

Designing a DS-CDMA system over FPGA platforms

X. Revés; A. Gelonch; F. Casadevall

Universitat Politècnica de Catalunya, Dept. of Signal Theory and Communications

Jordi Girona 1-3, 08034 Barcelona (Spain)
{xreves, antoni, ferran}@xaloc.upc.es

Abstract

Digital technologies are modifying the way that physical layers are implemented. The FPGA emerging devices provides to the designer the possibility to modify the hardware functional qualities without giving up high performance. This is a very interesting quality that can help building the so called Software Radios. Following this idea, in the letter a DS-CDMA low cost mobile communications system capable to operate in picocellular (e. g. Indoor) environments is presented. All the digital signal processing from intermediate frequency to base band has been implemented using a SHaRe platform with nine Xilinx 4K family FPGA chips for a maximum gate count beyond 2 million logic gates.

Key words

Software Radio, FPGA, CDMA, Indoor WLAN, hardware implementation.

1. Introduction

In the design of digital radio terminals many aspects have to be considered. Now we are close to third generation mobile systems deploying. During a period of time third generation will share the air with second generation and its derivatives. Then one can expect a transition phase where multimode terminals will be common. This period of time will place constraints to the way that radio terminals are built. The Software Radio approach [1] could solve part of the problems that this evolution imposes.

Many are the implications related to multimode terminals and these can be found in every network layer. When focusing on the physical layer implementation, the available and affordable technologies must be considered. Software Radios ideally consider a fully digital transmission and reception chains which require powerful devices able to cope with high computational demands which are specially found in the intermediate frequency (IF) stages.

In a receiver (similarly in a transmitter), after analog to digital conversion, which is a key point in software radio terminals, is time to process the raw digital data. Several kind of devices are offered at the market ranging from partially programmable ASICs (Application-Specific Integrated Circuit), that provide a limited set of configurable blocks to cope with different radio interface formats, to the digital signal processors (DSPs) based on a central processing unit (CPU). Both have different areas of application, while the first are mainly devoted to translating signal from IF to base

band with the corresponding channel selection, the second ones are used for typical base band tasks of demodulation, decoding, etc. It is not a good solution to exchange the position of these devices within the reception chain. Another approach that allows mixing all the stages within a reception chain would be using FPGA (Field Programmable Gate Array) technology to implement IF and base band stages, then being a good candidate for software radios [2]. This approach is valid because FPGAs can handle the high speed demanding operations of channel selection and frequency translation as well as the, in general, more complex tasks related with base band processing. In general it is assumed to use FPGAs in high co-operation and co-ordination with a DSP. But this is not strictly required because FPGAs can themselves incorporate, if required, a standard or a tuned CPU core. Then, FPGAs can be understood as general devices where any digital function can be implemented (obviously there exist limitations). Independently of the capacity of FPGAs to handle different types of processing algorithms, they offer programming flexibility and high performance at the same time. But, of course, a different programming methodology must be used compared to the one used in ASICs or DSPs.

In this paper we will deal about implementation of digital stages of an indoor DS-CDMA system focusing on particular architectures well suited for FPGA devices, providing significant figures about resource utilisation. The main goal resides in identifying adequate solutions for each constituent part of the system, on the basis of the predefined architecture of an FPGA, in order to minimise the resource allocation.

2. General system description

2.1. System architecture

The DS-CDMA [3] indoor system presented here, which works at the 2.4GHz frequency band, has an star architecture where all the terminals interact with its own base station that has the transmission, reception and control capabilities, as user control, channel assignment, code distribution, power control, etc. Each base station can bear up to 64 different channels of 32kb/s simultaneously assigned to a maximum of 16 different users. About the user terminals we distinguish a voice terminal, using DPCM (Differential Pulse Code Modulation) at 32kb/s similar to that used in DECT (Digital European Cordless Telephone), a data terminal at 9.6 kb/s and delays below 30 ms and a video terminal, using standard H261 which is capable to adapt its rate up to 2Mbits/s depending on the image quality desired.

The CDMA radio link will use traffic channels (TCH) to carry data, voice and video. These channels will be complemented with associated control channels (ACCH) for power control, channel information, status of communication,

etc. Other signalling channels used are the pilot channel (PICH), to allow correct synchronism of the mobile to the network, the broadcast paging channel (BPCH), for paging, access to the network or handover, and the random access channel (RACH), to allow the mobile asking a traffic channel.

The parts of this system implemented using an FPGA approach range from the physical channels to intermediate frequency. It does not include channel coding, puncturing, interleaving, etc. of information coming from upper layers. The service that the FPGA-based layer provides is a simple raw service of transport of data over the radio channel.

