UNIVERSITY OF L'AQUILA

DEPARTMENT OF INFORMATION ENGINEERING,
COMPUTER SCIENCE AND MATHEMATICS

PH.D. PROGRAM IN ICT - SYSTEMS ENGINEERING, TELECOMMUNICATIONS AND HW/SW PLATFORMS

XXXII CYCLE

DESIGN AND RECONFIGURABLE FPGA IMPLEMENTATION OF SATELLITE DIGITAL TRANSPARENT PROCESSORS

SSD - ING-INF/01

PH.D. STUDENT
Giuseppe Marini

COURSE COORDINATOR
Prof. Vittorio Cortellessa

ADVISOR
Prof. Marco Faccio

CO-ADVISOR
Prof. Fortunato Santucci

A.A. 2018/2019

Sommario

Nel contesto delle comunicazioni satellitari, un ruolo fondamentale è svolto dal processore di bordo. In particolare, per quanto riguarda lo sviluppo di transponder satellitari, si è passati da transponder bent-pipe analogici a quelli rigenerativi digitali. Recentemente, grande interesse stanno riscuotendo una nuova generazione di on-board processor chiamati processori semi-trasparenti. Tali transponder si compongono di una sezione digitale trasparente, chiamata Digital Tranparent Processor (DTP), dedicata al routing del traffico dati, e da una sezione rigenerativa deputata al media access control. Questa divisione conferisce ai processori semi-trasparenti la capacità di riconfigurare in maniera dinamica i piani di connettività e di adattarsi all'evoluzione degli standard di comunicazione. In particolare, per quanto riguarda il DTP, il suo funzionamento permette di disaccoppiare la progettazione della sezione di bordo del satellite dagli standard di comunicazione del livello fisico.

In questo contesto, un grande rilievo assume la necessità di definire criteri di progettazione della sezione digitale trasparente che tengano conto della complessatà hardware e i vincoli di link-budget. Alla luce di ciò, è stato definito un modello equivalente di rumore che caratterizza i comportamenti non ideali del processore di bordo. Il modello si basa su una definizione estesa di figura di rumore che consente di tener conto, nel link-budget, del comportamento non ideale (e.g. errori di quantizzazione e di arrotondamento e distorsioni lineari) dei vari blocchi del DTP.

In questo lavoro di tesi vengono forniti i seguenti contributi: 1) un calcolo esplicito della complessità implementativa di un DTP e 2) un metodo per realizzare un intero flusso di progettazione che parte dai requisiti di sistema e porta alla definizione dettagliata delle componenti hardware (HW) dei vari blocchi che compongono il DTP. Attraverso esempi numerici, si dimostra che, dato un requisito di link-budget, si osservano differenze significative nella complessità HW con differenti scelte di progettazione. Inoltre, grazie all'implementazione dell'intera catena di elaborazione del DTP su FPGA, viene presentata un'ulteriore estensione dell'approccio modellistico al fine di incorporare il consumo energetico delle architetture HW.

Summary

In the context of satellite communications, a key role is played by the on-board processor. In particular, the development of satellite transponders have evolved from the analog bent-pipe transponders to the digital regenerative ones. Recently, a new generation of transponders, namely semi-transparent transponders, have been emerging. The core components of these transponders are i) a digital transparent section, namely Digital Transparent Processor (DTP), which is devoted to the routing of the traffic data, and ii) a regenerative section that is devoted to the media access control. In this way, the dynamic reconfiguration of connectivity plans and the adaptation to evolving communications standards, are achieved by the semi-transparent transponder. Regarding the DTP, this section can efficiently decouple the on-board design from the physical layer communications standard.

In the above context, a great relevance is taken by the need to define design criteria of the transparent digital section that take into account the complexity of the hardware and the link-budget constraints. In this perspective, an equivalent noise model has been recently defined by our research group in order to characterize the non-ideal behavior of the on-board processor. The model relies on an extended notion of noise figure that allows a radio link-budget to natively incorporate the non-ideal behavior (e.g., quantization and rounding errors and linear distortions) of the various DTP segments.

In this thesis work the following research advances are provided: 1) an explicit computation of the implementation complexity of a DTP and 2) a methodology to carry out a complete design flow that moves from system requirements and brings to a detailed definition of digital hardware (HW) components in the DTP. Through numerical examples we demonstrate that, given a requirement on the overall link budget performance in contexts of practical interest, significant differences in HW complexity are obtained for different design choices. Moreover, having also completed a full FPGA implementation of a whole DTP chain, a further extension of the modelling approach in order to incorporate power consumption of HW architectures is presented. Results are provided as obtained for different configurations, requirements and design objectives.

Contents

Chapter 1

Introduction and Background

1.1 Overview and motivations

Within the rapidly evolving track of fifth generation (5G) wireless technology, huge efforts have been made in the very latest years to incorporate Satellite Communications (SatCom). Indeed, in this global frame SatCom can provide a valuable resource to extend and complement terrestrial networks both in terms of throughput and global connectivity [1] due to its capability of broadcasting telecommunication services to wider geographical areas and delivering broadband connectivity to populated remote regions, which are typically inaccessible or under-served by the terrestrial communication infrastructures. In other sectors such as aeronautical, maritime, military, rescue and disaster relief the SatCom is the only viable option. Until the fourth generation (4G) of mobile cellular communication systems, the satellite mobile system have developed independently of the terrestrial system. In the 5G perspective the SatCom has been recently depicted to be integrated with other networks and will be not a stand-alone network that provides 5G services. The third generation partnership project (3GPP) has so far identified the role of the satellites in the 5G scenarios. The three main roles identified by 3GPP for satellites in the 5G eco-system are [2–4]:

- Fostering the roll-out of 5G service in un-served areas that cannot be covered by the terrestrial 5G network.
- Reinforcing the 5G service reliability by providing service continuity for M2M/IoT devices or for passengers on board moving platforms.
- Enabling 5G network scalability by providing efficient multi-cast/broadcast resource for data delivery towards the network edged.

Historically, at the same time as the first cellular operators providing 1G analogue services, the first major satellite operator, Inmarsat, using the L band and

global beam coverage satellite, provided low data rate services to the maritime market of ships. In the mid-1990s, several regional geostationary earth orbit (GEO) satellite systems emerged focusing on land vehicles while using both the Ku and L bands. Research activities in the late 1980s and early 1990s focused on non-GEO constellations and resulted in the proposal of medium earth orbit (MEO) and LEO satellite system. From the first generation GEO satellites with 5-10 spot we moved to satellite systems with around 100-200 spot.

Looking toward the future to 2020–2025, there will be a trend to larger and more powerful GEO satellites for example: ViaSat-2, Inmarsat's Global Xpress (GX) network, Intelsat's EpicNG platform. On the other side, also the mega-constellations of low Earth orbit (LEO) satellites will play a crucial role. In this context we can take into account, for example: LeoSat, the OneWeb system, the SpaceX Starlink satellite system [5]. The integration of GEO High Throughput Satellites (HTS) with the terrestrial systems will provide a global large-capacity coverage. However, this comes with the challenge of a large propagation delay. Mega-LEO constellations, which are LEO systems consisting of hundreds of satellites, can circumvent this issue and it has recently received significant attention. Mega-LEO constellation can be used to provide LTE broadband services to areas that are not connected to a terrestrial infrastructure as demonstrated in [1,6]. Nevertheless, the large relative speed of the LEO satellites requires frequent handovers and compensation of large Doppler shifts. Furthermore, Inter-satellite links (ISL) are likely envisaged to avoid a pure "store-and-forward" approach, while supporting delay sensitive services and applications and maintaining a limited number of terrestrial gateways.

As a large number of small size, cost-efficient LEO satellites are involved, an efficient design and use of on-board resources becomes mandatory. For instance, enabling single-hop peer-to-peer communications reduces the capacity demand on feeder links, but requires larger on-board processing.

When on-board transponders are specifically considered three different categories can be taken into account:

- Bent-pipe.
- Digital transparent.
- Regenerative.

The bent-pipe architecture is the classical satellite architecture. It can be described as filter, amplify and forward architecture which receives the signal, amplifies it, converts it to the respective downlink frequency and amplifies it with the power amplifier to downlink the signal to earth. All the higher level functions which provide the connectivity must be provided by complex gateway on earth.

Digital transparent processors are used in satellites to introduce a higher level of flexibility. This architecture adds a processor in the signal path to filter and switch/route the signals to the required beams. It does not regenerate the signal so it could also be used for analog signals. This architecture brings a lot of additional flexibility in terms of signal processing capabilities such as filtering and routing. It is more complex and more power consuming as the bent-pipe architecture.

Architectures based on a regenerative processor add additional features as the received signals can be regenerated, and therefore, some additional gain can be achieved within the overall link-budget. This architecture leads to higher power consumption and introduces additional complexity in designing the signal processing. Along the last decade, a new type of on board architectures have received major attention. This architecture known as digital semi-transparent can be considered halfway between digital and regenerative transponder. This kind of architectures, also known as translucent architectures, are emerging as a viable alternative to provide broadband connectivity in modern network topologies with larger users' populations and a variety of requirements in terms of bandwidths and QoS, while maintaining the payload complexity affordable [7, 8].

The digital semi-transparent payloads flexibility in the frequency planning, that also enables dynamic reconfiguration of connectivity plans and the adaptation to evolving communications standards, are achieved by separating traffic processing and traffic control sections.

The payloads are based on a Digital Transparent Processor (DTP) that can efficiently decouple the on-board design from the physical layer communications standard, the frequency and connectivity plan; therefore, payload operations can be dynamically updated during the satellite lifetime. This perspective is extremely appealing in many application domains [9]. A DTP-based satellite payload is composed of analog receiving and transmission chains with a fully digital but non regenerative chain between them. As a whole, it can be regarded as a hybrid analog-digital on-board chain. Both modelling and design approaches have so far almost missed the ability to capture in an adequate way these important features. In the digital chain included in the DTP the analysis and synthesis processes involved in on-board channelization are of major concern, and various architectures have been proposed in the literature. In [10] the implementation for both discrete filter bank and DFT (Discrete Fourier Transform) modulated oversampled method has been analyzed. In [11] the benefits provided by a polyphase network and DFT solution in terms of flexibility and complexity have been assessed, also in the context of tree-structured filter banks [12]. Furthermore, once an architecture is selected, different implementations can be envisaged, that encompass completely parallel solutions and pipelined ones. The pipelined solutions are considered rather than full parallel implementations in order to reduce overall HW complexity (especially in terms of the amount of multiplication and sum blocks).

In order to address the design issues related to the double nature of the transparent section, in [13], the characterization of both linear distortion and rounding noise terms has been proposed for a satellite DTP. In [14] a preliminary description of the DTP HW architecture has been presented in order to address the problem of computation of the DTP hardware complexity.

1.2 Outline and Contributions

In this section, the contents of the thesis and the main contributions are outlined. In [P1] a first validation of the design choices have been obtained through an FPGA-based prototype benchmark. In [P4] the modelling and design framework is fully presented, and it basically relies on an extended notion of noise figure to incorporate imperfections of digital sections into the link budget. In [P3] the theoretical framework for DTPs is further extended to incorporate technology-related elements into the analysis and design process. The complexity, previously based on elementary blocks [P4], is expressed in terms of hardware primitives. A particular goal and related novel contribution is to estimate the power consumption and include it both in performance evaluation and design approach. In order to obtain expressions for hardware complexity and power estimation that hold true for currently available commercial FPGAs, a Virtual FPGA is defined. The DTP hardware components are mapped on the Virtual FPGA and the hardware complexity and the power estimation are computed and compared to simulation results carried out by Xilinx tools on a commercial FPGA.

Chapter 2 provides the description of the architecture of each block within the DTP chain. For each DTP block the functional description and the hardware design are provided. The material presented in this chapter is based on the journal [P4], while a short description of the processing chain blocks is given also in the conference paper [P2].

In Chapter 3, an equivalent noise model for the specific DTP architecture that composes the satellite transparent transponder is provided. The material presented in this chapter is based on the journal [P4].

Chapter 4 provides the hardware complexity analysis of the whole DTP chain. An analytical model to estimate the power consumption of the specific DTP is presented. The material presented in this chapter is based mainly on the conference publications [P2,P3] and on the journal [P4].

Chapter 5 includes the hardware complexity analysis of the basic building

blocks and of the whole DTP chain and the preliminary power estimation results. The material presented in this chapter is based mainly on the conference publications [P1,P3] and on the journal [P4].

Conclusion are drawn in Chapter 6.

Chapter 2

DTP Architecture: From Functional to Hardware Design

In this chapter, the architecture of the DTP chain is presented. Starting from the reference system scenario, the digital elaboration blocks are designed. For each elaboration block, the functional description and the hardware design are provided. The elaboration chain is implemented in the VHDL language.

2.1 DTP Requirements

According to the reference system scenario described in [15], which was motivated by the European Space Agency (ESA) project, the coverage of the european geographic area is implemented by partitioning the reference area into 79 beams. In particular, the Ka-band is considered, with a 500 MHz uplink bandwidth in the range 29.5-30.0 GHz and a 500 MHz downlink bandwidth in the range 19.7-20.2 GHz. A multi-beam coverage and a four frequency reuse pattern is assumed; therefore each beam is assigned a 125 MHz bandwidth (with single polarization) or a 250 MHz bandwidth (with double polarization). The coverage pattern is generated on board by adopting a beamforming scheme that involves an analog beamforming network and an Array-Fed Reflector (AFR) with a seven Feeds Per Beam scheme [15]. Fig. 2.1 shows the architecture of the transparent transponder that includes an analog front-end for beam forming (Beam Forming Network), the next part of the RF chain (Low Noise Amplifier, Mixer and Automatic Gain Control, etc.), and the digital core that is representative of the DTP. For each beam a DTP chain is considered. The digital section is composed by a Digital RX Chain, a Switch and a Digital TX Chain. The Switch performs the routing intra-beam and inter-beam. The Digital RX Chain performs the decomposition of the uplink signal into J independent signal. The Digital TX Chain realizes the complementary

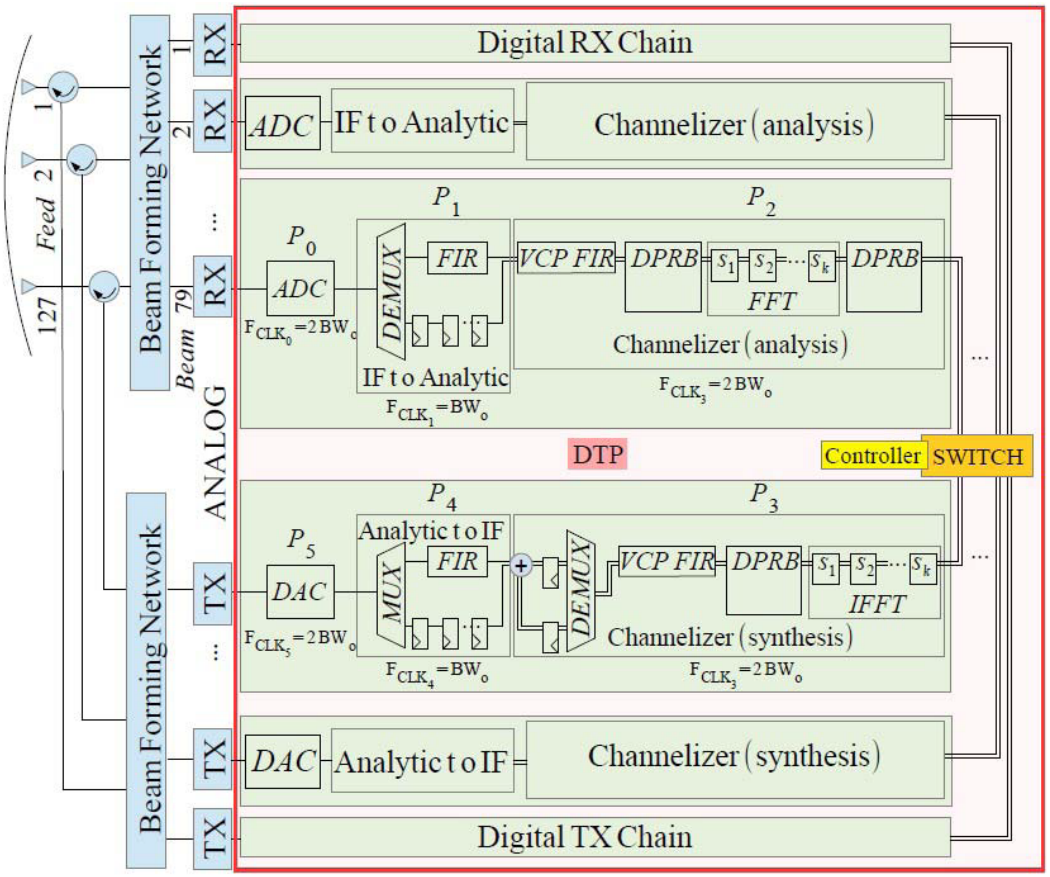

Figure 2.1: Block diagram of the transponder architecture.

operation. In the present manuscript the on-board switch section is considered as a transparent element from the signal processing perspective therefore it was not subject of analysis. DTP synthesis and analysis channelizers [16] are designed as non-critically sampled and endowed with a sum-to-one feature. The sum-to-one feature is described in Fig. 2.2. The processed bandwidth is split in several Elementary Switched Bandwidths (ESBWs) that may be independently managed in order to configure the best frequency planning. In fact, the ESBW is the smallest unit that can be allocated, alone or in combination with adjacent ESBWs, to host on-board processing of communication carriers. The adjacent ESBWs are filtered with a sum-to-one transfer function so that, if they are contiguously switched from the uplink input to the downlink output, no linear distortion occurs at their intermediate boundaries when considering a user channel whose bandwidth spans over multiple ESBWs. Fig. 2.2 further illustrates a possible trade-off that can be made between lost bandwidth resources and processor hardware complexity: if smaller size ESBWs are used, the transition bandwidth - which is unavoidably

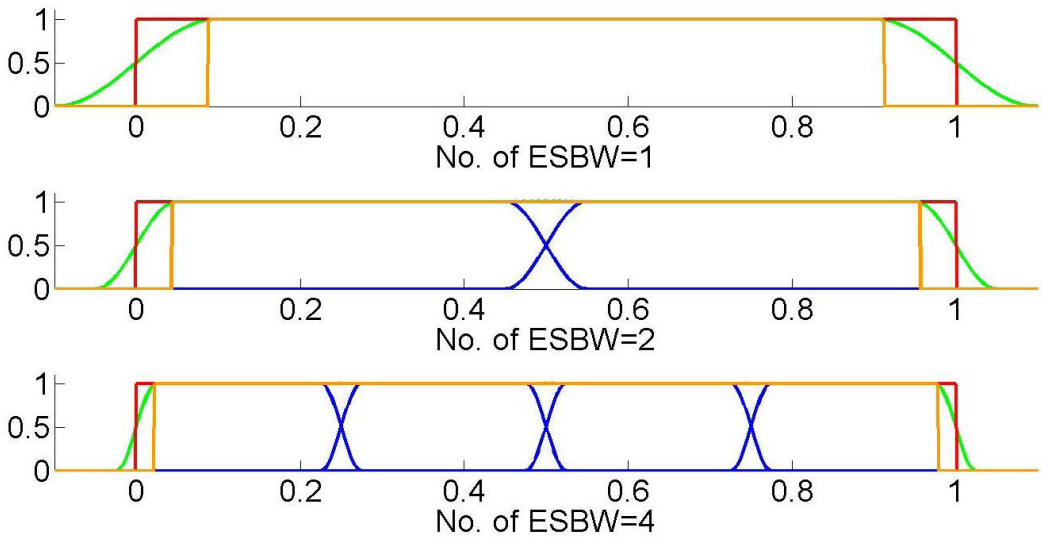

Figure 2.2: Reduction of edge transition-bandwidth waste with increase of J.

lost at user channel edges - becomes smaller and user channels can be set closer with minimum guard bands. However, this requires a larger number of channels J and increased hardware complexity.

