Efficient fixed-point refinement of DSP dataflow systems

Erwan Nogues*, and Daniel Menard*
*UMR CNRS 6164 IETR Image Group, INSA de Rennes
Email: {erwan.nogues,daniel.menard}@insa-rennes.fr

Abstract—With the current extensive deployment of digital communications, new standards are required every few years to regularly provide with new features. More throughput and better radio coverage w.r.t. former standards are examples of mandatory improvements. Generally, a new standard consists in modifying elements of the systems incrementally: add a receive antenna, use higher order modulation, etc. The design methodology is then crucial to ensure system quality while maintaining a short time for delivery. This paper proposes to use dataflow modelling for its ability to represent complex systems at a high level of abstraction. The dataflow representation inputs a 2-step incremental design method that aims at ensuring perfect performance independence to run simulations in parallel and keep the time-to-design reasonable.

I. INTRODUCTION

With the explosion of digital communications in the late 90’s, more throughput is constantly required for the telecommunications operators to provide more service to the end-customer. Indeed the content of the mobile internet is growing even exponentially to reach the market demand. The total mobile data traffic was of 1 million of terabyte per month [1] in 2012. In 2015, the data will reach 6 millions of terabyte per month. Hence new technologies are needed to provide more throughput. On the other end, to ensure fast penetration of these new technologies, the initial system evolves incrementally with new access techniques, new modulations, more antennas, higher interference cancellation and so on. Currently pervasive in mobile networks, the 3GPP has studied improved features for 3G networks and proposed the High Speed Downlink Packet Access (HSPDA) technology. Type 0 is the baseline receiver also called RAKE receiver and was the first to be developed. From this receiver, several enhancements were proposed to the latest Multiple Input Multiple Output (MIMO) receiver. Type 1 receiver is a RAKE receiver with two antennas, type 2 receiver uses an channel equalizer on one antenna, type 2i is the evolution of type 2 with interference awareness. Type 3 receiver is a two-antenna using a two-branch equalizer. Finally the type M receiver uses the MIMO technique. A new receiver has been proposed every year since 2006.

To obtain low time-to-market, component-based design is used. In this reuse-based approach, new processing blocks are added at each release. High accuracy and fast simulations runtimes are then mandatory at the design phase for a fast turnaround. In many cases, the system design phase is crucial as some important decisions for the final product shall be taken. At a high system level, the nature of the different key blocks varies: hardware accelerators, dedicated processing units or co-processors. To keep a fast design phase, they cannot be changed. A rigorous classification is necessary to have an accurate picture of the processing block complexity.

Pure language based approaches (C/C++ or MATLAB algorithms) fail at these challenges as they do not constrain the modelling approach enough to improve implementation and simulation productivity. Very popular on system designer side, dataflow modelling techniques provide a high level representation and help describing hierarchically the application. This model of computation is commonly used for telecommunications systems [2], [3]. In the previous work several methods were proposed to design at a system level. Several tools are available to design systems at top level [4] where dataflow modelling is mainly used to explore multiple design scenarios in a short period of time.

However algorithm designers must also quickly and easily define the mathematical representations that meet the design’s requirements. Fixed-point refinement is a crucial design step to reach an efficient implementation. The implementation cost (area, energy consumption, etc.) is minimized as long as the application quality criteria like Bit-Error Rate is fulfilled. Fixed-point simulation is thus then critical as it defines the actual feasibility of the system. Application quality evaluation is one of the most difficult parts of fixed-point refinement. Analytic approaches are very efficient in terms of execution time but are limited to smooth systems. At the opposite, fixed-point simulations can lead to long execution times but can handle any systems. Thus, these methods can be used only if the optimization algorithm strictly limits the number of iterations. Numerous word-length optimization have been proposed. Stochastic approaches like genetic algorithms [5], simulated annealing [6] or Greedy Randomized Adaptive Search Procedure (GRASP) [7] lead to results closed to the optimal solution but at the expense of long execution times due to the great number of iterations. Deterministic approaches
based on greedy algorithms [8]–[10] reduce the execution time but also the solution quality. Even for these deterministic approaches, the number of iterations is still important. System level approach have been proposed in [11] and [12] to reduce the complexity by reducing the number variables in the optimization process. Nevertheless, the number of iterations is still high. Thus, in the context of industrial applications, like mobile phone receiver where several hours a required to evaluate the application quality with fixed-point simulations, fixed-point refinement must limit dramatically the number of application quality evaluations.

In practical applications, testing 5000 frames of the system takes 4 hours to simulate one case of the standard [13]. Because 10 frequency bands and 8 use cases need to be tested, then one fixed-point refinement takes $4 \times 10 \times 8 = 320$ hours or 13 days. Assuming a system where only 2 interfaces needs to be sized in fixed-point, testing 5 bit-configuration takes 133 days. Running simulations in parallel is then a key feature to achieve reasonable simulation times, e.g. 1 day on 133 computers.