2.2. Down link structure

Down Link (see Figure 1) is mainly based on an orthogonal multiplexing of information and control channels using pseudonoise sequences. The sequences used to multiplex channels are called Walsh or Hadamard sequences and those used to isolate adjacent cells are called GOLD sequences. Each user has a maximum of 4 QPSK-CDMA channels each of them at 64kb/s separated by different Walsh sequences at 1024Mchips/s. This provides a maximum of 256kb/s per user which is the limit assumed in the system.

In transmission two separated antennas are used to transmit twice the same signal with a determined delay between them longer than the CDMA chip resolution. In the receiver side a RAKE structure can take advantage of this transmission diversity. To simplify the receiver, a pre-RAKE [4] structure is included in the transmitter (base station) for each user which is adjusted using the channel estimation values obtained by the mobile terminal.

2.3. Up link structure

In this case a simple QPSK scheme was used for transmission where the maximum speed for both, the inphase and quadrature channels, was established in 128 kb/s. This data rate was spread using GOLD sequences up to 4096 kchips/s, obtaining a processing gain of 32 when transmitting at the maximum speed (256 kb/s total). To simplify the design an asynchronous access was implemented, then a synchronism stage in the base station receiver must be implemented. In the Figure 2 the FPGA sections of the Up Link transmitter and receiver schemes can be observed.

3. Physical implementation

3.1. Block optimisation

Figure 1 and Figure 2 give a general view of the blocks implemented using the FPGA approach for up and down links of the base station and mobile terminal respectively. In each transmitter the information bits are given to the FPGA which generates a QPSK modulation centred at 8192kHz and sampled at 32768kHz. Inversely, in the receiver the bits are obtained after synchronisation, frequency adjust, channel estimation, etc.

As can be observed some blocks in the down link will be similar to those in the up link. But when each block is optimised to reduce the resources required to implement it, reusing the block is not possible. Nevertheless, what is important is to identify which structure better fits to that type of block. What can certainly be reused is the way a function is implemented in every case. Then the concrete implementation can be adjusted in a more or less automatic way. Consider, for

instance, two different filtering stages of the down link. First, in transmission, the FIR filter is used to generate a pulse with root raised cosine shape. To this filter only one of every eight input samples is non-zero (4096kHz input zero padded and filtered to get 32768kHz output). The property can be used to reduce the number of multipliers and consequently reduce the amount of logic. Another filtering stage would be the half band filter used in the down link receiver before decimating by 2 (HB FIR). The samples entered to this filter are non-zero and then simplifications can only be done considering the decimation process and through the selection of an adequate filter with, for instance, zero-valued coefficients in every odd sample. Of course until here the simplifications are not only useful for FPGA-based implementations but also for any other kind of implementation.

On the basis of the previously stated simplifications, the FPGA implementation can have different shapes, where considerations about number of bits, number of coefficients, etc. are very important. The first filtering process mentioned can efficiently be implemented as shown in the Figure 3 where only two multipliers with their inputs multiplexed are used to compute the output. In the case of the second filtering process, the transposed FIR structure shown in the Figure 4 fits well to our purposes (and in general in most FPGA FIR implementations). Here the multipliers are implemented as distributed arithmetic multipliers taking advantage that the coefficients of the filter are constants. In both filtering cases FPGA internal RAM and fast internal adders are the key that allow an efficient implementation of the blocks.

Other examples in the transmission/reception chains can be found. After general simplifications, efficiently mapping the block onto FPGA consists mainly on identifying how the algorithm can be translated to the internal structures. The use of internal RAM/ROM and fast adders provides a good mechanism to implement typical signal processing tasks but some times require an algorithm reshaping or even a special control.

3.2. Resource utilisation

The system roughly described above, although represents a simplification with respect to commercial products, is an example that incorporates all the functions that imply a higher signal processing demand. That is, intermediate frequency processing and synchronisation algorithms [1]. The blocks that perform these tasks must use resources optimally for both reduce the amount of resources required and also to reduce the power consumption of the devices. This has been the main goal of the design.

The information here provided of resource allocation is expressed in terms of Logic Elements (LE). One LE is defined as the composition of one 4-input look up table (LUT), equivalent to a RAM of one bit wide an 16 addresses of depth, and one flip-flop. It should be noted that figures given are approximated because are extracted from CAD tools reports where several blocks are mixed and/or other support functions are considered, like μ processor access interface.