The DTP model is composed of an ADC, a "IF to Analytic" block, a channelizer of analysis (working on J-channels), the switch, and their dual blocks. Fig. 2.3 shows the DTP processing chain for a simplified scenario, where the routing from different beams (and ADCs) is not considered. The processes within the chain are labeled from P_0 (i.e. the ADC) to P_5 (the Digital to Analog Converter, DAC). When considering the internal structure of each stage, three basic building

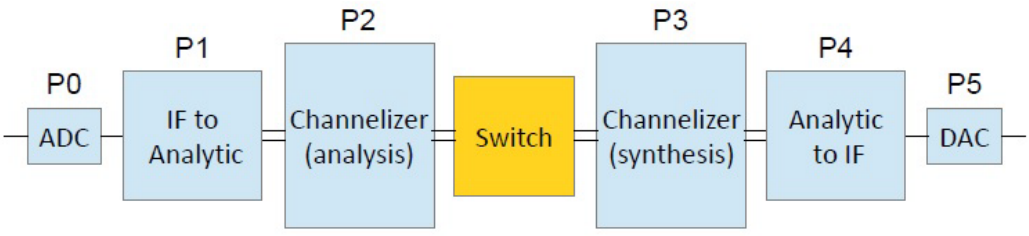

Figure 2.3: DTP processing chain.

blocks have to be considered to implement input channels' separation and recomposition of the output pass-band signal: 1) FIR (Finite Impulse Response) filter elements, implemented in direct form, 2) FFT butterfly structures and 3) Dual Port RAM Buffers (DPRB). The architecture also includes some Saturation and

Rounding Blocks (SRB) that are necessary to maintain a limited-size fixed-point arithmetic; in some cases these blocks can also perform the shift in the binary words in order to apply a scaling of signal amplitude.

2.2 Analog-to-Digital Converter

A signal with maximum spectral extension equal to $2f_0$ and spectrum centered on the Intermediate Frequency f_0 (I.F.) is assumed as the input of the ADC. This signal provided by the analog RF section of the on-board receiver is sampled, in accordance with the Nyquist Theorem, at the frequency $F_s = 4f_0$ and quantized and encoded in two's complement through n_s bits.

2.3 IF to Analytic

The "IF to Analytic" block then performs the extrapolation of the analytical signal from the IF input digital signal. The elaboration is composed by a complex filtering process followed by a decimation process performed with a conversion factor equal to 2 (the output sample period is $T_{o1} = 1/(2f_0)$). A prototype low-pass FIR filter, with order $(N_1 - 1)$ and pass-band approximately equal to f_0, is shifted in frequency in order to obtain the complex filter with the pass-band centered in f_0. In order to obtain an efficient synthesis of the block, the polyphase implementation of the decimator can be considered. This solution is based on the definition of two polyphase components both for the input signal ($S_0^{(0)}, S_1^{(0)}$) and for the complex FIR filter response ($\underline{H}_0^{(1)}, \underline{H}_1^{(1)}$). As a final result, for a generic implementation of the block the decimated output signal can be expressed as:

$$\underline{S}^{(1)} = (S_0^{(0)} * \underline{H}_1^{(1)} + S_1^{(0)} * \underline{H}_0^{(1)}) \tag{2.1}$$

The prototype low-pass filter $H^{(1)}$ must be designed at the input sampling frequency $F_{s1} = 4f_0$; therefore, the shift in frequency can be obtained by taking the product between the coefficients and the exponential term $e^{\frac{jn\pi}{2}}$. The complex filter impulse response can be expressed as:

$$\underline{h(n)}^{(1)} = h(n)^{(1)}(\cos(\frac{\pi}{2}n) + j\sin(\frac{\pi}{2}n)) \tag{2.2}$$

When computing the polyphase components in this case, it can be observed that one of these is only composed by the real part and the other one only includes the imaginary part:

$$\underline{h(n)}_0^{(1)} = h(2n)^{(1)}\cos(\pi n) \tag{2.3}$$

$$\underline{h(n)}_1^{(1)} = jh(2n+1)^{(1)}\cos(\pi n) \tag{2.4}$$

The prototype filter is designed at the input frequency ($F_{s1} = 4f_0$) with the transition band centered in f_0; as a consequence a Half-Band filter can be considered: its even coefficients are all null, with the exception of the coefficient h(0) that is equal to 0.5. As the process of extrapolation of the analytical signal implicitly induces an amplitude scaling of the output signal components (i.e. $(1/2)x_I(t); (1/2)x_Q(t)$), an amplification factor of 2 is introduced in filter's coefficients ($A_1 = 2$). Therefore, the non zero coefficient of the even component takes a unit value: the implementation of the corresponding polyphase branch does not require any calculation, but rather a status register for the management of the delay. Similar remarks are given in [17], where the "IF to Analytic" block is called Digital Anti-aliasing Filter (DAF). Figure 2.4 shows the complete and detailed Register Transfer Level (RTL) architecture of the block.

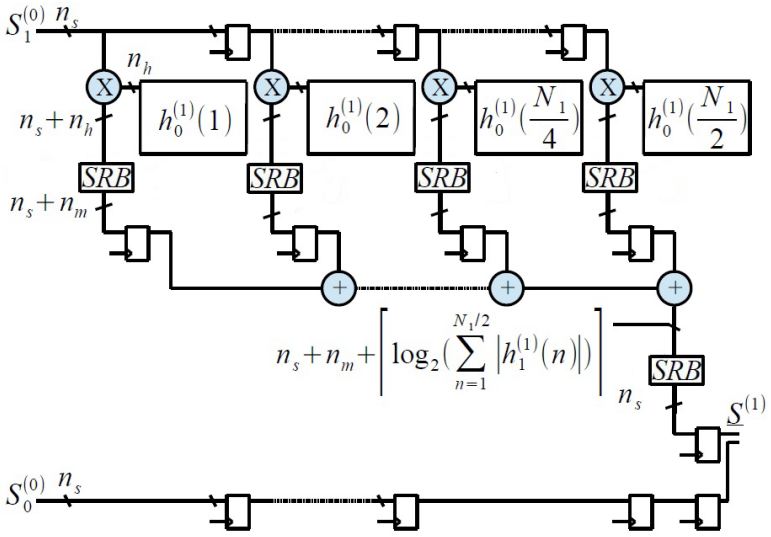

Figure 2.4: RTL diagram of the "IF to Analytic" process.

2.4 Analysis Channelizer

Within the analysis channelizer the analytic signal band is decomposed into J independent complex signals whose spectrum is shifted towards a low frequency range. A "non critically sampled" solution (i.e. with an oversampling factor equal to 2) based on a polyphase network and an FFT block can be considered among the viable solutions to implement the channelizer. This architecture, that is discussed in section 7.2.4 of [18], is based on the concept of "extended set of polyphase

filters" and on the definition of the J polyphase branches, that are in turn interpolator filters with oversampling factor I ($I = 2$ in our case). By considering a decimation factor equal to M (M is equal to $J/2$) the polyphase components of the prototype FIR filter can be computed as:

$$h_p(m) = h(mM - p) \tag{2.5}$$

for p ranging from 0 to $K-1$. In the classical definition of polyphase components, where K is equal to the decimation factor M, all the prototype filter coefficients are split into M different and completely independent components.
However, the parameter K expresses the number of extrapolated channels ($K = J$): thus, for the "non critically sampled" implementation, the number of polyphase components is as large as twice the decimation factor used to obtain them (i.e. $K = I\ M = 2\ M$). Therefore, not all the $2\ M$ components can be completely different and independent of each other. The $(i + M)^{th}$ component (with i from 0 to $M - 1$) differs from the i^{th} only for an additional initial coefficient that takes null value. In other words, the $(i + M)^{th}$ component is a delayed version of the i^{th} one [18].
Therefore, in the "non critically sampled" channelizer only $J/2$ polyphase components can be extrapolated from the prototype filter. These components are directly used in the first $J/2$ branches of the polyphase network, while a delayed version of these components (representative of the set extension) are used in the remaining branches.

These delays can be introduced at the input of the filter blocks [19], or at the output of the blocks. Fig.7.15 of [18] shows the parallel implementation of the analysis channelizer, where each branch is an interpolator that can be further decomposed in a polyphase architecture. In this case each branch is implemented through two parallel sub-branches with the same input and with the outputs that must be serialized in a single data path with double data rate. From the generic polyphase component of the filter $H_i^{(2)}$ two components are further extrapolated ($H_{i,k}^{(2)}$ with $i = 0, 1, ..., J/2 - 1$ and $k = 0, 1$). As a matter of fact, in order to save hardware resources the parallel architecture can be replaced with an equivalent solution, based on a recursive utilization of a single elaboration branch, that in [20] is called "Variable Coefficient Polyphase Filter" (VCPF). Figure 2.5 shows the RTL diagram of the equivalent polyphase network, where input and output signals are denoted as complex signals. The status register within the VCPF allows to store and isolate the needed samples of the J polyphase components of the input signal. To perform this task, for each block of multiplication and accumulation (i.e. for each tap), a First In First Out (FIFO) register composed by J elements is allocated. Each polyphase component of the input signal is processed with the right polyphase component of the filter. All the polyphase components of the

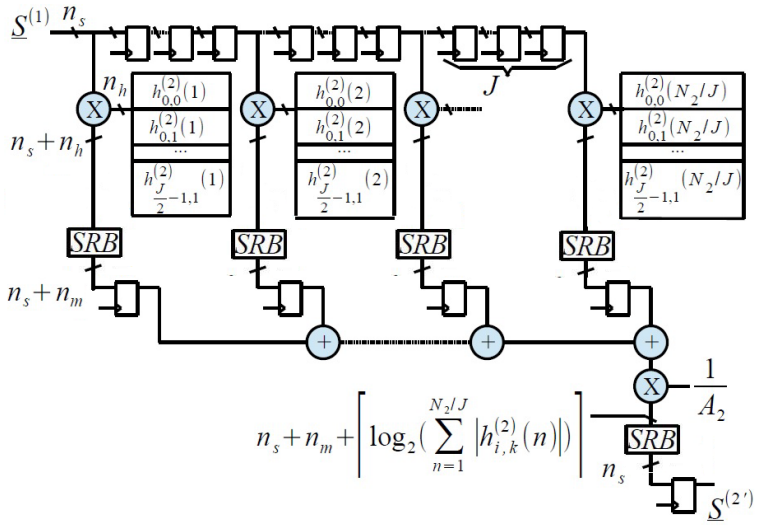

Figure 2.5: RTL view of the VCPF used in the analysis channelizer.

filter are stored within the VCPF and are used in cyclic way. Whenever a new input sample arrives, the status register is updated and the new content is used to produce two output samples. The elaboration frequency is as large as twice the input throughput and the two output samples are derived from the same content of the status register, although they are processed through the two polyphase sub-components ($H_{i,k}^{(2)}$ with $k = 0, 1$). These two samples are processed sequentially; however, within the serial output they should be spaced by the $J - 1$ samples related to the other branches. For this reason, a buffer needs to be allocated after the VCPF to perform the reorganization of the data in order to provide the right input sequence to the FFT block. This buffer must support the reorganization of $2J$ complex-words. To perform this task, an amount of memory as large as twice the reordering length must be allocated. This memory may be structured as a Dual Port RAM Buffer (DPRB), wherein writing and reading to/from different locations can be performed concurrently and in an independent way. The buffer may be also used to manage the introduction of the delays needed in the stream of samples of the last $J/2$ branches of the polyphase network.

Among the viable alternatives the FFT block can be obtained through a Decimation In Frequency Radix-2 algorithm implemented by a pipelined feedback (FB) architecture. Details on the hardware implementation of this architecture are given in [21] for a number of points (i.e. channels) equal to 32. By generalizing this architecture for $J = 2^k$ channels, we can consider that the polyphase network and the RAM buffer are followed by a FFT block that is implemented through $log_2 J = k$ cascaded stages. A single stage is composed by a FIFO memory block,

a butterfly unit and a multiplier. The adders within the butterfly network and the external multiplier are complex [22]. According to Fig. 2.6 of [21], Figure 2.6 shows the RTL diagram of a generic FFT stage, wherein the complex adders and the complex multiplier are explicitly detailed. The length of the FIFO depends

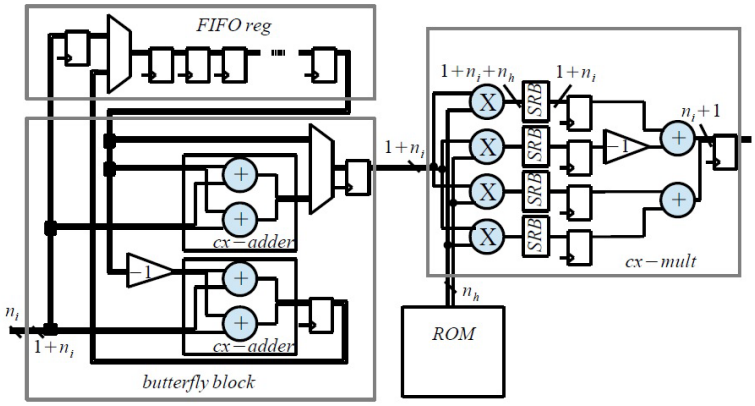

Figure 2.6: RTL diagram of the butterfly stage.

both on the total number of the J FFT points and on the index of the stage (i-th stage with $i = 1, 2, ..., log_2 J$). In particular, the FIFO of the i-th stage is composed by $J/2^i$ registers. The butterfly unit is composed by two complex adders (each of them is composed by two adders), two multiplexers and all the typical connections required to implement the radix-2 butterfly element. In order to keep the response delay time of the butterfly unit at a limited level and then maximize the elaboration frequency clock, it is possible to introduce a buffer on the complex adders' output. This yields an appropriate management of the internal links and uniform latencies. It is worth of mention that all the considered memory elements are intended to store complex sequences and the registers have therefore multiplicity equal to two. Finally, the complex multiplier is composed by four multipliers followed by two adders. Also in this case a buffer level is inserted to maximize the elaboration frequency clock both in the multiplier and adder levels.

A buffer element is finally required at the output of the FFT block to reorganize the order of the samples in the sequence. Indeed, if the input of the radix-2 FFT is in the "natural order", the output result is in a "reverse order"; therefore, in order to obtain again the right sequence, a buffer manipulation must be applied [21]. For the FFT blocks this rearrangement involves a number of complex-words equal to the number J of channels, and the DPRB must then store $2J$ complex-words. Figure 2.7 shows the block diagram of the overall analysis channelizer in the equivalent non parallel architecture.

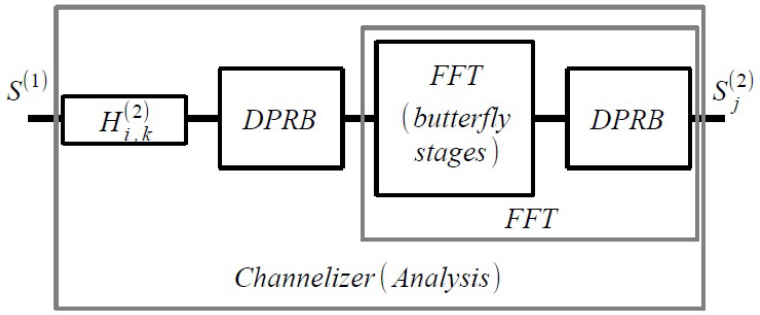

Figure 2.7: Block diagram of the analysis channelizer in the non parallel architecture.

