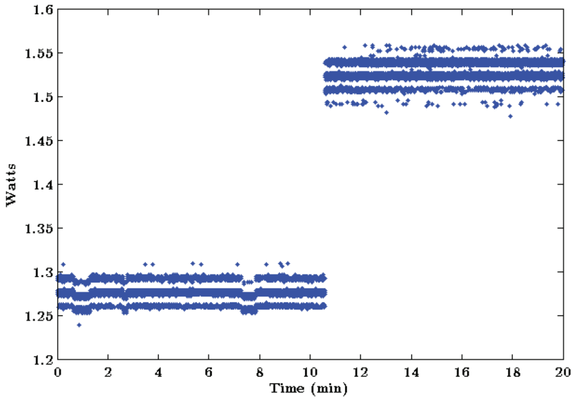

Figure 4: 2048-FFT measured power consumption

Table II: Average output power measured for each FFT configuration

FFT Config.	Idle regime average output power (W)	Processing regime average output power (W)	Power difference between Idle and Processing Regimes (W)
64-FFT	1.23	1.39	0.16
128-FFT	1.23	1.39	0.16
256-FFT	1.24	1.42	0.18
512-FFT	1.25	1.43	0.18
1024-FFT	1.26	1.48	0.22
1536-FFT	1.27	1.50	0.23
2048-FFT	1.28	1.52	0.24
64-FFT double-stream	1.25	1.55	0.30

A similar experimental setup was built to evaluate the power overhead introduced by DPR. As before, two operation regimes were considered: the *Idle* regime is similar to the one previously considered; and the *Reconfiguration* regime where the FFT processor is repeatedly switching between 64-FFT and 2048-FFT configurations. The switching between these two configurations was chosen because it corresponds to the scenario with the largest measured reconfiguration time - 2.3 ms - and, consequently, the largest reconfiguration energy overhead. Power measurements were retrieved considering the system in *Idle* regime for 10 min followed by 10 min in the *Reconfiguration* regime. Figure 5 presents the power consumption evolution along time. Again, it is possible to identify the power behaviours and map them to the operation regimes defined for this experience. The power increase observed during the second half of the experience is a consequence of partial reconfiguration activities, such as sending partial bitstreams to the FPGA configuration port through the ICAPE2 primitive.

Average output power obtained is 1.22 W for the *Idle* regime and 1.26 W for the *Reconfiguration* regime. The measured power overhead introduced by DPR is about 40 mW, which represents 3.28% of the power consumption in the *Idle* regime.

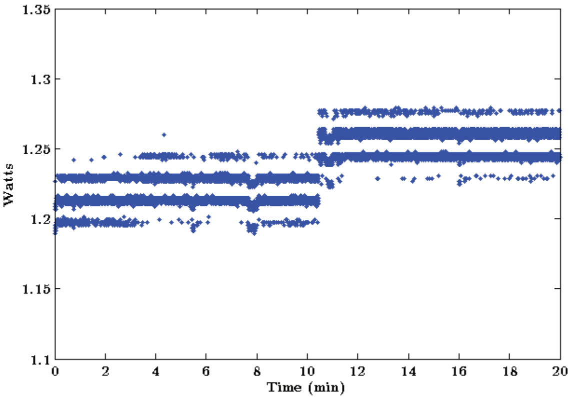

Figure 5: Measurements for reconfiguration power consumption

So, while computing, going from a 2048-FFT to a 64-FFT saves 130 mW at the cost of expending 40 mW (for 2.3 ms) during reconfiguration. DPR allows for power savings if the time interval between FFT reconfigurations is considerably larger than the reconfiguration time. Recalling the IEEE 802.22 time parameters and considering that the operation parameters of radio devices, such as the FFT size, will remain the same for an amount of time that is many times larger than the reconfiguration times, DPR can improve system's power efficiency. If radio device operation would need to be reconfigured every few milliseconds, DPR power overhead would become critical. However, this does not seem to a reasonable scenario in wireless communications.

VI. Conclusion

A dynamically reconfigurable FFT processor for flexible OFDM baseband processing engines, with support for FFT sizes and throughputs required by most recent and widely used wireless standards was implemented. Regarding FFT algorithm and architecture, a Mixed-Radix-2^2/2/3 SDF approach was adopted. The FFT processor was implemented on a Xilinx Virtex-7 FPGA board and Dynamic Partial Reconfiguration (DPR) techniques were explored to achieve run-time system reconfiguration. The use of DPR allows the reutilization of resources, leading to better resource usage efficiency.