In this paper, an efficient hierarchical approach based on dataflow modelling is proposed for fixed-point refinement. Dataflow modelling is commonly used for application quality analysis and system design. Fixed-point information is added on top of its representation and describes a 2-step methodology to design the new blocks of a new application release. This efficient methodology is quality driven and guarantees that once a data is sized, no loop-back is necessary in further design steps. It can also be done in parallel for practical implementation in the industrial field. This approach limits drastically the number of iterations in the fixed-point refinement process. After presenting the related work, the dataflow representation of the system is given introducing the 2-step methodology. In section IV, the study is applied on a 3GPP type 2 receiver from a type 0 receiver. Section V presents the simulation results.

II. RELATED WORK

Regarding dataflow modeling, Geilen et al. [2] describe a systematic top-down design approach. It uses high-level system dataflow modelling tools so as to investigate architectural alternatives for the telecom front-end. It can explore system tradeoffs and find a optimized, for example, analog-digital partitioning.

Dataflow modelling can be used to characterize a telecommunication application in terms of execution latency [14]. It can also serve as an input for memory optimization. Desnos et al. [15] use a hierarchical Synchronous Data-Flow modelling to define the boundaries of shared memory requirements of Multiprocessor System-on-Chip (MPSoC) in early stages of development. The technique is used on telecommunications applications and is useful for the system designers to determine the memory budget of the final SoC.

In [12], a complete system development methodology is described for a telecommunications system. It focuses mainly on Matlab-to-hardware design flow and is partly manual and partly tool assisted. However the tools for fixed-point designing are not explicitly cited.

For fixed-point definition, Parashar et al. [11] proposed a mixed framework which incorporates both simulation and analytical approaches to design the fixed point part of complex systems. A single noise source model for characterization of the output noise at the subsystem level is used to interoperate between simulation and analytical modelling. The optimization technique is intended to be applied recursively across all the levels of the system until it is small enough to be handled by the conventional wordlength optimization algorithms.

In our approach, we use the advantage of dataflow representation to select the key blocks to focus on at the design phase. We use a pragmatic approach driven by quality to find quickly the appropriate data width at different levels: interface level and processing functions level. We also use a unique test bench.

III. DATAFLOW REPRESENTATION OF THE SYSTEM

This section presents the dataflow modelling as an efficient method to design the interfaces in the system. After describing the data-size refinement in fixed point, a methodology is proposed to address system design in a two-step manner to ensure quality and reduced simulation time. The dataflow is efficient to get rapidly a high level description of complex systems. It is used to identify at a high level of abstraction where the data is transmitted between processes and the type of information that is conveyed. Hence it is very popular with system designers as it can be used to explore architectural strategies.

Moreover, the dataflow chart provides a precise information of the data path of the system. In that sense, the data width in between actors is shown in a generic model on Fig. 1. With only a functional description of the processing actors Proc 1, Proc 2 and Proc 3, all the interfaces can be defined with respect to a quality requirement (error rate, footprint, etc.).

A. Fixed point refinement

Fixed-point refinement aims at exploring the trade-off between the implementation cost and the application quality. To illustrate this concept, the evolution of the quality criteria according to the word-length of a variable located in the application is depicted in figure 2. When the word-length is high, the application quality is equal to a reference value. The decrease of the word-length up to the value $w_1$ leads to a very slight degradation of the application quality. For this
interval, this variable have negligible influence on the global application quality. Between the word-length \( w_1 \) and \( w_2 \), the application quality evolution can be assimilated as a bend offering a good trade-off between implementation cost and application quality. When the word-length is lower than \( w_2 \), the application quality degradation is high. For this interval this variable becomes predominant on the application quality degradation. Classical refinement fixed-point explores the word-length between \( w_1 \) and \( w_2 \) and the aim is to budget the global degradation between the different variables. This requires to test numerous combinations to select the best trade-off. In the proposed method, the aim is to remain close to the word-length \( w_1 \) and not to explore the bend between \( w_1 \) and \( w_2 \) so as to reduce dramatically the number of iterations compared to classical word-length optimization. In our approach the different variables of the optimization process can be processed separately. The aim is to find for each variable the value \( w_1 \) which leads to a very slight degradation of the application quality criteria. The obtained solution is probably pessimistic but it guarantees that the fixed-point refinement cannot affect other variables. Hence, simulations can run in parallel.