2.5 Synthesis Channelizer

In the synthesis channelizer the J input signals are combined in order to obtain the output analytic signal. The channelizer is still in the "non critically sampled" version with oversampling factor equal to 2, and a parallel implementation of the block can be derived from Figure 7.16 of [18]. Also in this case the architecture is based on the "extended set of polyphase filters"; in particular, $J/2$ components are extrapolated from the prototype filter and are directly used in the first $J/2$ branches. A delayed version of these components, defined with a positive time delay, are used in the remaining $J/2$ branches. In order to meet the causality property, a delay in each branch can be introduced, so that the first $J/2$ branches of the polyphase network operate on a delayed version of the input signals, while the other branches operate directly on the output signals of the IFFT block. In turn, each branch of the polyphase network is a decimator, with decimation factor equal to 2, that can be implemented through a further polyphase decomposition. Such a decomposition can be done through the allocation of two additional branches that operate on two polyphase components of the branch signal: the overall output can be obtained by the sum of both processing components. In an equivalent way, the branch can be implemented with a unique "variable coefficient polyphase filter". An additional element then follows for the composition of the overall result given by the sum of the two sub-branches elaborations.

Moreover, in order to save hardware resources, the architecture can be serialized by considering a single VCPF: the latter one is not only able to manage the decimation defined for a single branch, but also the process related to the polyphase network. As shown in Figure 2.8, the block that performs this task is a simple VCPF with only one modification in the status register: $2J$ complex samples have to be stored through a FIFO for each tap (while J samples are involved in the analysis block). Indeed, the status register must be able to manage the J

polyphase components related to the J branches of the channelizer and also the two components of the decimation process for each branch. The proper routing

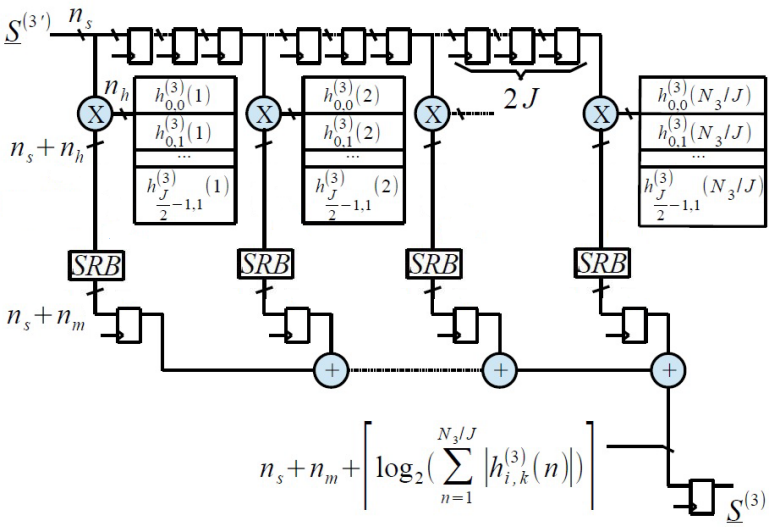

Figure 2.8: RTL view of the VCPF used in the synthesis channelizer.

of samples within the status register must be guaranteed by the Dual Ports Ram Buffer that is interposed between the IFFT block and the VCPF: the buffer also manages the delay element defined in the polyphase network and the output of the IFFT block that is not in the natural order. Moreover, a memory space ables to store $4J$ complex samples is allocated to the DPRB to provide the required interface. Figure 2.9 shows the equivalent implementation of the synthesis channelizer, that is composed by the in place radix-2 IFFT block, the DPRB, the VCPF and the elements for the composition of the overall result of the decimation branches. It can be noticed that, while the round-off of the accumulation process can be avoided in the VCPF, it is applied to the output of the final sum as the same number n_s of bits used for the input signal are used to represent the output signal (see Figure 2.9). The implementation of the IFFT is coincident with the implementation of the FFT block used in the analysis channelizer: therefore, the IFFT block is composed of $log_2 J = k$ stages of butterfly elements and complex multipliers in the same architecture shown in Figure 2.6. The only difference is in the sign of the twiddle factors.

2.6 Analytic to IF

The "Analytic to IF" block processes the output signal of the synthesis channelizer in order to obtain the discrete IF signal. For this purpose the condition of

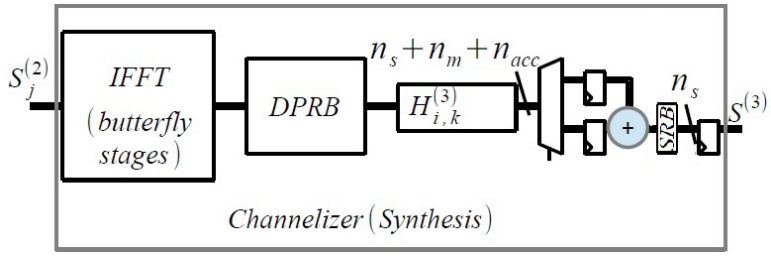

Figure 2.9: Equivalent implementation of the synthesis channelizer.

oversampling is restored on the channelizer output through an interpolator operating with conversion factor equal to 2, that can be implemented in a polyphase. The Analytic signal is then obtained and, by taking its real part, the discrete IF signal is extrapolated.

By considering that the input signal is represented with sampling frequency $1/T_{o3} = 2f_0$, the frequency is increased to $1/T_{o4} = 4f_0$ after interpolation and coincides with the design frequency $F_{s4} = 4f_0$ of the prototype low-pass FIR filter of the interpolator. The filter has order $(N_4 - 1)$ and pass-band approximately equal to f_0. The interpolator operates with a complex version of the prototype FIR filter that is shifted in frequency in order to obtain a pass-band centered in f_0. By resorting to the z-transform representation, the process implemented by the block can be described through the following general expression:

$$S_0^{(4)} = \Re[\underline{S}_0^{(4)}] = (H_{0\Re}^{(4)} S_{\Re}^{(3)} - H_{0\Im}^{(4)} S_{\Im}^{(3)}) \tag{2.6}$$

$$S_1^{(4)} = \Re[\underline{S}_1^{(4)}] = (H_{1\Re}^{(4)} S_{\Re}^{(3)} - H_{1\Im}^{(4)} S_{\Im}^{(3)}) \tag{2.7}$$

wherein the real output signal is described through two polyphase components obtained from the real parts (subscripts $\Re$) and the imaginary parts (subscripts $\Im$) of the input signal and of the polyphase components of the filter response. It can then be observed that in this specific case the complex filter has the same characteristics of the filter used in the block "IF to Analytic"; therefore, the architecture of the "Analytic to IF" block can also be simplified. Indeed, the exponential term $e^{\frac{jn\pi}{2}}$ that induces the shift in frequency of the prototype filter produces the cancellation of some portions of the complex filter components. In particular, it results $H_{0\Im}^{(4)} = H_{1\Re}^{(4)} = 0$ and then:

$$S_0^{(4)} = \Re[\underline{S}_0^{(4)}] = H_{0\Re}^{(4)} S_{\Re}^{(3)} \tag{2.8}$$

$$S_1^{(4)} = \Re[\underline{S}_1^{(4)}] = -H_{1\Im}^{(4)} S_{\Im}^{(3)} \tag{2.9}$$

Furthermore, an amplification factor ($A_4 = 2$) is considered for compensating the amplitude scaling induced by interpolation. It can be further observe that also

in this case a Half-Band filter is obtained with even coefficients of null value and the only exception of the coefficient $h(0)$ that is equal to 1. Figure 2.10 shows the simplified RTL architecture of the block.

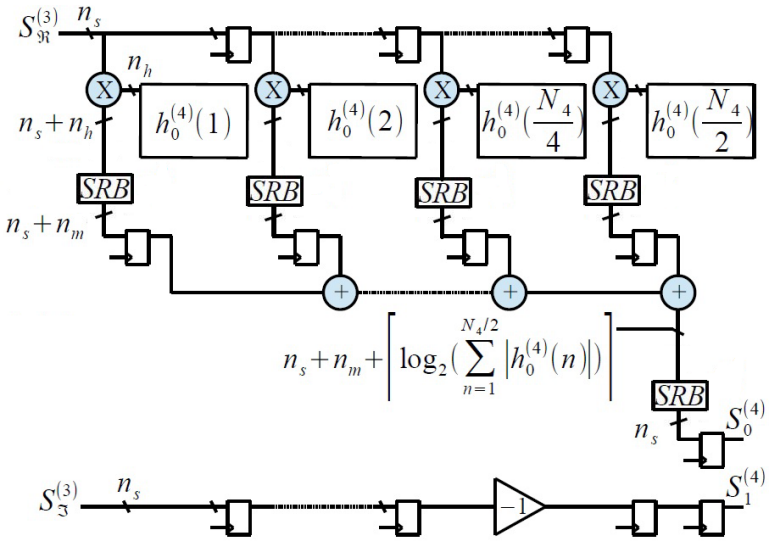

Figure 2.10: RTL diagram of the "Analytic to IF" process.

Chapter 3

Noise Model of the Digital Transparent Processor Chain

Motivated by the need of fully exploit the flexibility features of the semi-transparent payload, an important aspect that influence the DTP performance is presented in this chapter. While the processed bandwidth imposes directly the sampling speed of the digital section, the modulation and coding schemes impose a requirement for the signal-to-noise ratio that the link-budget must satisfy. The subject of this chapter concerns the evaluation of the DTP performances in term of noise contribution. Moving from [13], an extension of the noise model is presented. This noise model is able to support full exploration of the design alternatives. Furthermore, in order to overcome some initial assumptions, the extended model also allows to handle different values of the power spectral density across different ESBWs and more general constraints on pass-band and stop-band in filters' design.

3.1 An Extended Notion of Noise Figure

From the hardware architecture of the DTP described in the Chapter 2 the following sources of degradation can be modeled:

- Quantization errors caused by the Analog-to-Digital (ADC) conversion.
- Non idealities in filters' implementation, namely the adoption of finite length impulse responses due to windowing and the finite number of bits for representation of their coefficients; these effects can also be cast in terms of linear distortion with respect to a reference (ideal) behavior.
- Use of a fixed-point arithmetic in the registers operating within the whole processing chain.

In [13], the hybrid receiving chain is sketched like in Fig. 3.1 where the "Analog

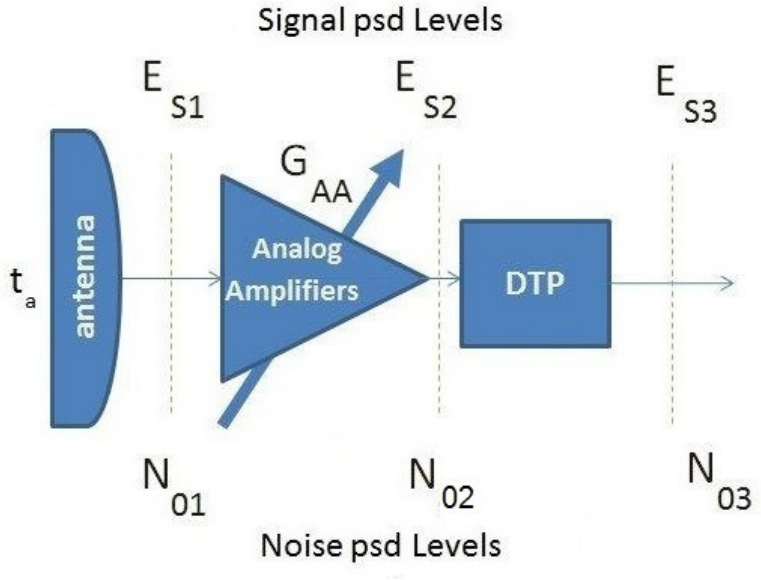

Figure 3.1: Signal and noise levels in the on-board receiving chain.

Amplifier" (AA) block is actually intended to denote the whole RF analog receiving front-end and typically includes an LNA (Low Noise Amplifier), a mixer, an AGC (Automatic Gain Control) stage and an AAF (Anti-Aliasing IF Filter). According to the AWGN channel model, the first noise contribution taken at the input of the AA block is characterized by a power spectral density $N_{01} = kt_a$, where t_a is the antenna (noise) temperature and k is the Boltzmann's constant. The next block AA is characterized by the equivalent noise temperature t_{AA} , the noise figure F_{AA} and the overall power gain G_{AA}.

In the DTP block, the sources of degradation can be characterized, according to a typical and widespread assumption, as AWGN components and the flat power spectral density can be expressed as:

$$N_{DTP} = N_{0R} + \delta^2 E_{S2} \tag{3.1}$$

where the term N_{0R} is used to denote the power spectral density related to the overall rounding and quantization noise along the whole DTP chain, and the second term is intended to account for the linear distortion noise in FIR filters implementation and is proportional to the useful signal power spectral density with a coefficient δ accounts for ripple in the FIR filters. By extending the approach recently devised in [23], [24] for quantization noise in ADC, in [13] a noise figure

F_{DTP} is defined as follows:

$$F_{DTP} = \frac{N_{01} + N_{DTP}}{N_{01}} \tag{3.2}$$

From system level performance requirements, in [13] is defined an upper bound for the DTP noise power spectral density that relies on the noise figure:

$$N_{DTP}^{MAX} = F_{DTP}\, N_{01} - N_{01} \tag{3.3}$$

When the impact of DTP internal blocks on the linear distortion noise is concerned, the blocks may be considered as sampling rate conversion blocks. In particular in [13] is derived the following general expression for the power spectral density of the related noise components at the output of each generic block k that applies both for interpolators and decimators:

$$N_{Lk} = c_k S_i \delta_k^2 F_{sk} T_{ok} \tag{3.4}$$

where c_k is a correction term that depends to the specific block, $1/T_{ok}$ is the sampling frequency at the output of the k-th block and F_{sk} is the filter design frequency. Furthermore, S_i is used to denote the (flat) signal power spectral density and $\delta_p = \delta_s = \delta_k$ accounts for ripple, which is assumed to contribute in the same way for both pass-band and stop-band regions of the filter design.

3.2 Analog-to-Digital Converter

According to [13], the quantization process among the various aspects that contribute to the characterization of analog-to-digital converters is consider. Since the dynamics of the input signal is much larger than the ADC quantization step, the quantization noise can be modeled as a white noise with zero mean and power spectral density $N_{R0} = (q_{s0}^2/12)T_{o0}$, that is superimposed to the useful signal. The quantization step $q_{s0} = V_{FS}/2^{(n_{s0}-1)}$ is computed from the full-scale voltage V_{FS}, the n_{s0} bits of the ADC encoding, and the sampling period $T_{o0} = 1/(4f_0)$. Therefore, the total noise contribution at the output of this stage can be expressed as:

$$N_{R0}^T = N_{R0}; \quad N_{L0}^T = 0 \tag{3.5}$$

when considering that the linear distortion term N_{L0}^T can be neglected.

3.3 IF to Analytic

In this stage let us consider a prototype filter with order $N_1 - 1$: the term N_{R1}, that describes the power spectral density of the noise due to round-off can

be written as:

$$N_{R1} = (\frac{N_1}{2}\frac{q_{m1}^2}{12} + a_1\frac{q_{s1}^2}{12})T_{o1} \tag{3.6}$$

where the output sample period is $T_{o1} = 1/(2f_0)$ and the quantization steps can be expressed as $q_{m1} = V_{FS}/2^{(n_{s0}+n_{m1}-1)}$ and $q_{s1} = V_{FS}/2^{(n_{s1}-1)}$ for the multiplications and accumulation results, respectively. The term a_1 accounts for round-off on the processing free branch: a_1 is equal to 1 if $n_{s0} \leq n_{s1}$ (the processing free branch is without round-off), otherwise is equal to 2. In those expressions n_{s0} denotes the number of bits of the input signal provided by the ADC, n_{m1} is the additional number of bits used for the products and n_{s1} is the number of bits for the output representation. As far as the linear distortion term is considered, the power spectral density S_i of the input signal and the related linear distortion noise at the output can be specified for the k-th ESBW and denoted as $S_i(k)$ for $k = 1, ...J$.