The impact of DPR latency and power consumption overhead was measured and evaluated for CR communication scenarios. The largest reconfiguration time measured (2.3 ms) is within an acceptable range and shows that DPR application in flexible wireless communications is viable. Considering that radio device parameters, like the FFT size, will remain the same for time intervals many times bigger than the reconfiguration times, DPR-based implementation can lead to power savings, compared with a fixed hardware implementation for the worst-case operation scenario.

References

[1] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, Feb 2005.

[2] C.-I. Badoi, N. Prasad, V. Croitoru, and R. Prasad, "5G Based on Cognitive Radio," *Wireless Personal Communications*, vol. 57, no. 3, pp. 441–464, 2010. [Online]. Available: http://dx.doi.org/10.1007/s11277-010-0082-9

[3] P. Banelli, S. Buzzi, G. Colavolpe, A. Modenini, F. Rusek, and A. Ugolini, "Modulation Formats and Waveforms for 5G Networks: Who Will Be the Heir of OFDM?: An overview of alternative modulation schemes for improved spectral efficiency," *Signal Processing Magazine, IEEE*, vol. 31, no. 6, pp. 80–93, Nov 2014.

[4] G. Fettweis, M. Krondorf, and S. Bittner, "GFDM - Generalized Frequency Division Multiplexing," in *IEEE 69th Vehicular Technology Conference, 2009. VTC Spring 2009.*, April 2009, pp. 1–4.

[5] J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, and J. Zhang, "What Will 5G Be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[6] O. Font-Bach, N. Bartzoudis, X. Mestre, D. Lopez-Bueno, P. Mege, L. Martinod, V. Ringset, and T. Myrvoll, "When SDR meets a 5G candidate waveform : Agile use of fragmented spectrum and interference protection in PMR networks," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 56–66, December 2015.

[7] T.-H. Yu, O. Sekkat, S. Rodriguez-Parera, D. Markovic, and D. Cabric, "A Wideband Spectrum-Sensing Processor With Adaptive Detection Threshold and Sensing Time," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 58, no. 11, pp. 2765–2775, Nov 2011.

[8] M. L. Ferreira and J. C. Ferreira, "Reconfigurable NC-OFDM Processor for 5G Communications," in *Embedded and Ubiquitous Computing (EUC), 2015 IEEE 13th International Conference on*, Oct 2015, pp. 199–204.

[9] P. Boopal, M. Garrido, and O. Gustafsson, "A reconfigurable FFT architecture for variable-length and multi-streaming OFDM standards," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2013, pp. 2066–2070.

[10] G. Wang, B. Yin, I. Cho, J. Cavallaro, S. Bhattacharyya, and J. Takala, "Efficient architecture mapping of FFT/IFFT for cognitive radio networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 3933–3937.

[11] C. Vennila, G. Lakshminarayanan, and S.-B. Ko, "Dynamic Partial Reconfigurable FFT for OFDM Based Communication Systems," *Circuits, Systems, and Signal Processing*, vol. 31, no. 3, pp. 1049–1066, Jun. 2012. [Online]. Available: http://link.springer.com/article/10.1007/s00034-011-9367-9

[12] P.-A. Hsiung, M. D. Santambrogio, and C.-H. Huang, *Reconfigurable System Design and Verification*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2009.

[13] L. Crockett, R. Elliot, and M. Enderwitz, *The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC*. Strathclyde Academic Media, 2014. [Online]. Available: http://books.google.pt/books?id=9dfvoAEACAAJ

[14] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965. [Online]. Available: http://www.jstor.org/stable/2003354

[15] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, ser. Signals and communication technology. Springer, 2004. [Online]. Available: https://books.google.pt/books?id=Uj7-m45kRtUC

[16] I. Cho, T. Patyk, D. Guevorkian, J. Takala, and S. Bhattacharyya, "Pipelined FFT for wireless communications supporting 128-2048 / 1536 -point transforms," in *2013 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2013, pp. 1242–1245.

[17] *UG1037 - Vivado Design Suite: AXI Reference Guide*, Xilinx Inc., June 2015.

[18] *IEEE Std 802.22^{TM}-2011: Standard for Local and metropolitan area networks - Specific requirements - Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Policies and procedures for operation in the TV Bands*, IEEE 802.22 Working Group on Wireless Regional Area Networks Std., 2011.