B. Two-step approach for fixed-point refinement of the system

The two-step approach focuses first on the new interfaces to design. Several methods exist to optimize the data widths and rely on the fixed-point refinement. As stated in section III-A, the word-length sizing during fixed-point refinement can be computed for different strategies: either guarantee a minimum quality degradation (when the word length is greater than) \( w_1 \) or define an optimized word length for a given application quality. Nguyen et al. [7] show that the latter method can lead to numerous simulations with no predefined number of simulations due to data dependency. We choose the first approach as it can guarantee the system data independence leading to the predictable number of simulations. These simulations can run independently in parallel.

During this latter step, the floating point representations of the kernel blocks are kept with their floating point representation where no quality degradation is expected.

In the second step, the processing function is sized regarding fixed-point refinement. This operation consists in defining the fractional part first and then the integer part of internal variables. The criterion to fulfill is similar to the one of the previous section. The simulations is stopped once the \( w_1 \) of figure 2 is reached for the fractional part and then for the integer part.

IV. APPLICATION TO RECEIVER DESIGN

In a typical communication system, the physical layer is at the intersection of the radio-frequency domain and protocol stack levels. The physical layer is in charge of presenting to the upper layers the information bits after demodulating and correcting the received samples. For system design, it consists in assembling different functional blocks. In most cases, the systems can reuse previous units to speed-up the design phase and concentrate all the efforts on the new processing blocks.

A generic block diagram of the Wideband Code Division Multiple Access (WCDMA) receiver is given in Fig. 4.

The system is composed of three sub-systems:
- dig RF is the radio frequency front-end in charge of receiving and converting the samples in to the digital baseband interface.
- WCDMA receiver is in charge of demodulating and equalizing the received baseband samples
- Outer block performs demapping of the modulation and the channel decoding so as to remove all the possible errors.

The WCDMA receiver is made of different processing blocks. The Pilot Processing (PIPR) provides the fine channel estimates to the RAKE receiver. The delay unit stores the samples with the correct delay values per finger of processing for the RAKE receiver. The RAKE receiver utilizes the multipath
estimation to perform a coherent reconstruction of the signal. The Despreader (DESP) unit despreads the received chips from the spreading sequence generated Pseudo Noise Generator (PNGN). After the wide deployment of WCDMA, more sophisticated devices needed a higher level of throughput. New modulations and new receiving techniques are introduced in standards with HSDPA. Several types of advanced receivers were defined to classify their topology:

- Type 0 is a RAKE receiver using one antenna
- Type 1 is a RAKE receiver using two antennas
- Type 2 is a chip level equalizer using one antenna

Different sets of requirements were successively introduced in the requirements specification [13]. In figure 5, the classic topology of the input signal for the receiver is depicted. The received signal is composed of the symbols of interest, the same symbols after being filtered by the multipath reflection and the noise coming from other cells. The receiver type 0, also known as the RAKE receiver, has the main advantage of being simple to implement because the different paths of propagation are handled individually [16]. However it is suboptimal in case of inter-symbols interference and for high geometry factor. In order to improve the quality of the receiver in those cases, the receiver type 2 uses a chip level equalizer. The purpose is to reduce the Intersymbol interference (ISI). For high Signal-to-Noise Ratio (SNR) it is equivalent to zero forcing equalizer and for low SNR it is equivalent to the RAKE receiver ensuring the backward compatibility.

This paper focuses on the evolution of the receiver from type 0 receiver to a type 2 receiver using the proposed methodology.

A. Upgrading the RAKE receiver to an advanced HSDPA receiver architecture

The design methodology of section III is used as it secures the quality of the system while maintaining a limited number of simulation steps for fast design. A major step of the flow described in Fig. 1 is to describe the system using the dataflow mechanisms. This high level representation allows to identify which blocks can be re-used when upgrading the system. It also allows to focus on the new parts to design and check if they can correctly fit into the new system.

- Levinson-Durbin solver
- Chip equalizer

The equalizer is a linear equalizer using a Finite Impulse Response (FIR) structure. The processing can be summarized as additions and multiplications which is simple to describe in fixed-point representation. The Levinson-Durbin computes the filter taps of the equalizer [17]. It needs to support more complicated operations such as division and is more critical to translate into fixed-point.

![Figure 6. HSDPA receiver](image)

Besides, the connecting interfaces of these actors are also to be designed as their bit-width is not known.