In the "IF to Analytic" block the noise resulting from the filtering process in the k-th ESBW is due to the linear distortion, that occurs because of the ripple (δ_{1p} denotes the maximum ripple excursion) in the pass-band ($A_1^2 S_i(k)\delta_{1p}^2$), and of the non-ideal signal cancellation in the stop-band (δ_{1s} is the maximum gain in the stop-band): the last term induces a cross-talk contribution (deriving from the $(J+1-k)$-th ESBW) that, due to the decimation process, is reported in the k-th ESBW. Therefore, for $k = 1, ...J$ the power spectral density of the noise due to the filtering process can be written as:

$$N_{L1}(k) = A_1^2\,[S_i(k)\,\delta_{p1}^2 + S_i(J+1-k)\,\delta_{s1}^2] \tag{3.7}$$

where the already introduced factor $A_1 = 2$ denotes the amplification used for the optimal coefficients representation. Finally, the overall noise power spectral density at the block output is expressed as:

$$N_{R1}^T = (A_1^2 N_{R0}^T) + N_{R1};\;\; N_{L1}^T(k) = N_{L1}(k) \tag{3.8}$$

3.4 Analysis Channelizer

The power spectral density of the rounding noise introduced within the channelizer is given by the sum of the contributions produced both in the equivalent polyphase network and in the FFT block. Within the polyphase network more round-offs processes are involved. The multiplication results are expressed with $n_{s1}+n_{m2}$ bits (with an associated quantization step equal to $q_{m2} = V_{FS}/2^{(n_{s1}+n_{m2}-1)}$) and also in the accumulation results of the J equivalent branches of the VCPF a round-off is performed in order to obtain a representation based on $n_{s2'}$ bits (that defines a quantization step equal to $q_{s2'} = V_{FS}/2^{(n_{s2'}-1)}$). By assuming for the various stages of the FFT a representation of the multiplication results based on

n_{s2} bits (that defines a quantization step equal to $q_{s2} = V_{FS}/2^{(n_{s2}-1)}$) an equivalent source of noise with power spectral density equal to $N_{R2-FFT} = J(\frac{q_{s2}^2}{3})T_{o2}$ must be considered. For a prototype filter with order equal to $N_2 - 1$ the channelizer round-off noise term can be expressed as:

$$N_{R2} = J(\frac{2N_2}{J}\frac{q_{m2}^2}{3}\frac{1}{A_2^2} + \frac{q_{s2'}^2}{6} + \frac{q_{s2}^2}{3})T_{o2} \tag{3.9}$$

The above expression takes into account a partial scaling of the amplification factor $A_2 = J$ used for the optimal coefficients representation and the decimation process that defines an output sample period equal to $T_{o2} = J/(4f_0)$. The scaling by a factor $2/A_2$ is applied after the accumulation process.
As already stated, the ripple in the pass-band filter response (δ_{p2} is the ripple bound) directly induces a distortion in the ESBW of interest, while the residual stop-band gain (δ_{s2} is the largest stop-band gain) is responsible for interference contributions from adjacent ESBWs. Furthermore, a noise contribution from adjacent ESBWs is due to the decimation process. In the "non critically sampled" operation considered for our channelizer, only the error produced in the $(J/2) - 1$ adjacent ESBWs is reported within the useful bandwith of a given ESBW. If $S_i(k)$ denotes the power spectral density of the input signal within the k-th ESBW (for k from 1 to J), the power spectral density of the noise to be considered at the stage output from the k-th band ($N_{L2}(k)$) is due to the distortion in the pass-band and to the error generated within the adjacent ESBWs with index equal to $((k+2\,n-1)\%J)+1$ (for n from 1 to $(J/2)-1$ and "%" that denotes the module operation, i.e. $(k + 2\,n - 1) - \lfloor (k + 2\,n - 1)/J \rfloor J$).

$$\begin{aligned} N_{L2}(k) &= A_1^2[S_i(k)\delta_{p2}^2 + \\ &+ \sum_{n=1}^{(J/2)-1} S_i(\,((k + 2\,n - 1)\%J) + 1)\,\delta_{s2}^2] \end{aligned} \tag{3.10}$$

The total noise contribution at the channelizer output is obtained by combining the terms N_{R2} and $N_{L2}(k)$ with the terms at the upstream levels, thus obtaining:

$$N_{R2}^T = N_{R1}^T + N_{R2};\;\; N_{L2}^T(k) = N_{L1}^T(k) + N_{L2}(k) \tag{3.11}$$

3.5 Synthesis Channelizer

The rounding noise contribution at the channelizer output can be defined from both the FFT and the polyphase network block. For the FFT block, by taking a representation based on $n_{s3'}$ bits and denoting with $q_{s3'} = V_{FS}/2^{(n_{s3'}-1)}$) the corresponding quantization step, the related noise power spectral density can be written as $N_{R3-FFT} = J(\frac{q_{s3'}^2}{3})T_{o3}$ (where the output sample period is $T_{o3} = 1/(2f_0)$).

We then consider the noise contribution due to round-offs within the polyphase network, wherein the output complex samples of a generic branch are given by the sum of two partial results elaborated from a polypase FIR filter with N_3/J coefficients. Since the single element of the branch yields a noise power spectral density contribution given by $(\frac{N_3}{J}\frac{q_{m3}^2}{12})T_{o3}$, by considering the multiplier results expressed with $n_{s3'} + n_{m3}$ bits (with a related quantization step equal to $q_{m3} = V_{FS}/2^{(n_{s3'}+n_{m3}-1)}$) we amplify by a factor of 2 and add the term $(\frac{q_{s3}^2}{12})T_{o3}$ induced by the external round-off performed by the sub-components recomposition block ($q_{s3} = V_{FS}/2^{(n_{s3}-1)}$ is the quantization step related to the data representation, at the channelizer output, that is based on n_{s3} bits). Finally, by also considering that the elaboration is performed on a complex sequence, the noise contribution of the polyphase network is given by $N_{R3-poly} = 2(2\frac{N_3}{J}\frac{q_{m3}^2}{12} + \frac{q_{s3}^2}{12})T_{o3}$ and the total round-off noise contribution of the synthesis channelizer can be rewritten as:

$$N_{R3} = (J\frac{q_{s3'}^2}{3} + \frac{N_3}{J}\frac{q_{m3}^2}{3} + \frac{q_{s3}^2}{6})T_{o3}. \tag{3.12}$$

An interpolation process with interpolation factor equal to $J/2$ is defined on each of the J signals in input to the synthesis channelizer, and a filtering for cancellation of the so-called signal images is then applied. For the k-th input signal, the ripple in the pass-band (with ripple bound δ_{p3}) defines an error with power spectral density equal to $\delta_{p3}^2(A_1^2 S_i(k))$ ($A_1^2 S_i(k)$ is the power spectral density of the input signal). Moreover, the presence of $(J/2) - 1$ images and the non-ideal stop-band behaviour (δ_{s3} denotes the largest stop-band gain), induces noise terms with power spectral density $\delta_{s3}^2(A_1^2 S_i(k))$. In the case of "non critically sampled" operations considered in our system, those noise contributions are spaced in band of a frequency range equal to the ESBW bandwidth. Furthermore, for the synthesis channelizer process the interpolated versions of the J signals are shifted in frequency and then combined (summed), so that errors due to non-ideal filtering are combined as well. In particular, in the k-th portion of band of the output signal the error due to the ripple in the pass-band of the k-th signal considered is summed with portion of errors generated in the stop-band of signals related to other ESBW. By still considering the "non critically sampled" condition and denoting by $S_i(k)$ the power spectral density of the input signal within the k-th ESBW (for k from 1 to J) the power spectral density of the linear distortion term at the output for the k-th band ($N_{L3}(k)$) is due to the distortion in the pass-band and to the residual signals generated in the adjacent ESBWs with index equal to $((k+2n-1)\%J)+1$ (for n from 1 to $(J/2)-1$ and "%" that denotes the module

operation, i.e. $(k + 2\,n - 1) - \lfloor (k + 2\,n - 1)/J \rfloor J$).

$$\begin{aligned} N_{L3}(k) &= A_1^2[S_i(k)\delta_{p3}^2 + \\ &+ \sum_{n=1}^{(J/2)-1} S_i(\,((k + 2\,n - 1)\%J) + 1)\,\delta_{s3}^2] \end{aligned} \tag{3.13}$$

In summary, the total noise contributions at the channelizer output are again obtained by combining the terms N_{R3} and $N_{L3}(k)$ with the terms at the upstream levels, and can be expressed as:

$$N_{R3}^T = N_{R2}^T + N_{R3};\;\; N_{L3}^T(k) = N_{L2}^T(k) + N_{L3}(k) \tag{3.14}$$

3.6 Analytic to IF

The sample period at the "Analytic to IF" block output corresponds to the ADC sampling interval (i.e. $T_{o4} = 1/4f_0$). For the power spectral density of the round-off noise (N_{R4}), let us consider that the quantization step adopted in expressing the multiplication result refers to a representation of the data path through $n_{s3} + n_{m4}$ bits (i.e. $q_{m4} = V_{FS}/2^{(n_{s3}+n_{m4}-1)}$). Nevertheless, assuming an output representation based on n_{s4} bits, a quantization step equal to $q_{s4} = V_{FS}/2^{n_{s4}-1}$ is defined for the accumulation result. The term a_4 is equal to 1 if $n_{s3} \leq n_{s4}$, otherwise is equal to 2.
As an interpolation process with factor 2 is also involved in the "Analytic to IF" block, filtering of an input signal image is in turn required. The noise resulting from the filtering process for the k-th ESBW is again due to the linear distortion in the pass-band (δ_{4p} is the ripple bound in the pass-band with $A_1^2 S_i(k)\delta_{4p}^2$ being the related noise power spectral density), and to the adjacent channel contributions (δ_{4s} is the maximum gain in the stop-band). This last term defines a cross-talk contribution from the $(J + 1 - k)$-th ESBW that, due to the extrapolation of the real output signals, is reported as an alias signal in the k-th ESBW band.
Therefore, the power spectral density of both round-off and linear distortion noises, introduced by the block in the k-th ESBW ($k = 1, ..., J$), can be written as:

$$N_{R4} = 2(\frac{N_4}{2}\frac{q_{m4}^2}{12} + a_4\frac{q_{s4}^2}{12})T_{o4} \tag{3.15}$$

$$N_{L4}(k) = A_1^2\,[S_i(k)\,\delta_{p4}^2 + S_i(J + 1 - k)\,\delta_{s4}^2] \tag{3.16}$$

Finally, when considering the propagation within the block of the noise terms introduced upstream and the scaling factor of $1/4$ resulting on the output power spectral density, due to the extrapolation of the IF signal from the analytic signal,

the overall noise terms (roud-off and linear distortion) at the block output turn out to be:

$$N_{R4}^T = \frac{1}{4}(N_{R3}^T + N_{R4}); \;\; N_{L4}^T(k) = \frac{1}{4}(N_{L3}^T(k) + N_{L4}(k)) \tag{3.17}$$

Chapter 4

Computation of the Hardware Complexity and Power Estimation

The computation of the hardware complexity and the power estimation of the DTP are formalized in this chapter. For each process within the elaboration chain, the hardware complexity is evaluated through a general approach that relies on the computation of the number of necessary operations. The positive aspect of this method is the ability to catch the high level hardware aspects of the DTP not depended to the final technology platform used to realize the DTP, for example FPGA or ASIC. However, the more the method is general and the more the calculation of complexity will be distant from the real value. For this reason, in the case of the FPGAs, a more detailed computation of the hardware complexity is provided. In this context, the analytical expressions, to estimate the power consumption of the DTP, are proposed.

4.1 DTP Hardware Complexity

When considering hardware complexity analysis a preliminary remark is concerned with the level of detail that is retained in HW description: since any specific hardware platform is not assumed in order to guarantee the widest exploitation of the design approach proposed in this manuscript, some specific implementation details are either neglected or simply assumed without further exploration. For instance, filter coefficients are here assumed to be stored in ROMs, while a viable alternative might consist in RAM memories loaded in the start-up phase. A second point is related to the impact on memory size of extra bits related to Error Correction Codes: as these techniques are technology dependent, they are not accounted for in HW complexity computation presented in the following. Finally, the impact of internal rounding blocks in filter implementation has been also neglected:

indeed, the adoption of cleaver techniques such as the one proposed in [25] can make it negligible the impact of rounding blocks in the overall HW complexity computation.

4.1.1 IF to Analytic

The pass-band input signal, with bandwidth $2f_0$ centered at the Intermediate Frequency f_0, is sampled with a rate equal to $4f_0$ and quantized and encoded in two's complement through n_{s0} bits. The sampled signal is processed in P_1, the analytic signal extractor, in order to obtain the analytic representation out of the pass-band real IF signal. The architecture of the "IF to Analytic" block is obtained by assuming a polyphase implementation of the decimation process. From the Half-Band prototype filter with order equal to $N_1 - 1$, a multiplication by the $\ldots$0,1,0,-1$\ldots$ sequence is needed to obtain the frequency shifted complex filter, then two polyphase components are extrapolated. Also due to Half-Band properties a filtering block is needed only for one of the components of the complex output signal, while the other component is generated from a processing free branch. This latter branch requires only a buffer to compensate the delay introduced into the other filtering branch. Equation (4.1) returns the number of two words multipliers (n_{MULT1}) and adders (n_{SUM1}) needed for the implementation of the elaboration branch.

$$n_{MULT1} = \frac{N_1 - 1}{2}; \quad n_{SUM1} = \frac{N_1 - 1}{2} - 1 \tag{4.1}$$

Within the elaboration branch a status register composed by $[(N_1 - 1)/2] - 1$ elements is allocated, while the FIFO buffer for delay management in the other branch is composed by $(N_1 - 1)/4$ elements. In both structures, the amount of FFs for state register implementation ($n_{FF^{SR1}}$) depend on n_{s0}, turn out to be:

$$n_{FF^{SR1}} = \frac{3N_1 - 7}{4} n_{s0} \tag{4.2}$$

In the elaboration branch the results of multiplications, between the coefficients and the status register content are rounded to $n_{s0} + n_{m1}$ bits and buffered. This buffering, introduced to break the combinatorial logic delay, causes a unitary latency that must be inserted also in the other branch. Therefore, at the product level the allocation of Flip-Flops ($n_{FF^{MULT1}}$) is needed. The accumulation process is applied to the multiplication results by using the pipelined binary tree adder. Within the tree adder some registers may be needed to apply the buffering of the n_{BL1} partial results levels ($n_{FF^{SUM1}}$). Finally in the term $n_{FF^{SUM1}}$ we should account for buffering the result of the rounding operation (with n_{s1} bits

representation) applied to the accumulation word. The following expression are then obtained:

$$n_{FF^{MULT1}} = \frac{N_1 - 1}{2}\,(n_{s0} + n_{m1}) + n_{s0}$$
$$n_{FF^{SUM1}} = n_{s0}\,n_{BL1} + 2\,n_{s1} + \sum_{i=1}^{n_{BL1}} \lceil \frac{\frac{N_1-1}{2}}{2^{n_{UBL1}*i}} \rceil * (n_{s0} + n_{m1} + n_{UBL1} * i) \tag{4.3}$$

where:

$$n_{BL1} = \lfloor \frac{\frac{N_1-1}{2}}{n_{UBL1}} \rfloor \leq \log_2 \frac{N_1 - 1}{2} \tag{4.4}$$

The total amount of FFs for the implementation of the IF to Analytic block is given by:

$$n_{FF1} = n_{FF^{SR1}} + n_{FF^{MULT1}} + n_{FF^{SUM1}} \tag{4.5}$$

The storage of the non-null polyphase components, assuming a coefficients representation based on n_{h1} bits, is performed with the allocation of a number of ROM bits equal to:

$$n_{ROM1} = \frac{N_1 - 1}{2}\,n_{h1} \tag{4.6}$$

The "IF to Analytic" block introduces a time delay on the output signal. The delay Δt_{P1} can be expressed as follow:

$$\Delta t_{P1} = (\frac{N_1 - 1}{2} + n_{BL1} + 2)\,\frac{1}{4f_0} \tag{4.7}$$

4.1.2 Analysis Channelizer

The analysis channelizer (P_2) is composed by a polyphase network, that is implemented through a "Variable Coefficient Polyphase Filter" (VCPF), an in place FFT block and two memory elements (Dual Port RAM Block, DPRB). The first DPRB is required to reorganize the output of the VCPF and to insert the delays needed for the management of the "extended set of polyphase components" (see [16] and [26]), while the second DPRB is used to reorganize the output of the FFT block and to restore the "natural order" in the channelizer output sequence.
From the prototype filter, with order equal to $N_2 - 1$, the extended set of J polyphase components is extrapolated. The filtering process is defined on the complex input signal and real filter coefficients. Hence a multiplicity of two must be considered in the hardware complexity computation of the block. Therefore, for the implementation of the VCPF the required number of two words multipliers ($n_{MULT2,1}$) and adders ($n_{SUM2,1}$) is equal to:

$$n_{MULT2,1} = 2\,\frac{N_2}{J}; \quad n_{SUM2,1} = 2\,(\frac{N_2}{J} - 1) \tag{4.8}$$

For a (power of 2) number J of channels, the in place FFT block is composed of $k = log_2\, J$ cascaded stages; they are defined through butterfly blocks followed by complex multipliers, while in the last stage the complex multiplier is not strictly needed. Within the butterfly blocks 4 adders (for the sum of real words) are allocated, while in the complex multipliers we are concerned with allocation of 4 multipliers and 2 adders (all operating on two real words). Then the amount of multipliers ($n_{MULT2,2}$) and adders ($n_{SUM2,2}$) within the FFT stages is given by:

$$n_{MULT2,2} = 4\,(k-1); \quad n_{SUM2,2} = (6\,k-2) \tag{4.9}$$

The total amount of multipliers (n_{MULT2}) and adders (n_{SUM2}) required in the channelizer is then expressed as:

$$\begin{aligned} n_{MULT2} &= n_{MULT2,1} + n_{MULT2,2} \\ n_{SUM2} &= n_{SUM2,1} + n_{SUM2,2} \end{aligned} \tag{4.10}$$

In the VCPF, for each multiplier and adder pair (i.e. for each of the $(N_2/J) - 1$ taps) a First In First Out (FIFO) register, composed by J elements, is allocated in order to extrapolate J polyphase components of the complex input sequence. The input signal is produced from the "IF to Analytic" block and then the amount of FFs required for the implementation of this status register ($n_{FF^{SR2}}$) must be referred to a number of bits equal to n_{s1}.

$$n_{FF^{SR2}} = 2\,(\frac{N_2}{J} - 1)\,J\,n_{s1} \tag{4.11}$$

The polyphase components of the filter (with coefficients represented by n_{h2} bits), as well as the twiddle factors of the FFT (in which the real and complex component are represented with $n_{h2,fft}$ bits), are stored within ROM memories, with a ROM memory for each multiplier. Moreover, ROM blocks with a buffered output are considered to support high processing speed. Therefore, for the implementation of the ROM blocks, also FFs elements must be accounted for both in the "VCPF" ($n_{FF^{ROM2,1}}$) and in the FFT block ($n_{FF^{ROM2,2}}$).

$$\begin{aligned} n_{FF^{ROM2,1}} &= 2\,\frac{N_2}{J}\,n_{h2} \\ n_{FF^{ROM2,2}} &= 2\,(k-1)\,n_{h2,fft} \end{aligned} \tag{4.12}$$

When considering the representation of results at this stage based on $n_{s1} + n_{m2}$ bits, a buffering with $n_{FF^{MULT2,1}}$ FFs is used to break the combinatorial delay between the multiplication and the accumulation processes. The accumulation process is yet performed by the pipelined binary tree adder that involves an amount of registers that provide the buffering of n_{BL2} internal levels. After the round-off

to $n_{s2'}$ bits of the accumulation, also the quantized word is buffered (all FFs of this stage are accounted in $n_{FF^{SUM2,1}}$), finally leading to the following expressions:

$$\begin{aligned} n_{FF^{MULT2,1}} &= 2\,\frac{N_2}{J}\,(n_{s1}+n_{m2}) \\ n_{FF^{SUM2,1}} &= 2\,n_{s2'} + 2\sum_{i=1}^{n_{BL2}} \lceil \frac{\frac{N_2}{J}}{2^{n_{UBL2}*i}} \rceil * (n_{s1}+n_{m2}+n_{UBL2}*i) \end{aligned} \tag{4.13}$$

where:

$$n_{BL2} = \lfloor \frac{\frac{N_2}{J}}{n_{UBL2}} \rfloor \leq \log_2 \frac{N_2}{J} \tag{4.14}$$