The hardware platform used to check the validity of the system is not tuned for this specific design. It is a pre-designed platform with enough flexibility to accommodate different applications. This platform is called SHaRe (**R**econfigurable **H**ardware **S**ystem) [5] which provides up to 8 user programmable FPGAs interconnected in a flexible way

with access to additional resources like RAM, ROM, FIFOs and programmable clocks. Each one of the FPGAs can be (re)programmed individually at any time by simply performing write cycles to a memory address over the host

bus (VME bus). The main processor over the bus controlling the whole system is an Sparc CPU running Solaris. This host processor provides connection to an IP network thus obtaining a higher functionality of the system.

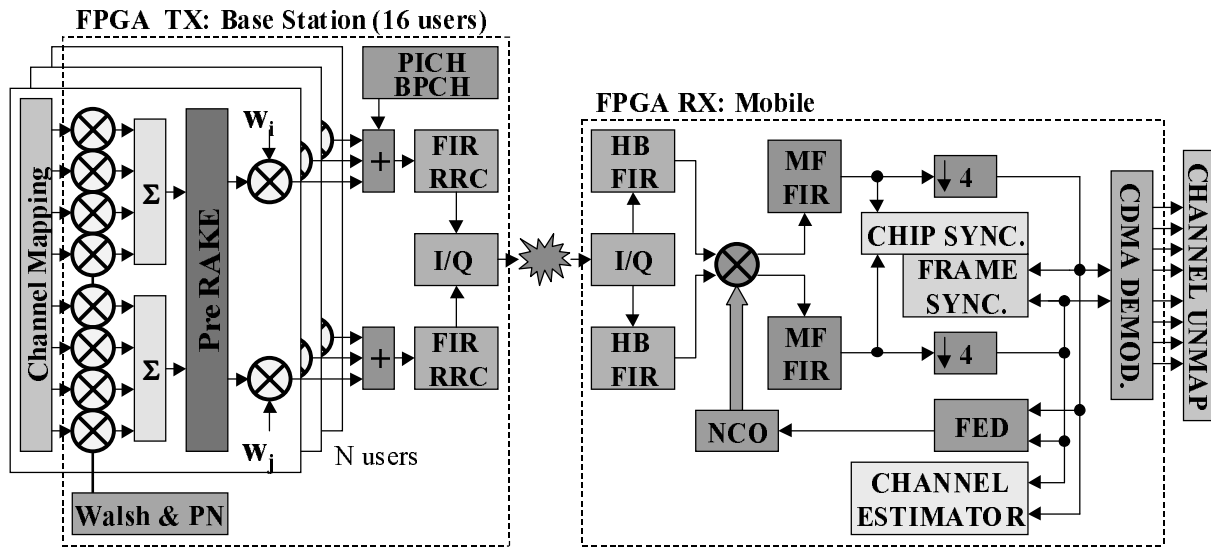


Figure 1: Down Link Transmitter and Receiver

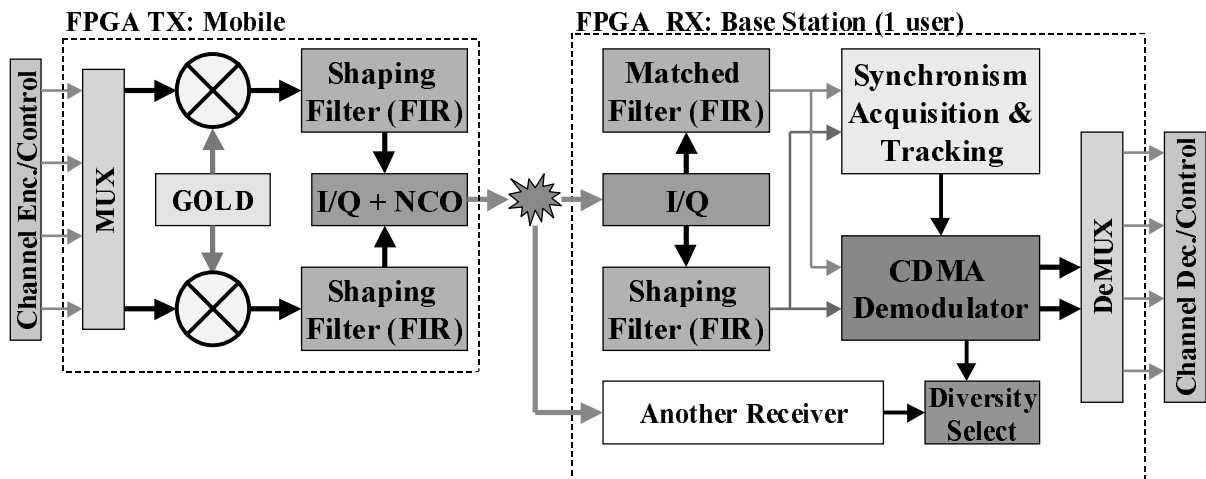


Figure 2: UpLink Transmitter and Receiver

In Table 1 and Table 2 the LE utilisation for the mobile terminal and base station is summarised. The values concern only to the specific part named. The figures show that the complete terminal system would fit into a state-of-the-art FPGA with about 7000 LEs. This was a huge FPGA some years ago, but today this amount of LEs can be found into relatively small FPGAs like a Xilinx Virtex XCV400 [6]. The base station part for a single user would need less than 9000 LEs which are also available into that FPGA. The increase of resource utilisation for more users (16 in this case) is not linear because most of the transmitter is reused, as it can be observed in the Figure 1 where IF stages are not repeated for each user. In any case, about 100000 LEs should be employed to construct this part of the base station.

The actual implementation presented is not based on Xilinx Virtex family but on Xilinx 4K family [7]. The basic elements of both families are quite similar but Virtex includes more RAM bits, more and faster interconnect logic, more input/output standards, dedicated functions (e.g. multipliers), etc., as correspond to a more modern family. As stated before, the platform used, called SHaRe, has 8 user-programmable FPGAs with a count of up to 50176 LEs (with 8 XC4085 FPGAs