B. Channel equalization

As stated in Fig. 8, the wanted signal $d_n$ needs to be retrieved from a received symbol $r_n$ (equation 1).

$$r_n = h_0.d_n + \sum_{k=1}^{L_h} h_k.d_{n-k} + w_n$$

where $h_0.d_n$ is the symbol of interest, $\sum_{k=1}^{L_h} h_k.d_{n-k}$ is the ISI and $w_n$ represents the thermal noise and the extra cell noise. From this received sample, the equalizer taps [18] can be defined as follows in equation 2.

$$\hat{d}_n = \sum_{k=1}^{L_{p-1}} p_k + r_{n-k}$$

The goal is then to find the linear equalizer and its taps \{p_0, p_1, \ldots, p_{L_p-1}\} which minimises the Mean Square Error (MSE): $MSE = E[|d_n - \hat{d}_n|^2]$. The received samples at the input of the equalizer can be rewritten in a matrix form (figure 7) with introducing the $H$ matrix.

![Figure 7. Equation with a matrix presentation](image)

After transformations, it can be proved that equation 2 can be rewritten as complex linear system of the form: $Rx = b$ where $R = H^H H + \frac{\sigma_w^2}{\sigma^2} I$ with $h = [h_0 h_1 h_2 \ldots h_{L_p}]$, $\sigma_w^2$ is the gaussian noise power and $\sigma^2$ is the multi-user signal power. The $R$ matrix is a Toeplitz-Hermitian matrix and can...
be solved using the Levinson-Durbin algorithm. After solving the system, the filter tap are provided to the linear equalizer for chip level equalization removing the ISI of the received samples.

C. Application quality requirements

The proposed method tests the complete system at each step. The use case description is crucial to ensure that each step is completed and can move to the next one. In the example of a receiver, the exhaustive list of the use case is made available by the 3GPP standard groups and can be checked thoroughly. The requirements specifications [13] define several configurations of multipath channel types (Fig. 8).

C. Application quality requirements

The proposed method tests the complete system at each step. The use case description is crucial to ensure that each step is completed and can move to the next one. In the example of a receiver, the exhaustive list of the use case is made available by the 3GPP standard groups and can be checked thoroughly. The requirements specifications [13] define several configurations of multipath channel types (Fig. 8).

V. SIMULATIONS RESULTS

The simulations were conducted respecting the requirements of [13]. At each step, the achieved performance are compared to the requirements. If the performance is acceptable, the design flow can move to the new step.

A. First step: designing the interface sizes

In this section, we will focus on the interface definition between the equalizer and the despreader. The processing actors use floating point representation and the goal of this step is to size the data interface. We assume that the other interfaces are already designed as they can come from the previous generation of the receiver, i.e., the WCDMA receiver. Fig. 9 shows that the interface between the equalizer and the despreader needs to be designed.

Figure 9. Equalizer quantization

The Levinson-Durbin processing actor and the equalizer are run with the floating point version as proposed in the methodology. The complete system is tested with only one unknown variable: N, the number of bits of the interface between the equalizer and the despreader. All the use cases are tested and the results are checked versus the system requirements. In Fig 10, the results are given for two levels of interference: -6 dB and -3dB [13].

From Fig. 10, it can be seen that the quality reaches a limit from where the quality is not better with more representation bits. This stands for \( w_1 \) of Fig. 2 and confirms the assumption of III-B. In this particular case, 13 bits are sufficient to complete the system requirements.

B. Second step: designing the kernel processing

As stated in IV-B, the Levinson-Durbin processing unit is not trivial to translate into fixed-point representation. The interfaces are now fixed from the first step.

Figure 10. Application results with different interface sizes
Two dimensions need to be computed: the number of necessary bits for the integer part and the fractional part. In practice, the fixed-point refinement is performed using ac fixed C++ class [19]. This class proposes a 32-bit representation where the fractional part is noted by Y in the Fig. 11.

From this notation, the basic operations that are needed by the processing can be derived as a function of Y for the addition, the multiplication and the division as described in Fig. 12.

The first run of simulation consists in trying different Y values from 10 up to 18 and are run in parallel. From the results in Fig. 10, the $w_1$ value of section III-B is reached when the target margin of 2.3 dB is reached. It shall be noted that the best margin is similar to the first step. The results in table I show that 14 bits is sufficient for the fractional part.

When optimizing the integer part, the same principle applies. As a consequence, the same tests were run again targeting the same margin. The average margin of 2.32 dB can be achieved with 20 bits on the integer part as seen in the results of table II.

### VI. CONCLUSION

Ensuring at the very early stage of a design that it will be compliant to its specifications is a complex task and requires advanced de-risking methods. A new system is usually merely an upgrade of a legacy one but the specification compliance typically needs to be completely recomputed as the system is upgraded. In this paper, a hierarchical dataflow representation is used to describe a complex system at a high level of abstraction. The dataflow representation inputs a 2-step incremental design method that can ensure perfect compliance to the requirements. We show how this method is perfectly suited for fixed-point refinement of the new actors. We guarantee the maximum quality after sizing with a constant control of the number of simulations with no loop-back to previous steps.

### REFERENCES