In our "non critically sampled" operations for each sample at the VCPF input two outputs are generated sequentially, that should be spaced by $J-1$ samples. To introduce this displacement, and the delays required to manage the extended set of polyphase components, a first DPRB is allocated with a number $n_{FF^{DPRB2,1}}$ of FFs required to store $4\,J$ complex samples. At the FFT block output a second DPRB performs the rearrangement of the sequence in order to recover the "natural order". This rearrangement involves a number of complex samples equal to J with $n_{FF^{DPRB2,2}}$ FFs required to store $2\,J$ complex samples. The following expressions are then obtained:

$$n_{FF^{DPRB2,1}} = 8\,J\,n_{s2'}; \quad n_{FF^{DPRB2,2}} = 4\,J\,n_{s2} \tag{4.15}$$

For the i-th stage of the FFT block a FIFO register with length equal to $J/2^i$ is associated to the butterfly block. At the input of each stage the data path length is increased by 1 to avoid overflow. The data representation at the input of the first FFT stage is imposed by the constraint at the output of the polyphase network (i.e. $n_{s2'}$ bits). After the last stage the data representation is done with n_{s2} bits, assuming this value as the number of bits required for the FFT output. The number $n_{FF^{FFT2}}$ of FFs takes into account the allocation of the FIFO within the butterfly stages and the buffering of both internal product and sum results:

$$n_{FF^{FFT2}} = \sum_{i=1}^{k} 2\,(2+\frac{J}{2^i})(n_{s2'}+i) + \sum_{i=1}^{k-1} 6\,(n_{s2'}+i) \tag{4.16}$$

Therefore, the total amount of FFs for the process P_2 is given by:

$$\begin{aligned} n_{FF2} &= n_{FF^{SR2}} + n_{FF^{ROM2,1}} + n_{FF^{MULT2,1}} + \\ &+ n_{FF^{SUM2,1}} + n_{FF^{DPRB2,1}} + n_{FF^{FFT2}} + \\ &+ n_{FF^{ROM2,2}} + n_{FF^{DPRB2,2}} \end{aligned} \tag{4.17}$$

In the i-th stage of the FFT block three multiplexer with two input and one output are used, except at the last stage which has only two multiplexer. The number of

multiplexer with two input and one output (n_{MUX2}) is expresed as:

$$n_{MUX2} = 4\,k + 2\,(k-1) \tag{4.18}$$

The number of ROM bits for the coefficients ($n_{ROM2,1}$) and the twiddle factors ($n_{ROM2,2}$) memorization is given by:

$$\begin{aligned} n_{ROM2,1} &= N_2\; n_{h2} \\ n_{ROM2,2} &= \sum_{i=1}^{k} 2\,(\frac{J}{2^i})\; n_{h2,fft} \end{aligned} \tag{4.19}$$

The time-delay introduced by the Analysis Channalizer (Δt_{P2}) is given by the VCPF filter ($\Delta t_{P2,1}$), the FFT block ($\Delta t_{P2,2}$) and the DPBR blocks ($\Delta t_{DPBR2,1}$ and $\Delta t_{DPBR2,2}$ respectively). The time-delays are described by the following expressions:

$$\begin{aligned} \Delta t_{P2,1} &= ((\frac{N_2}{J} - 1)\; J + n_{BL2} + 2)\; \frac{1}{4f_0} \\ \Delta t_{DPBR2,1} &= (\frac{4\;J}{2} + 1)\; \frac{1}{4f_0} \\ \Delta t_{P2,2} &= (\sum_{i=1}^{k}(3 + \frac{J}{2^i}) + 4)\; \frac{1}{4f_0} \\ \Delta t_{DPBR2,2} &= (\frac{2\;J}{2} + 1)\; \frac{1}{4f_0} \\ \Delta t_{P2} &= \Delta t_{P2,1} + \Delta t_{DPBR2,1} + \Delta t_{P2,2} + \Delta t_{DPBR2,2} \end{aligned} \tag{4.20}$$

4.1.3 Synthesis Channelizer

In the synthesis channelizer the input signals related to the J ESBWs are recombined in order to obtain the output analytic signal. As for the analysis block, the channelizer may be implemented by combining the FFT with a filtering process performed through the polyphase network. Therefore, the blocks involved in this section are the same of the analysis channelizer and similar considerations hold true for the discussion of this block with a very few exceptions.
Also in this case the FFT block is implemented by $k = log_2(J)$ stages composed by a FIFO register, a butterfly network and a complex multiplier. The number of two words multipliers ($n_{MULT3,1}$) and adders ($n_{SUM3,1}$) involved in the implementation of the FFT must be incorporated, in order to obtain the overall term related to the synthesis channelizer, with the elements needed for the VCPF of the polyphase network. For the FFT block we obtain:

$$n_{MULT3,1} = 4\,(k-1); \quad n_{SUM3,1} = (6\,k - 2) \tag{4.21}$$

From the prototype filter, with order equal to $N_3 - 1$, a first set of $J/2$ polyphase components are extrapolate and, for each of them, a further decomposition with factor of two is performed. Two polyphase components of the analytic signal are provided at the output of the polyphase network. These components must be summed to obtain the output signal, so that, an adder block is allocated after the polyphase network [26]. As a result, the number of multipliers ($n_{MULT3,2}$) and adders ($n_{SUM3,2}$) involved in the VCPF are obtained as:

$$n_{MULT3,2} = 2\,\frac{N_3}{J}; \quad n_{SUM3,2} = 2\,\frac{N_3}{J} \tag{4.22}$$

The total amount of multipliers (n_{MULT3}) and adders (n_{SUM3}) needed within the synthesis channelizer is then expressed as:

$$\begin{aligned} n_{MULT3} &= n_{MULT3,1} + n_{MULT3,2} \\ n_{SUM3} &= n_{SUM3,1} + n_{SUM3,2} \end{aligned} \tag{4.23}$$

The input data representation size of the FFT is driven from the switch and is intended to be identical to the one used in output from the analysis channelizer, i.e. with n_{s2} bits. After the last stage the data representation is done with $n_{s3'}$ bits, assuming this value as the number of bits required for the FFT output. Therefore the implementation of the FFT stages requires a number of FFs equal to:

$$n_{FF^{FFT2}} = \sum_{i=1}^{k} 2\,(2 + \frac{J}{2^i})(n_{s2} + i) + \sum_{i=1}^{k-1} 6\,(n_{s2} + i) \tag{4.24}$$

Also in this case additional FFs must be considered for the implementation of the buffered ROM memories that are used for storing both twiddle factors (real and imaginary component are represented with $n_{h3,fft}$ bits) and filter coefficients (represented by n_{h3} bits). In the FFT ($n_{FF^{ROM3,1}}$) and in the VCPF ($n_{FF^{ROM3,2}}$) the number of ROM memories (and then the number of output buffer) is equal to the number of multipliers, thus leading to:

$$\begin{aligned} n_{FF^{ROM3,1}} &= 2\,(k-1)\,n_{h3,fft} \\ n_{FF^{ROM3,2}} &= 2\,\frac{N_3}{J}\,n_{h3} \end{aligned} \tag{4.25}$$

Between the FFT and the Polyphase network a DPRB is interposed to reorganize the sample sequence. The number $n_{FF^{DPRB3}}$ of FFs employed to implement the DPRB must be suitable to store $4J$ complex words expressed through $n_{s3'}$ bits, i.e.

$$n_{FF^{DPRB3}} = 8\,J\,n_{s3'} \tag{4.26}$$

For the synthesis channelizer two polyphase components of the J input signals (related to the J channels) must be extrapolated. Therefore, for each of the $(N_3/J) - 1$ FIFOs that compose the status register, a number of $2J$ registers at $n_{s3'}$ bits ($n_{s3'}$ is the number of bits of the data in input to the polyphase network) are required. Therefore, the number of FFs needed for the implementation of the VCPF status register is equal to:

$$n_{FF^{SR3}} = 4\,(\frac{N_3}{J} - 1)\,J\,n_{s3'} \tag{4.27}$$

Additional FFs ($n_{FF^{MULT3,2}}$) are needed for the buffering of the product results of the VCPF, that are represented with $n_{s3'} + n_{m3}$ bits.

$$n_{FF^{MULT3,2}} = 2\,\frac{N_3}{J}\,(n_{s3'} + n_{m3}) \tag{4.28}$$

The accumulation results is provided with a data representation based on $n_{s3'} + n_{m3} + n_{acc3}$ bits (n_{acc3} is the number of additional bits useful to avoid internal overflow within the accumulation process). The round-off process that produces the required output data representation is applied (representation based on n_{s3} bits) to the sum result, before the output buffering, . The process of accumulation within the polyphase network and the external sum, considering a number of n_{BL3} buffered levels within the pipelined tree adder, involves a number of additional FFs given as:

$$\begin{aligned} n_{FF^{SUM3,2}} &= 4\,(n_{s3'} + n_{m3} + n_{acc3}) + 2\,n_{s3} + \\ &+ 2\sum_{i=1}^{n_{BL3}} \lceil \frac{\frac{N_3}{J}}{2^{n_{UBL3}*i}} \rceil * (n_{s3'} + n_{m3} + n_{UBL3} * i) \end{aligned} \tag{4.29}$$

where:

$$n_{BL3} = \lfloor \frac{\frac{N_3}{J}}{n_{UBL3}} \rfloor \leq \log_2 \frac{N_3}{J} \tag{4.30}$$

The total amount of FFs within the synthesis channelizer is given by:

$$\begin{aligned} n_{FF3} &= n_{FF^{FFT3}} + n_{FF^{ROM3,1}} + n_{FF^{SR3}} + \\ &+ n_{FF^{DPRB3}} + n_{FF^{MULT3}} + n_{FF^{SUM3}} + \\ &+ n_{FF^{ROM3,2}} \end{aligned} \tag{4.31}$$

The number of ROM bits for twiddle factors ($n_{ROM3,1}$) and coefficients ($n_{ROM3,2}$) memorization is given by:

$$\begin{aligned} n_{ROM3,1} &= \sum_{i=1}^{k} 2\,(\frac{J}{2^i})\,n_{h3,fft} \\ n_{ROM3,2} &= N_3\,n_{h3} \end{aligned} \tag{4.32}$$

for a total amount of ROM bits equal to:

$$n_{ROM3} = n_{ROM3,1} + n_{ROM3,2} \tag{4.33}$$

Also in this block there are n_{MUX3} multiplexer in the FFT:

$$n_{MUX3} = 4\,k + 2\,(k-1) \tag{4.34}$$

The time-delay introduced by the Synthesis Channalizer (Δt_{P3}) is given by the VCPF filter ($\Delta t_{P3,2}$), the FFT block ($\Delta t_{P3,1}$) and the DPBR block ($\Delta t_{DPBR3,1}$). The time-delays are described by the following expressions:

$$\begin{aligned}
\Delta t_{P3,1} &= (\sum_{i=1}^{k}(3 + \frac{J}{2^i}) + 4)\,\frac{1}{4f_0} \\
\Delta t_{DPBR3,1} &= (\frac{2\,J}{2} + 1)\,\frac{1}{4f_0} \\
\Delta t_{P3,2} &= ((\frac{N_3}{J} - 1)\,J + n_{BL3} + 5)\,\frac{1}{4f_0} \\
\Delta t_{P2} &= \Delta t_{P3,1} + \Delta t_{DPBR3,1} + \Delta t_{P3,2}
\end{aligned} \tag{4.35}$$

4.1.4 Analytic to IF

The ”Analytic to IF” block processes the output signal of the synthesis channelizer in order to obtain the discrete IF signal. For this purpose the condition of oversampling is restored on the channelizer output through a polyphase interpolator operating with conversion factor equal to 2. The analytic signal is then obtained and, by taking its real part, the discrete IF signal is generated. Also in this case, for considerations similar to the ones given for the "IF to Analytic" block, the block may by implemented through a simplified architecture, wherein, only one of the two polyphase component of the output signal requires a true elaboration process (filtering), while the other aligns the delay.
Considering a filter order equal to $(N_4 - 1)$, and an input data-path based on n_{s3} bits, for the status register of the block the number of required FFs is equal to:

$$n_{FF^{SR4}} = \frac{3N_4 - 7}{4}\,n_{s3} \tag{4.36}$$

The number of two words multipliers (n_{MULT4}) and adders (n_{SUM4}) needed for the implementation of the elaboration branch is equal to:

$$n_{MULT4} = \frac{N_4 - 1}{2}; \quad n_{SUM4} = \frac{N_4 - 1}{2} - 1 \tag{4.37}$$

Additional FFs ($n_{FF^{MULT4}}$) are needed for the buffering of the product results (represented by $n_{s3} + n_{m4}$ bits):

$$n_{FF^{MULT4}} = \frac{N_4 - 1}{2} (n_{s3} + n_{m4}) + n_{s3} \tag{4.38}$$

Considering a number n_{BL4} of buffer levels within the pipeline tree adder, the number of FFs required for the accumulation process is equal to:

$$n_{FF^{SUM4}} = n_{s3}\, n_{BL4} + 2\, n_{s4} + \sum_{i=1}^{n_{BL4}} \lceil \frac{\frac{N_4-1}{2}}{2^{n_{UBL4}*i}} \rceil * (n_{s3} + n_{m4} + n_{UBL4} * i) \tag{4.39}$$

where:

$$n_{BL4} = \lfloor \frac{\frac{N_4-1}{2}}{n_{UBL4}} \rfloor \leq \log_2 \frac{N_4 - 1}{2} \tag{4.40}$$

The total amount of FFs for the implementation of the IF to Analytic is given by:

$$n_{FF4} = n_{FF^{SR4}} + n_{FF^{MULT4}} + n_{FF^{SUM4}} \tag{4.41}$$

The storage of the coefficients, represented through n_{h4} bits, requires a number of ROM bits equal to:

$$n_{ROM4} = \frac{N_4 - 1}{2}\, n_{h4} \tag{4.42}$$

The "Analytic to IF" block introduces a time delay on the output signal. The delay Δt_{P4} can be expressed as follow:

$$\Delta t_{P4} = (\frac{N_4 - 1}{2} + n_{BL4} + 2)\, \frac{1}{4 f_0} \tag{4.43}$$

4.2 DTP Power Estimation Model

In order to estimate the power consumption of the DTP, a hardware library is defined and a power estimation is associated to each element in the library. From the RTL description of the DTP, a mapping between the RTL components and the elements in the library is accomplished. As a result, the DTP power estimation can be expressed as the sum of the power of the library's components that compose the DTP. In this way, the power estimation depends on the level of abstraction of the hardware library. In the case of an FPGA implementation, the hardware library is composed by the programmable logic blocks of the FPGA. In order to generalize our power estimation model respect to a specific FPGA, a Virtual FPGA (vFPGA) with a specific hardware library is defined.

4.2.1 Virtual FPGA

In order to define the vFPGA, the hardware primitives are defined. These primitives provide the elementary functions used to implement the DTP's elaboration chain. In order to test the power estimation model through a real implementation on a commercial FPGA, it is reasonable to define the hardware library according to the basic blocks that can be found on commercial FPGAs.

The following four hardware primitives are defined:

- look-up table 3-input 2-output (LUT3)
- carry logic (CARRY4)
- Flip-Flop (FF)
- DSP

The LUT3 is the function generator that implements a boolean function with three independent input variables and two outputs. The LUT3 can be used with two outputs (LUT3o2) or with one output (LUT3o1). The CARRY4 is a dedicated fast look-ahead carry logic used to perform a fast arithmetic addition and subtraction. The FF is a 1-bit memory element and the DSP is an elementary block that performs the multiplication function. Let HW_{com} be the set of hardware primitives of the vFPGA, then:

$$HW_{com} = \{FF, LUT3o1, LUT3o2, CARRY4, DSP\} \tag{4.44}$$

For each high-level hardware component of the RTL description of the DTP given in Chapter 2, a mapping with the hardware primitives of the vFPGA can be written as:

$$\begin{aligned} REG(n) &\longrightarrow n \times FF + 1 \times MUX2to1(n) \\ ADDER(n) &\longrightarrow \left\lceil \frac{n}{4} \right\rceil CARRY4 + n \times LUT3o2 \\ MUX2to1(n) &\longrightarrow n \times LUT3o1 \\ MULT(n) &\longrightarrow 1 \times DSP(n) \end{aligned} \tag{4.45}$$

where $REG(n)$ is a n-bit register, $ADDER(n)$ is an adder that executes the sum of two words with n-bit, $MUX2to1(n)$ is a multiplexer that performs the switch between two words with n-bit, and $MULT(n)$ is a multiplier that executes the multiplication of two words and returns a word with n-bit.