on the board), 2 Mbytes of SRAM distributed along the devices, up to 512 Kbytes of FIFOs and up to 512 Kbytes of ROM. The mobile terminal fits over a single board. Also the base station for one user receiver and 16 transmitters fit over a single board, but every new base station receiver requires a new board. With this distribution chosen the maximum number of resources required over a

single board is about 9000 LEs that can be obtained completing a SHaRe board with the smallest devices it can bear to achieve the total of 9216 LEs. Although the numbers fit, it is not possible to adjust the design to that level and some room must be considered to install accessories, controls and even leave some resources free to ease the process of mapping of the blocks over the devices. All the boards involved in the design together with the main system controller, a Sparc diskless board for VME bus, are joined over a VME bus to allow a correct configuration and management of the application.

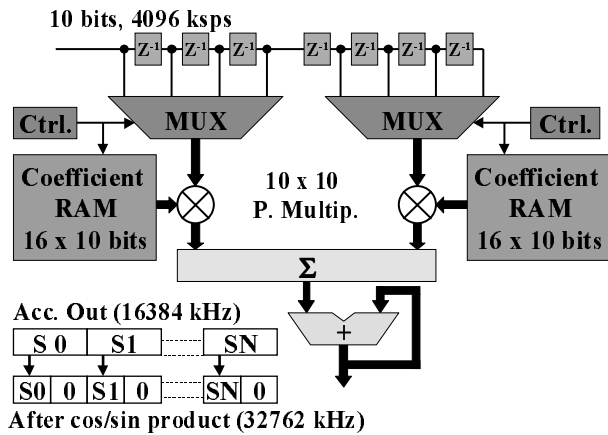


Figure 3: FIR filter with multiplexed multipliers

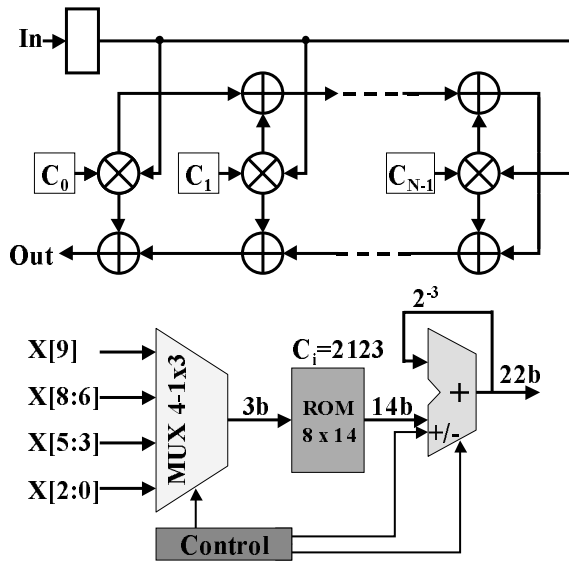


Figure 4: FIR transposed form and Block Distributed Arithmetic multiplier example

In general the blocks have been implemented exchanging area for speed. Reducing area at the cost of an increment of frequency clock does not save power consumption. Power consumption is minimised only when the mapping of a function over FPGA resources is optimally done. Power demanded by the FPGA devices depend highly on the physical properties of them but more advanced the technology used less the power required is. Of course, the flexibility provided by FPGAs over ASICs is paid as an increase of

power consumption when the FPGA option is selected. By other hand, the amount of LEs stated here is relatively low compared to the availability of them in commercial FPGAs, so a different structure with higher sampling rates can be considered simply increasing area. Note that in the digital implementation of a radio receiver only the very first filtering stages after sampling have high rates. As soon as the channel of interest has been selected the sampling rate is reduced and/or transformed to be adequate for the required demodulation.