4.2.2 Power Estimation

A first estimate of the dynamic power is calculated by taken into account the output bits of each internal block. The following expression for the dynamic power is adopted:

$$P_{dyn} = \frac{1}{2} F_c V^2 C_{net} S \tag{4.46}$$

where F_c is the clock frequency, V is the voltage supply, C_{net} is the capacitance of the network connected to the output bits of each block and S represents the switching activity on the output bits, as reported in [27] and [28]. The above expression is used to estimate the dynamic power on ASICs and FPGAs and it is referred to a single bit. Fig. 4.1 shows the charge and discharge of the output capacity of a CMOS circuit responsible for the dissipation of the dynamic power. The losses during the switching of the two transistors p and n are neglected and the power dissipated over an entire LOW-HIGH-LOW output cycle is considered. The load capacity of a CMOS logic port is influenced by the number of input ports

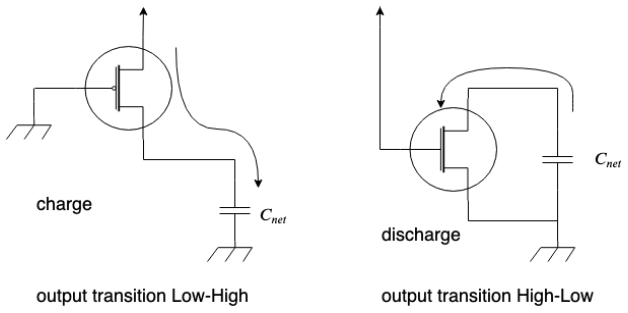

Figure 4.1: Charge phase and discharge phase of the output in CMOS circuits.

of the cascaded blocks connected to the logic port, namely the fan-out (FO).

Based on the above remarks, the model defined for estimating the dynamic power is as follows:

$$P_{dyn} = \frac{1}{2} F_c V^2 \, \overline{\gamma} \, \overline{\mathcal{O}} \tag{4.47}$$

where the vectors $\overline{\gamma}$ and $\overline{\mathcal{O}}$ are:

$$\begin{aligned} \overline{\mathcal{O}} &= (\mathcal{O}_{FF} \; \mathcal{O}^c_{FF} \; \mathcal{O}_{LUT} \; \mathcal{O}_{CARRY4} \mathcal{O}_{DSP}) \\ \overline{\gamma} &= (\gamma_{FF} \; \gamma^c_{FF} \; \gamma_{LUT} \; \gamma_{CARRY4} \; \gamma_{DSP}) \end{aligned} \tag{4.48}$$

and specifically:

$$\begin{aligned}
\mathcal{O}_{FF} &= \sum_{\forall Register} out \\
\mathcal{O}^{c}_{FF} &= \sum_{\forall Register} out \times FO \\
\mathcal{O}_{LUT} &= \sum_{\forall Adder} 2 * out + \sum_{\forall Mux} out \\
\mathcal{O}_{CARRY4} &= 5 \sum_{\forall Adder} \left\lceil \frac{out}{4} \right\rceil \\
\mathcal{O}_{DSP} &= \sum_{\forall Mult} out
\end{aligned} \tag{4.49}$$

The vector $\overline{\gamma}$ is composed by the coefficients necessary to adapt the model to the specific block. These coefficients represent the switching activity S and the load capacity C_{net}, introduced in the Eq. (4.46), for each hardware primitive of the vFPGA. The vector $\overline{\mathcal{O}}$ contains, for each hardware primitive of the vFPGA, the number of output bits. In the current stage of the development of our power estimation model, only the contribution of the fan-out on the registers has been reported in Eq. (4.49). The term $\mathcal{O}_{FF}$ take into account the power dissipation on the clock lines.

Chapter 5

Performance Results and Numerical Examples

5.1 Link Performance in Selected System Scenarios

The characterization of the DTP processing chain in terms of additional noise and related noise figures, allows to incorporate the effect of the on-board digital processing within the link budget analysis. In summary, the selection of the set of parameters of the DTP chain is performed in order to meet link performance requirements. Furthermore, by the explicit calculation of the hardware complexity for each block, the hardware resources for the DTP implementation can be computed. By resorting on such a comprehensive framework, it is possible to show how the adoption of different system configurations are directly related to the design issues of the digital on-board processing blocks.

Run-time adaptation of transmission formats (namely, coding and modulation schemes) according to the conditions of the communication channel is a standard practice in many cases [29], and the solution addressed within the ESA MHOMS project [30] supports a total of 27 ACM formats. Parameters' setting and performance of these 27 ACM formats on the Additive White Gaussian Noise channel are reported in Table 2 of [30]. The two extreme cases of modulation and coding modes allowed by the adopted ACM modulation and coding scheme, namely mode ACM1 (QPSK with Rb/Rs efficiency 0,71) and mode ACM27 (64 APSK with Rb/Rs efficiency 5.39) can be considered as design examples. In particular, as shown in [13], the analysis for the highest throughput ACM 27 format is equivalent to evaluate the worst case impact of the DTP on the link performance. In Table 5.1 the link budget for the ACM 27 is reported in the two cases of processed bandwidth equal to 125 MHz and 250 MHz, respectively. The link budget refers to an ideal digital processing chain, and then it does not take into account the

additional noise introduced by the DTP.

When moving from a 125 MHz processed bandwidth to a 250 MHz bandwidth while maintaining the same symbol rate R_s for the considered signals, the number of carriers in the uplink duplicates (from 3 to 6 in Table 5.1). While keeping the same input dynamics for the ADC, a reduced gain in the analog section of the on-board receiver must be considered (from 50.75 dB to 47.74 dB) in order to avoid clipping effects. It is worth of mention that this gain includes the contribution of the Low Noise Amplifier (LNA) stage and the Automatic Gain Control (AGC) stage. As gain variations are implemented in the later amplification stage, the impact of those moderate variations on the RX G/T can be found to be negligible. In general, those degradations can be counteracted by the additional margin that appears in the link budget and that is left available at the down-link ground receiver.

Nevertheless, according to the Friis' formula [31] and the extended notion of noise figure, the gain reduction of the analog section also has an impact on the scaling of DTP noise intensities. Specifically, the effect of DTP noise sources is quantified in the link budget in terms of reduction of the additional margin versus the overall DTP noise figure. Figure 5.1 reports the additional margin that the extended link budget makes it available at the ground receiver when 125 MHz (continuous black line) and 250 MHz (dotted blue line) are considered for the processed bandwidth, respectively: the larger the noise figure, the smaller the additional margin is.

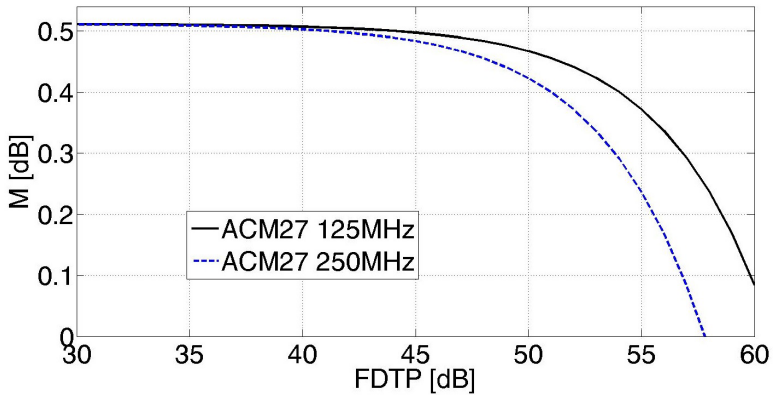

Figure 5.1: Additional margin at the ground receiver for the ACM27 mode, when 125 MHz (continuous black line) and 250 MHz (dotted blue line) of processed bandwidth are considered.

Let us consider for instance a DTP overall setting that relate to a $F_{DTP} = 54$ dB with a 125 MHz processed bandwidth: the additional margin M reduces to $M = 0.4$ dB. In order to keep the same 0.4 additional margin when a 250 MHz

Table 5.1: Link Budget for the system scenario

	ACM27 125 MHz	ACM27 250 MHz
Uplink:		
Up-Link Frequency [GHz]	28,5000	28,5000
Tx Antenna Gain [dBi]	64,20	64,20
Per-Carrier HPA Power [W]	50,0	50,0
Ground Station EIRP [dBW]	81,19	81,19
Free Space Attenuation [dB]	213,06	213,06
Atmospheric Suppl. Att. [dB]	0	0
Sat. RX antenna Gain [dBi]	41,30	41,30
Onb. RX G/T [dB/K]	13,70	13,70
C/N_0 Up [dBHz]	110,43	110,43
Single Channel R_s [MHz]	27.5	27.5
E_s/N_0 Up [dB]	36.04	36.04
No. of Uplink Carriers	3	6
SRRC roll-off	0.2	0.2
Analog Ampl. gain $G_{AA}[dB]$	50.75	47,74
Onboard ADC biasing:		
ADC Biasing (50 OHM) [Veff]	0.5*1.41	0.5*1.41
Downlink:		
Downlink Frequency [GHz]	19,9500	19,9500
Per-Carrier HPA Power [W]	14.35	14.35
Onboard TX Ant. Gain [dBi]	39,75	39,75
EIRP [dBW]	50,80	50,80
Atmospheric Suppl. Att. [dB]	0	0
Free Space Attenuation [dB]	209,96	209,96
Ground Receiver G/T [dB/K]	24.3	24.3
Ground RX Antenna Gain [dBi]	49.72	49.72
C/N_0 Downlink [dBHz]	93,65	93,65
Modulation and Coding:		
MHOMS ACM Mode:	ACM27	ACM27
Modulation:	64 APSK	64 APSK
Rb/Rs	5,39	5,39
E_s/N_0 (10E-7 BER) [dB]	19.25	19.25
Additional margin [dB]	0.5	0.5

Table 5.2: DTP design paramiters

	n_s	$20log_{10}(\delta)$	M [dB]
$BW = 125MHz$ $F_{DTP} = 54dB$ $J = 512$	15	-63	0.4
$BW = 250MHz$ $F_{DTP} = 51dB$ $J = 1024$	16	-69	0.4

processed bandwidth is considered, a smaller DTP noise figure would be required, namely $F_{DTP} = 51$ dB: this better performance required for the DTP would imply different settings of DTP parameters and increased hardware complexity.

Furthermore, the single ESBW bandwidth increases if the increase of the overall processed bandwidth is done while keeping the same value of J. In general, if the same level of flexibility in frequency resource assignment is required, the number J of extrapolated ESBWs has to increase linearly with the overall processed bandwidth. For instance, if an ESBW size of 244.1 kHz is selected for $J = 512$ and $BW_o = 125$ MHz, a number J of extrapolated ESBWs equal to 1024 is required when $BW_o = 250$ MHz. This remark leads to a further increase of the hardware complexity of the DTP.

As a first example of our design procedure, Table 5.2 reports the setting of main implementation parameters that are returned for the two considered scenarios ($BW_o = 125$ MHz and $BW_o = 250$ MHz): in both cases the same value of additional margin to the ground receiver (M = 0.4 dB) has been imposed, and the same ESBW size 244.1 kHz has been assumed. Furthermore, the hardware complexity computation refers to the Uniform Parameters design approach [P4].

When $BW_o = 250$ MHz is considered, a further increase in the hardware complexity is induced by the increase in the processing frequency of the digital system and by the increase of the internal buffering levels that guarantee the constraints on the internal combinatorial delays. In this regard Table 5.3 reports the computation results for the DTP hardware complexity expressed in terms of number of multipliers, number of adders, number of FFs, and total ROM size. It can be observed that significant differences occur especially in terms of number of FFs, and total ROM size.

Table 5.3: DTP hardware complexity

	No. of Mult.s	No. of Add.s	No. of FF.s (1/1024)	ROM (k bits)
$BW = 125MHz$ $F_{DTP} = 54dB$ $J = 512$	224	259	586	239
$BW = 250MHz$ $F_{DTP} = 51dB$ $J = 1024$	250	289	1365	5760

5.2 Design Approaches and Hardware Complexity

In computing hardware complexity we consider a DTP with the previously defined processing blocks. In [14] the design lies in the design approach denoted as UP design: the same number n_s of bits for the data-path at the I/O interface, and the same number n_m of additional bits for the products' representation within the filtering processes are adopted for all blocks in the chain.

In this manuscript the extended models presented in Chapter 3 and 4 are exploited to explore other (block-specific) design alternatives. Let the interface of the process P_i be considered: while the input data representation is forced by the choice done on the previous block ($n_{s\,(i-1)}$), the output data representation and the additional bits for the internal results can be independently selected for the generic process P_i. In particular, the constraints related to filter design can be specified on a stage basis and the bounds related to pass-band ripple and stop-band gain can be handled separately. A pass-band ripple constraint identical to the maximum stop-band gain constraint ($\delta = \delta_p = \delta_s$) is assumed, together with a balanced composition of the input denoted as a flat signal uniformly distributed within the whole bandwidth processed by the DTP.

Then, considering the same set of ESBWs ($J = 32, 64, 128$) and the same link budget parameters considered in [14], two alternative design approaches are evaluated and compared to the UP case. A first novel design procedure is concerned with an even splitting of the overall noise of the DTP along the various blocks of the chain: this Uniform Contribution (UC) of noise clearly requires that the parameters of each processing block need to be specifically selected. Let Fig. 5.2 be considered, where the additional margin at the ground receiver is plotted versus the DTP noise figure: the larger the additional margin required, the smallest the DTP noise figure that must be guaranteed and that drives HW design. Figs. 5.3, 5.4, 5.5 and 5.6 provide for the two mentioned design approaches the HW complexity of the whole DTP chain, expressed in terms of number of multipliers,

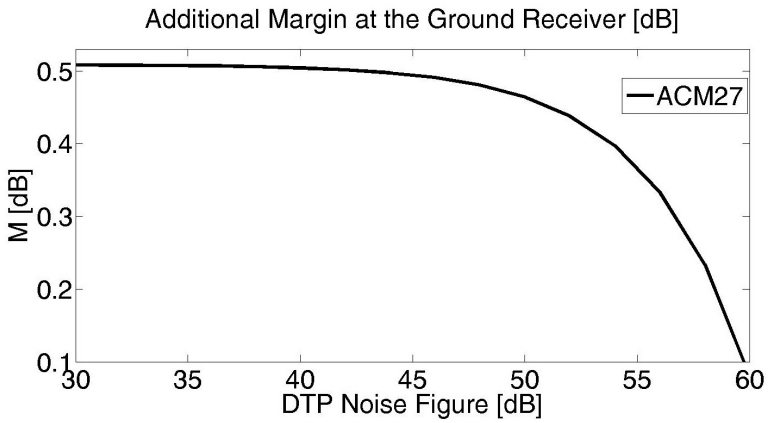

Figure 5.2: Additional margin at the ground receiver versus DTP noise figure.

number of adders, number of FFs, and total ROM size: the continuous lines are related to the UP design approach, while the resulting hardware complexity for the UC approach is plotted in dotted lines. By inspection of the figures, it can

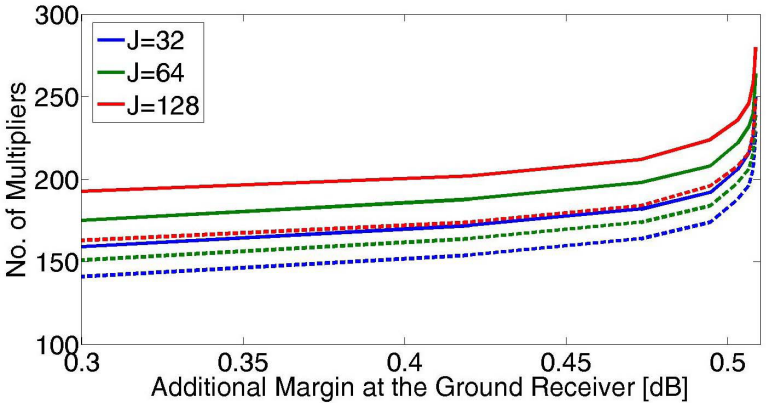

Figure 5.3: Number of Multipliers for the UP approach (continuous lines) and UC approach (dotted lines).

be clearly observed that for a given requirement on the overall DTP noise figure and number of ESBWs (J), the UC approach is able to provide an appreciable reduction of the hardware complexity of the whole processing chain with respect to the UP design approach. For instance, let a DTP noise figure requirement of 54 dB be considered, so that a link-budget degradation can be induced and only a 0.1 dB additional margin at the ground receiver is available. For $J = 64$ the implementation parameters for the two design approaches (D.A.) are reported in the first two rows of Table 5.4. It can be observed that quite different choices for the HW design of the various blocks are returned by the UC approach.

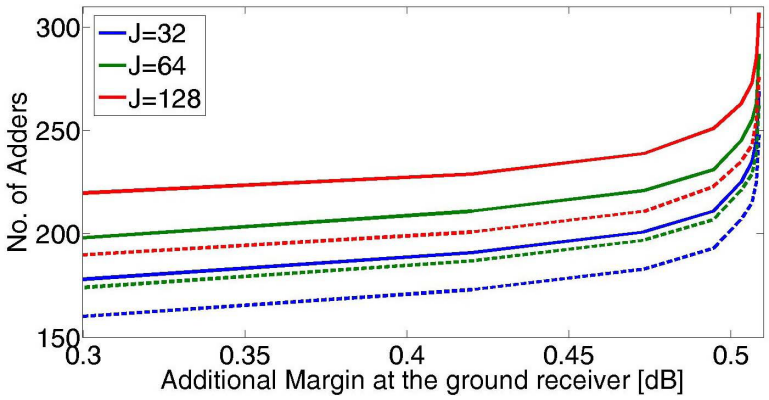

Figure 5.4: Number of Adders for the UP approach (continuous lines) and UC approach (dotted lines).

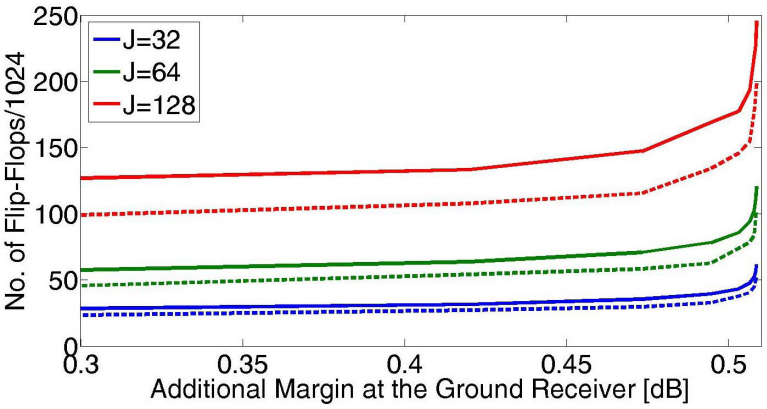

Figure 5.5: Number of Flip-Flops normalized to 1024 for the UP approach (continuous lines) and UC approach (dotted lines).