Table 1. LEs required to implement the terminal

Mobile Terminal	
Block Description	LEs
Terminal TX spreading and root raised cosine filters	400
Terminal TX frequency adjust NCO and IF translation	620
Terminal RX I/Q down conversion with half band filter	1060
Terminal RX frequency adjust (NCO + FED)	830
Terminal RX matched filters	1450
Terminal RX chip, bit and frame synchronism	1520
Terminal RX channel estimation and CDMA demodulation	500
Total Mobile Terminal	6380

Table 2. LEs required to implement the basestation

Base Station	
Block Description	LEs
Base station TX 16 users spreading, pre-RAKE and PICH + BPCH	1140
Base station TX 16 users I/Q shaping filter and IF translation	1090
Base station RX 1 user I/Q down conversion and matched filters	4390*
Base station RX 1* user chip synchronism and tracking	1180*
Base station RX 1* user CDMA demodulation	500*
Total Base Station 16 users	99350

* Multiply by 16 to compute the total base station LE requirements

4. Conclusions and future work

There are several ways to implement a re-configurable radio terminal but one that offers a wide application flexibility, ranging from high computational intensive tasks to algorithmically complex but relatively low speed tasks, and at the same time flexibility to modify its behaviour, is the one based on FPGAs. Their increasing capacity and speed, together with the progressive reduction of power consumption make of them a good candidate to occupy relevant positions in future radio terminals designed under the Software Radio line of thought. A DS-CDMA system taking advantage of the structural properties of FPGAs has been designed to check their adaptation to that kind of application and to explore the pros and cons when designing specific parts. Like in many digital systems here appears a trade-off between speed and logic resources used: higher is the speed required, higher are the resources required to implement the functions. It has been

observed that commercial FPGAs offer the range of resources required to build a system with these features.

Designing with FPGA pre-defined structures tend to modify the way the different blocks are implemented to efficiently use resources. This, in general, makes impact on the complexity of solution because typical signal processing structures cannot be directly translated. Finding the right mechanism will improve final design in terms of die and power consumption. All that has an interesting application when considering restrictions related to fully digital radio terminals because a better adaptation to FPGA structure will allow to implement more computationally intensive tasks. Because of this, it is important to identify clearly the blocks which require a more accurate tuning and get a set of possible solutions that may fit in many different applications. Also it is important to investigate which are the more interesting architectures for FPGA arrays plus complements (Memory, I/O, programmable clocks, etc.) to extend its flexibility and reusability.

To find the right use of FPGA devices in a wide range of structures and situations, an UMTS physical layer implementation based on FPGAs is a good testbed that will also serve to analyse which are good solutions to get a reconfigurable fully digital terminal. Of course AD and DA technology has a lot to say about that, but it is clear that optimum implementation of digital processing blocks (whatever algorithm is used) is a cornerstone.

5. References

- [1] C. Taylor, "Using Software Radio in 3rd Generation Communications System", ACTS Mobile Communications Summit, Aalborg, Denmark, Oct. 1997.
- [2] M. Cummings, S. Haruyama. "FPGA in the Software Radio". IEEE Communications Magazine. February 1999.
- [3] F. Adachi, M. Sawahashi, H. Suda, "Wideband DS-SS-CDMA for Next-Generation Mobile Communications Systems", IEEE Communications Magazine, September 1998.
- [1] K. Chapman, P. Hardy, A. Miller, M. George, "CDMA Matched Filter Implementation in Virtex Devices", Xilinx XAPP212(v1.0), March 2000.
- [4] R. Esmailzadeh, M. Nakagawa, "Pre-RAKE Diversity Combination for Direct Sequence Spread Spectrum Mobile Communications Systems", *IEICE Transactions on Communications*, vol. E76-B, no. 8, pp. 1008-1015, August 1993.
- [5] X. Revés, A. Gelonch, F. Casadevall, "Reconfigurable Hardware Platform for Software Radio Applications (SHaRe) in Mobile Communications Environments." Proc. ACTS Mobile Communication Summit, Sorrento, Italy, June 1999.
- [6] XILINX Virtex™ 2.5 V Field Programmable Gate Arrays. 2000.
- [7] XILINX XC4000E XC4000X Series Field Programmable Gate Arrays. 1999.

Acknowledgement: This work has been supported by CYCIT (Spanish National Science Council) under grant TIC98-0684.