Although the switch matrix has been explicitly neglected in both our model and in our design approach, it can be argued that the required HW resources of the overall transponder are affected by the switch and the related complexity computation should be addressed in the future. In this regard it can be simply observed that, whenever the switching functionality in the time domain needs to be supported (e.g. to support the MF-TDMA standard), the switch matrix implementation is based also on the allocation of several RAM memory blocks that reorganize the output sequences provided by the analysis channelizer (P_2). By looking at Table 5.4, it can be seen that for the UC approach the number of bits used to represent the data at the analysis channelizer output is larger with respect to the UP approach: therefore, the UC approach would likely induce a larger amount of

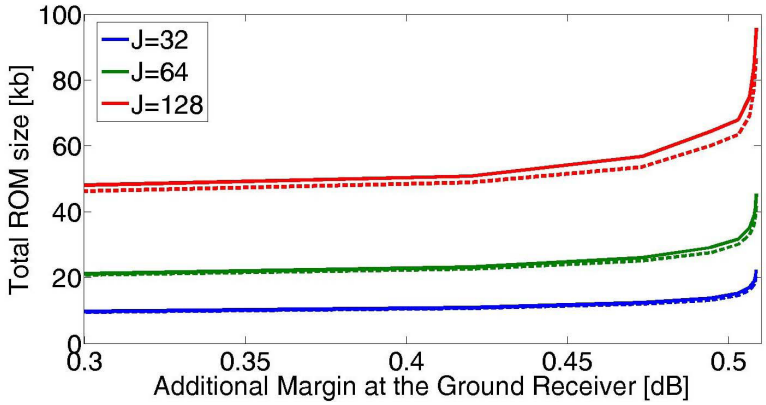

Figure 5.6: Total ROM size expressed in kb for the UP approach (continuous lines) and UC approach (dotted lines).

Table 5.4: DTP parameters as obtained with four approaches (UP, UC, MR and TO) for $F_{DTP} = 54\,dB$ and $J = 64$

D.A.	Block	P_0	P_1	P_2	P_3	P_4
UP	n_s bits	12	12	12	12	12
	δ [dB]	/	-55	-55	-55	-55
UC	n_s bits	7	8	14	8	7
	δ [dB]	/	-45	-57	-57	-45
MR	n_s bits	8	9	11	10	9
	δ [dB]	/	-62	-74	-74	-62
TO	n_s bits	9	10	12	10	9
	δ [dB]	/	-45	-57	-57	-45

RAM resources in the switch section. As a matter of fact, the UC approach can be definitely adopted whenever the critical elements driving the design process are the resources used for the elaboration chain (i.e. multipliers, adders, etc).

On the other hand, when also the switch complexity becomes a major concern, an approach for the definition of Minimum RAM (MR) requirements (minimum requirements for the switch) can be defined: it is based on apportioning the overall

noise of the DTP ($N_{DTP} = N_{R4}^T + N_{L4}^T$) so that quantization and rounding noises are prevalent (i.e. $N_{R4}^T >> N_{L4}^T$) and letting the analysis channelizer noise term be significantly larger than other noise sources in the chain. This approach induces a larger hardware complexity for the DTP processing chain. However, as reported in the third row of Table 5.4, the MR approach allows a data representation at the analysis channelizer output (P_2) that is based on only 11 bits.

Since from Table 5.4 the UP approach appears quite close to the MR one in terms of representation efficiency at the P_2 output and thus in the switch complexity, a final design alternative considered is concerned with a trade-off solution (TO): it moves from the switch complexity defined with the UP approach and introduces a reduction in the hardware complexity of the elaboration chain. The last row of Table 5.4 summarizes the implementations parameters related to this latter solution. TO is again obtained by maintaining dominant the noise contribution introduced by the analysis channelizer, but assuming a uniform distribution between quantization/rounding and linear distortion noise terms ($N_{R4}^T = N_{L4}^T$). Fig. 5.7 reports the overall hardware complexity, for a DTP elaboration chain characterized by a $F_{DTP} = 54\,dB$ and $J = 64$, as obtained for the four different design approaches. It can actually be observed that the TO approach is promising

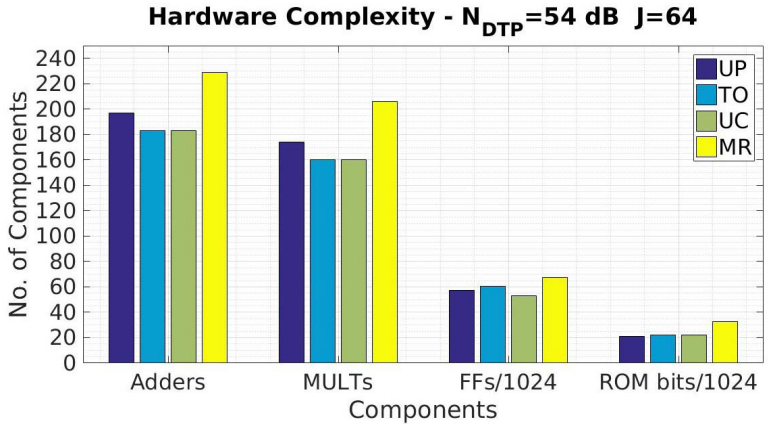

Figure 5.7: Hardware complexity of a DTP elaboration chain with $F_{DTP} = 54\,dB$ and $J = 64$ for the different design approaches. UP: Uniform Parameters; TO: Trade-Off solution; UC: Uniform Contribution; MR: Minimum RAM requirements for the switch.

in keeping the switch complexity at the same level provided by the UP approach and in achieving the same level of reduction in hardware complexity provided by the UC approach, with the only exception of the amount of storage elements (flip-flops).

5.3 VHDL Simulation: Hardware Complexity

The analytic framework presented so far has been validated and is able to enable quite easy evaluation of DTP performance along with computation of its hardware complexity. Nevertheless, developing a detailed description and simulation environment is a standard step in every hardware design and synthesis procedure. Therefore, the DTP has also been designed and implemented in VHDL. In order to check the agreement between the results provided by both the analytic method and the simulation and synthesis environment, we first consider only the blocks "IF to Analytic" and "Analytic to IF". In Fig.5.8 the RTL diagram of the unit under test (UUT) is depicted. The output of the "IF to Analytic" is used to feed directly the "Analytic to IF" block. Furthermore, the comparison between the model and the simulation/synthesis environment is carried out in terms of: i) the SNR at the input and at output of the UUT; ii) the hardware complexity.

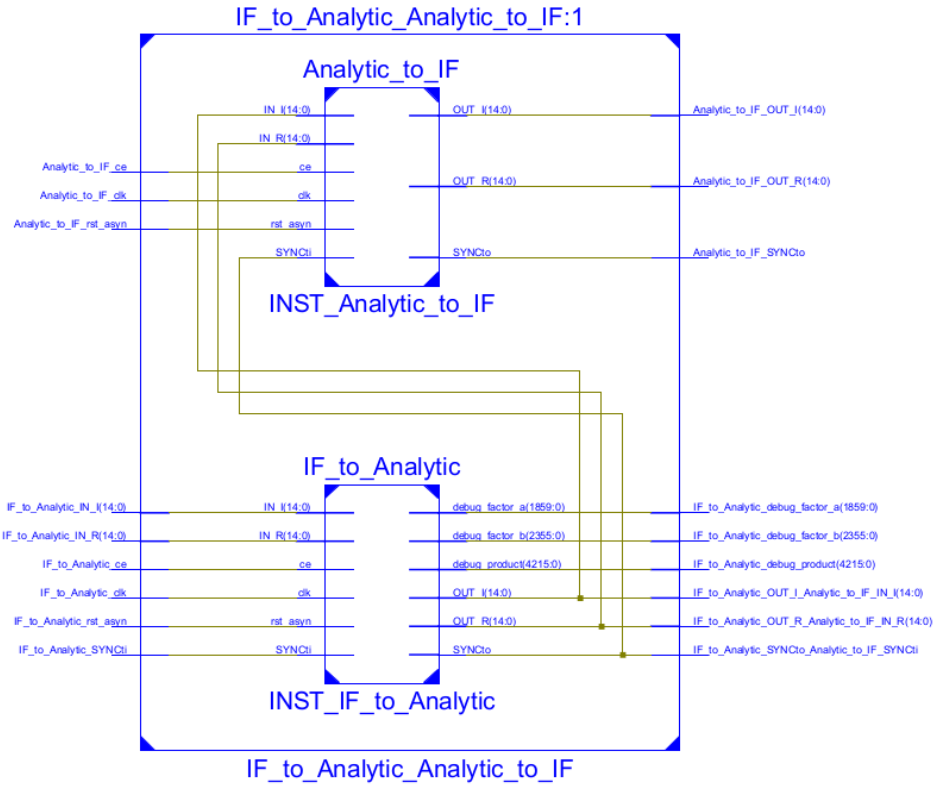

Figure 5.8: **RTL diagram of block under test.**

Three QPSK modulated signals have been generated to emulate the up-link transmission: Fig.5.9 shows the amplitude spectrum of the input signal ($|A(f)|$),

of the analytical signal ($|B(f)|$), and of the signal at the UUT output $|C(f)$.

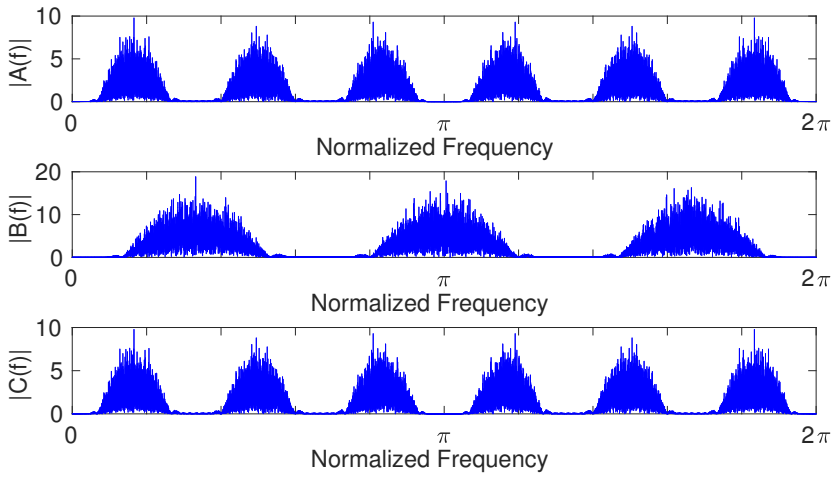

Figure 5.9: Spectral representation of signals along the UUT chain: a) $|A(f)|$: "IF to Analytic" input; b) $|B(f)|$: output of the "IF to Analytic" and input of the "Analytic to IF" block output; c) $|C(f)|$: "Analytic to IF" output.

When the SNR performance is considered, Table 5.5 reports the set of parameters adopted in seven test cases that cover a set of different requirements. The parameter F_{UUT} denotes the noise figure referred to the UUT and Fig.5.10 reports the SNR trend with respect to the noise figure as obtained from both the model and the simulation/synthesis: while a decreasing trend is of course observe, the agreement between model and simulation is very good.

Table 5.5: Summary of parameters for the test cases.

	F_{UUT} [dB]	n_s	n_m	N_1-1 and N_4-1	n_h	δ [dB]
Case 1	30	16	3	264	20	-87.26
Case 2	35	15	3	248	19	-82.71
Case 3	40	14	3	232	19	-78.33
Case 4	45	13	4	212	18	-72.66
Case 5	50	12	4	196	18	-68.33
Case 6	55	11	6	180	17	-63.61
Case 7	60	11	3	160	15	-56.92

When considering hardware complexity, three test cases are taken into account from Table 5.5: Case 1, Case 4 and Case 7. Fig.5.11, Fig.5.12 and Fig.5.13 report

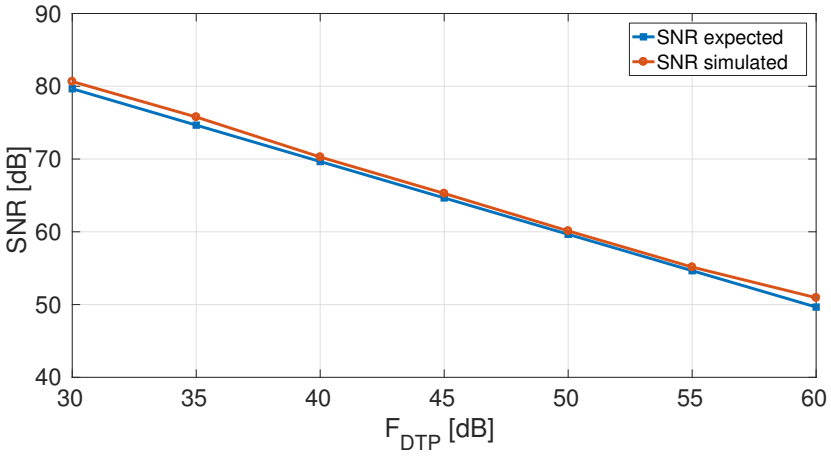

Figure 5.10: SNR versus noise figure of the UUT as obtained from both model and simulations.

the hardware complexity for the three cases, respectively. Also in this case the plots show a very good agreement between model and simulation.

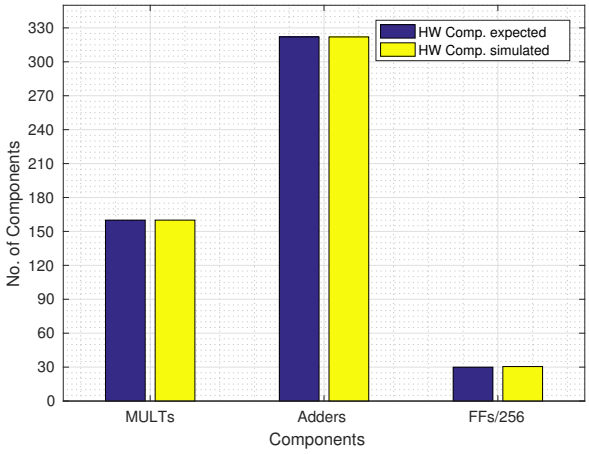

Figure 5.11: Hardware complexity (Case 1).

5.4 VHDL Simulation: Power Estimation

This section reports the tests performed with the FPGA Virtex-5 of the Xilinx family in order to compare the power estimation provided by our model and the

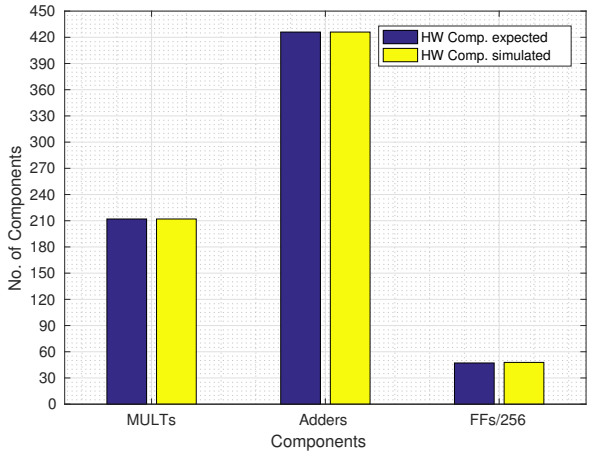

Figure 5.12: Hardware complexity (Case 4).

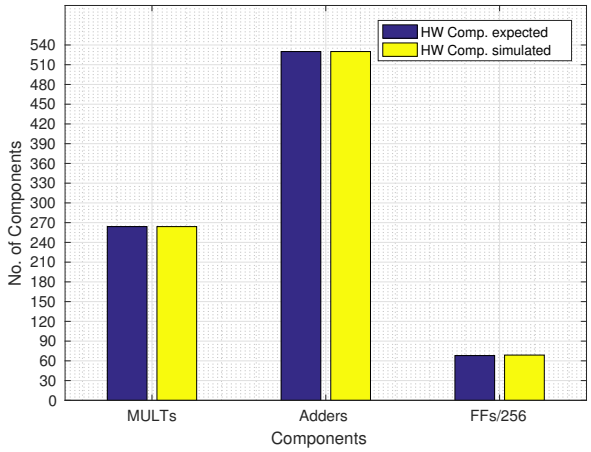

Figure 5.13: Hardware complexity (Case 7).

Vivado Power Analysis tool. The main goal at this preliminary stage is to compare the power trend between the model and Vivado Power Analysis. In this way, the model can be used to compare, in term of power consumption, different configurations of the DTP. In order to perform the power analysis with Vivado Power Analysis the following parameters must be defined: Static Probability and Toogle Rate. Static probability defines the percentage of the analysis duration during which the considered element is driven at a high logic level. Static probability is also referred to as percentage high. For example, if a signal is at Logic 1 for 40ns in a duration of 100ns, the static probability = 40/100 = 0.4. Toggle rate (%) is the rate at which the output of a synchronous logic element switches compared to a

given clock input. It is modeled as a percentage between 0 - 200%. A toggle rate of 100% means that on average the output toggles once during every clock cycle. A toggle rate of 200% implies that the output toggles twice during every clock cycle, changing on both rising and falling clock edges, and making the effective output signal frequency equal to the clock frequency. For example, if a signal changes at every four clock cycles with respect to a Clock of any frequency, then the Toggle Rate is: (1/4)*100 = 25%. The following parameters of the Vivado Power Analysis are used: Static Probability = 0.5 and Toogle Rate = 50.0%. The power analysis is performed separately for the following three sub-blocks of the DTP: "IF to Analytic", VCPF and FFT.

5.4.1 IF to Analytic

The parameters of the "IF to Analytic" block for a specific hardware configuration are:

$$(n_{si},\ n_{so},\ n_h,\ n_m,\ order)$$

where n_{si}, n_{so}, n_h and n_m are the number of bits of the input signal, the output signal, the filter coefficients and the additional bits after multiplication ($n_{si} + n_m$), respectively, while $order$ is the filter order. In Table 5.6 are reported the three test cases defined in order to compare our power model and Vivado Power Analysis. In each test case only the order of the FIR filter changes. The three cases differ for the size of the data path. Fig. 5.14 shows the results obtained from the simulation with Vivado Power Analysis (the red lines) and our power estimation model (the blue lines).

Table 5.6: Test cases for the "IF to Analytic" block

	n_{si}	n_{so}	n_h	n_m	$order$
Case1	13	11	15	3	[200, 300, 400]
Case2	17	16	15	3	[200, 300, 400]
Case3	17	16	15	9	[200, 300, 400]

When comparing the results, a limited deviations between the simulation and the estimation can be observed. For each case the two lines present a different slope but show the same trend. The difference in the slope can be due at the difference between the vFPGA used to define the power model and the Virtex-5 used as FPGA of reference.

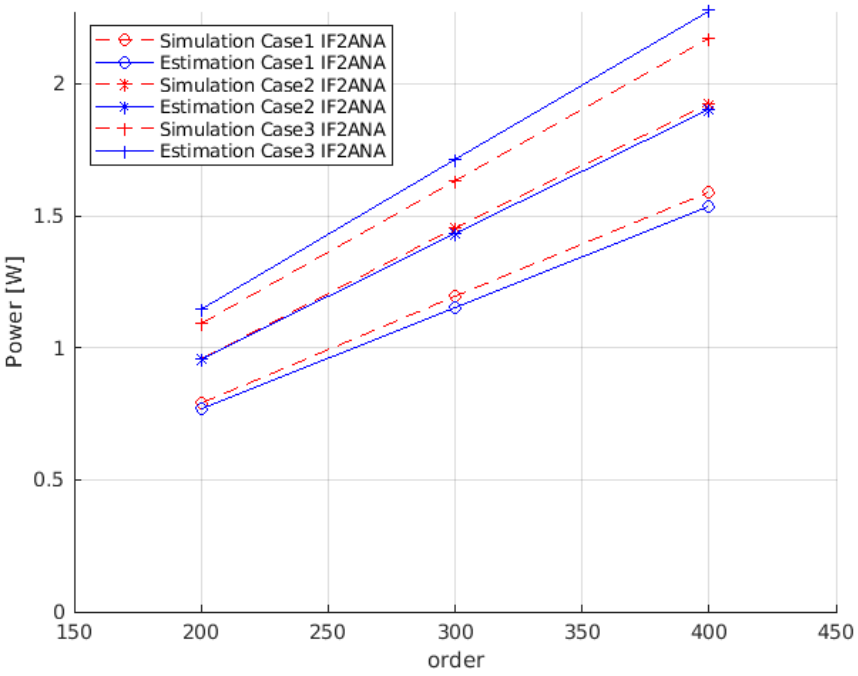

Figure 5.14: Power Estimation of the "IF to Analytic" (IF2ANA) block.

5.4.2 Variable Coefficients Polyphase Filter

The parameters of the "Variable Coefficients Polyphase Filter" block (VCPF) for a specific hardware configuration are:

$$(n_{si},\ n_{so},\ n_h,\ n_m,\ order,\ J)$$

where n_{si}, n_{so}, n_w and n_m are the number of bits of the input signal, the output signal, the filter coefficients and the additional bits after multiplication ($n_{si} + n_m$), respectively, while $order$ is the filter order and J is the number of channels in the channelizer. In Table 5.7 are reported the hardware parameters of the VCPF used to valuate the power consumption. Fig. 5.15 shows the results obtained from the simulation with Vivado Power Analysis (the red lines) and our power estimation model (the blue lines). The trend of the power is plotted to varying of J. Also for the VCPF can be observed a different slope between the two trend.

Table 5.7: Test case for the "VCPF" block

n_{si}	n_{so}	n_h	n_m	$order$	J
13	13	15	3	23	8
14	14	15	3	25	16
15	15	15	3	25	32
16	16	15	3	25	64
17	17	15	3	27	128

5.4.3 Fast Fourier Transform

The parameters of the FFT block for a specific hardware configuration are:

$$(n_{si},\ n_{so},\ n_w,\ J)$$

where n_{si}, n_{so} and n_w are the number of bits of the input signal, the output signal and the coefficients, respectively, while J is the number of channels. In Table 5.8 are reported the two test cases defined in order to compare our power model and Vivado Power Analysis. In each test case only the number of channels changes. The two cases differ for the size of the data path. Fig. 5.16 shows the results obtained from the simulation with Vivado Power Analysis (the red lines) and our power estimation model (the blue lines). In this case the different slope between the simulation and the estimation is less pronounced but not absent.

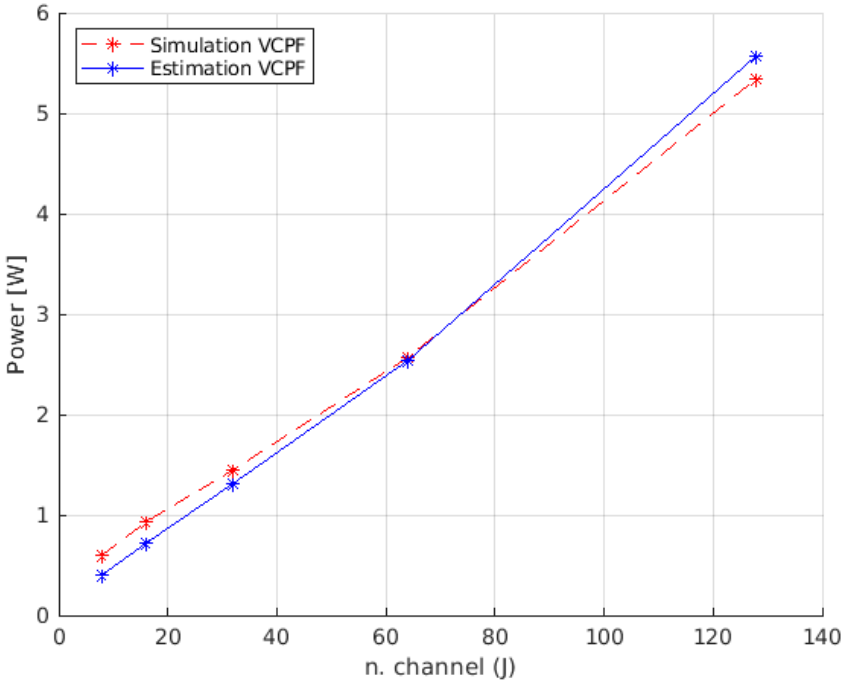

Figure 5.15: Power Estimation of the "Variable Coefficients Polyphases Filter" (VCPF) block.

Table 5.8: Test cases for the FFT block

	n_{si}	n_{so}	n_w	J
Case1	20	20	17	[8, 16, 32, 64, 128, 256, 512, 1024]
Case2	10	10	15	[8, 16, 32, 64, 128, 256, 512, 1024]

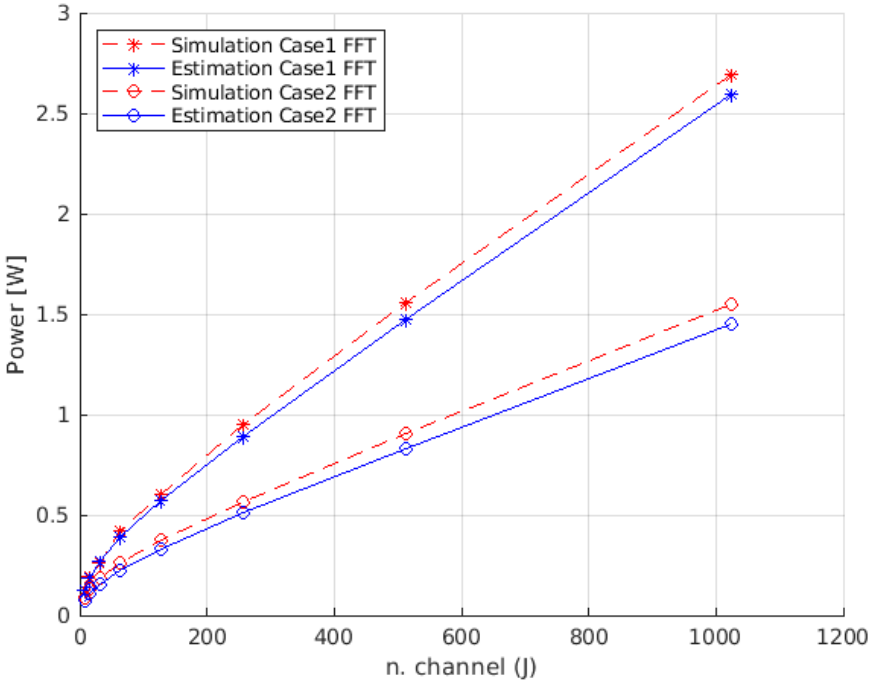

Figure 5.16: Power Estimation of the FFT block.

Chapter 6

Conclusions

In the emerging global framework for fifth generation (5G) wireless technologies, transparent satellites may be considered as an appealing solution to provide backhaul connectivity to the on-ground Relay Nodes, although semi-transparent architectures have also appeared as a viable alternative to provide broadband connectivity in modern network topologies. In this perspective, the thesis work contributed on the design of new generation transponders. In the context of an industry-driven project an explicit method to compute the HW complexity of digital transparent processors for satellite transponders has been presented. In order to support a complete design flow from system level requirements to HW design of single modules, the complexity of a DTP has been explicitly related to the overall processed bandwidth, to the selected modulation and coding schemes, and to performance degradation requirements. Rather general expressions have been derived for register sizes, data path width and number of required elementary operations for each block in the DTP. The results computed with the proposed models can be adapted to any target technology, once the nature of elementary operations has been described in detail. Four examples of design approaches have been proposed and validated: it can be seen that quite different design choices and complexities can be obtained while meeting the same link level requirements. An extension of the noise model early proposed in [13], [26], that support full exploration of the design alternatives is presented. Furthermore, in order to overcome some initial assumptions, the extended model also allows to handle different values of the power spectral density across different ESBWs and more general constraints on pass-band and stop-band in filters' design. A full FPGA implementation of a whole DTP chain has been presented in order to validate the theoretical framework on a real hardware device. In the last paper [P3], a preliminary work oriented to power estimation has been presented. In this paper a good power prediction of the model can be observed respect to the Xilinx Tools' estimations.

Ongoing work is concerned with further refinements of the models and the

formulation of an explicit optimization problem that is expected to yield a powerful framework for the careful design of advanced transponders in the 5G ecosystem. Moreover, in future works it would be interesting to consider DTP chain composed by hardware and software component in order to evaluate the trade-off between a solution entirely hardware or software.

Publications

In Reviewed Conferences Proceeding

[P1]: V. Sulli, G. Marini, F. Santucci, D. Giancristofaro and M. Faccio, "Performance modeling, design and FPGA-based validation of digital transparent satellite processors," 2018 IEEE Aerospace Conference, Big Sky, MT, 2018, pp. 1-11.

[P2]: V. Sulli, G. Marini, F. Santucci, G. Battisti and M. Faccio, "Performance and Hardware Complexity Trade-offs for Digital Transparent Processors in 5G Satcoms," 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2019, pp. 1-9.

[P3]: G. Marini, G. Battisti, V. Sulli, F. Santucci and M. Faccio, "Modelling of Power Consumption in FPGA Implementation of Digital Transparent Processors for 5GSAT," scheduled.

Journals

[P4]: V. Sulli, D. Giancristofaro, F. Santucci, M. Faccio and G. Marini, "Design of Digital Satellite Processors: From Communications Link Performance to Hardware Complexity," in IEEE Journal on Selected Areas in Communications, vol. 36, no. 2, pp. 338-350, Feb. 2018.

Bibliography

[1] O. Kodheli, A. Guidotti, and A. Vanelli-Coralli. Integration of Satellites in 5G through LEO Constellations. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, Dec 2017.

[2] 3rd Generation Partnership Project (3GPP). Technical Specification Group Radio Access Network. Study on New Radio (NR) to Support Non Terrestrial Networks (Release 15). Technical report.

[3] 3rd Generation Partnership Project (3GPP). Technical Specification Group Radio Access Network. Propagation delay and Doppler in Non-Terrestrial Networks (Release 15). Technical report.

[4] 3rd Generation Partnership Project (3GPP). 5G: Study on Scenarios and Requirements for Next Generation Access Technologies. Technical report.

[5] G. Giambene, S. Kota, and P. Pillai. Satellite-5G Integration: A Network Perspective. *IEEE Network*, 32(5):25–31, Sep. 2018.

[6] A. Guidotti, A. Vanelli-Coralli, M. Caus, J. Bas, G. Colavolpe, T. Foggi, S. Cioni, A. Modenini, and D. Tarchi. Satellite-enabled LTE systems in LEO constellations. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 876–881, May 2017.

[7] C. Haardt and N. Courville. Internet by satellite: a flexible processor with radio burst switching. In *2006 International Workshop on Satellite and Space Communications*, pages 58–62, Sep. 2006.

[8] P. Angeletti, R. De Gaudenzi, and M. Lisi. *From "Bent Pipes" to "Software Defined Payloads": Evolution and Trends of Satellite Communications Systems*.

[9] G. Tomasicchio, A. Pisano, D. Giancristofaro. New satcoms architectures with reconfigurable digital on board processing technologies. In *2nd ESA Workshop on Advanced Flexible Telecom Payloads*, April. 2016.

[10] S. Rahimi and B. Champagne. Perfect reconstruction dft modulated oversampled filter bank transceivers. In *2011 19th European Signal Processing Conference*, pages 1588–1592, Aug 2011.

[11] Håkan Johansson and Per Löwenborg. Satellite-5G Integration: A Network Perspective. *EURASIP Journal on Advances in Signal Processing*, Dec. 2006.

[12] H. G. Gockler, A. Groth, and M. N. Abdulazim. Parallelisation of digital signal processing in uniform and reconfigurable filter banks for satellite communications. In *APCCAS 2006 - 2006 IEEE Asia Pacific Conference on Circuits and Systems*, pages 1059–1062, Dec 2006.

[13] V. Sulli, D. Giancristofaro, F. Santucci, and M. Faccio. An analytical method for performance evaluation of digital transparent satellite processors. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7, Dec 2016.

[14] V. Sulli, D. Giancristofaro, F. Santucci, and M. Faccio. Computing the hardware complexity of digital transparent satellite processors on the basis of performance requirements. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2017.

[15] P. Gabellini, N. Gatti, G. Gallinaro, D. Giancristofaro, and P. Angeletti. Proposed reference system scenarios for performance assessment of a high-throughput highly-reconfigurable bent-pipe processor for access networks. In *ESA Workshop Adv. Flexible Telecom Payloads (ESA/ESTEC)*, pages 1–7, 2008.

[16] R. E. Crochiere and L. R. Rabiner. *Multirate Digital Signal Processing*. Prentice-Hall, 1983.

[17] Heinz G. Göckler and Helmut Eyssele. Study of on-board digital fdm-demultiplexing for mobile scpc satellite communications. *European Transactions on Telecommunications*, 3(1):7–14, 1992.

[18] Ronald E. Crochiere and Lawrence R. Rabiner. *Multirate digital signal processing*. Prentice-Hall, 1983.

[19] F. Wu, N. Palomo and R. Villing. FPGA realization of GDFT-FB based channelizers. In *27th Irish Signals and Systems Conference (ISSC) 2016*, pages 1–6, 2016.

[20] M. Iwabuchi, K. Sakaguchi and K. Araki. Study on multi-channel receiver based on polyphase filter bank. In *2nd Int. Conf. Signal Process*, pages 1–7, 2008.

[21] Y.Gao. Hardware implementation of a 32-point radix-2 fft architecture. Master's thesis, Lund University, 7 2015. http://www.eit.lth.se/sprapport.php?uid=856.

[22] L. Santhosh, A. Thomas. Implementation of radix 2 and radix 2^2 FFT algorithms on Spartan6 FPGA. In *4th International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 4–6, 2013.

[23] T.Neu. Direct RF conversion: From vision to reality. Technical report, Texas Instruments Incorporated, 2016.

[24] W. Kester. ADC Noise Figure - An Often Misunderstood and Misinterpreted Specification. Technical report, Analog Devices, 2014.

[25] J. Hormigo and J. Villalba. Optimizing dsp circuits by a new family of arithmetic operators. In *2014 48th Asilomar Conference on Signals, Systems and Computers*, pages 871–875, Nov 2014.

[26] V. Sulli, F. Santucci, M. Faccio, and D. Giancristofaro. Performance of satellite digital transparent processors through equivalent noise. *IEEE Transactions on Aerospace and Electronic Systems*, 54(6):2643–2661, Dec 2018.

[27] A. Lakshminarayana, S. Ahuja, and S. Shukla. High Level Power Estimation Models for FPGAs. In *2011 IEEE Computer Society Annual Symposium on VLSI*, pages 7–12, July 2011.

[28] Adel S. Sedra and Kenneth C. Smith. *Microelectronic Circuits*. Oxford University Press, 2010.

[29] A. Rico-Alvariño, A. Tato, and C. Mosquera. Robust adaptive coding and modulation scheme for the mobile satellite forward link. In *2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 530–534, June 2015.

[30] S. Benedetto, R. Garello, G. Montorsi, C. Berrou, C. Douillard, D. Giancristofaro, A. Ginesi, L. Giugno, and M. Luise. MHOMS: high-speed ACM modem for satellite applications. *IEEE Wireless Communications*, 12(2):66–77, April 2005.

[31] G. Maral, M. Bousquet. *SATELLITE COMMUNICATIONS SYSTEMS*. John Wiley Sons Ltd, 2009.