

Software Defined Radio

Enabling Technologies

Edited by

Walter Tuttlebee

Virtual Centre of Excellence in Mobile & Personal Communications (Mobile VCE)



JOHN WILEY & SONS, LTD

Wiley Series in Software Radio

Series Editor: Dr Walter Tuttlebee, Mobile VCE, UK

The Wiley Series in Software Radio aims to present an up-to-date and in-depth picture of the technologies, potential implementations and applications of software radio. Books in the series will reflect the strong and growing interest in this subject. The series is intended to appeal to a global industrial audience within the mobile and personal telecommunications industry, related industries such as broadcasting, satellite communications and wired telecommunications, researchers in academia and industry, and senior undergraduate and postgraduate students in computer science and electronic engineering.

Mitola: Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering, 0471384925, 568 Pages, October 2000

Mitola and Zvonar (Editors): Software Radio Technologies: Selected Readings: 0780360222, 496 Pages, May 2001

Tuttlebee: Software Defined Radio: Origins, Drivers and International Perspectives, 0470844647, £55, 350 pages

Tuttlebee: Software Defined Radio: Enabling Technologies, 0470843187, £55, 304 pages

Copyright © 2002 John Wiley & Sons Ltd, Baffins Lane, Chichester,
West Sussex PO19 1UD, England

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wileyeurope.com or www.wiley.com

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except under the terms of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London W1P 0LP, UK, without the permission in writing of the Publisher. Requests to the Publisher should be addressed to the Permissions Department, John Wiley & Sons Ltd, Baffins Lane, Chichester, West Sussex PO19 1UD, England, or emailed to permreq@wiley.co.uk, or faxed to (+44) 1243 770571.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the Publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Other Wiley Editorial Offices

John Wiley & Sons Inc., 605 Third Avenue, New York, NY 10158-0012, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Pappelallee 3, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 33 Park Road, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 22 Worcester Road, Etobicoke, Ontario, Canada M9W 1L1

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0-470-84318-7

Typeset in 10/12pt Times by Deerpark Publishing Services Ltd
Printed and bound in Great Britain by Biddles Ltd, Guildford and King's Lynn

This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

Contents

List of Contributors	xiii
Foreword - by Dr Joseph Mitola III	xvii
Abbreviations	xix
Biographies	xxvii
Introduction	xxxv
Part I: Perspective	1
1 Software Based Radio	3
<i>Stephen Blust – Cingular Wireless</i>	
1.1 A Multi-Dimensional Model Sets the Stage	3
1.2 What is Software Based Radio	5
1.2.1 <i>Software Defined Radio and Software Radio</i>	5
1.2.2 <i>Adaptive Intelligent Software Radio and Other Definitions</i>	8
1.2.3 <i>Functionality, Capability and SBR Evolution</i>	10
1.3 Architectural Perspectives for a Software Based Radio	11
1.3.1 <i>The Radio Implementer plane</i>	11
1.3.2 <i>The Network Operator plane</i>	12
1.4 Software Radio Concepts	13
1.5 Adoption Timeframes for Software Based Radio	15
1.6 Realization of Software Based Radio Requires New Technology	17
1.7 Power/Performance/Price Limitations of Handsets Dictates Inflexible Networks	17
1.8 Regulatory Concepts Facilitate SBR Introduction	18
1.9 Conclusions	20
Acknowledgements	21
References	21
Part II: Front End Technology	23
2 Radio Frequency Translation for Software Defined Radio	25
<i>Mark Beach, Paul Warr & John MacLeod - University of Bristol</i>	
2.1 Requirements and Specifications	26
2.1.1 <i>Transmitter Specifications</i>	26

1

Software Based Radio

Stephen M. Blust

Cingular Wireless

This chapter offers an introduction to software based radio (SBR), discusses some top-level global drivers for (SBR) designs, and postulates potential evolutionary implications for software radio in one key market space (commercial wireless). It provides a vision for the potential impact of software radio, supplying a context for the more detailed technology presented in subsequent chapters of this book.

SBR, also known as software defined radio (SDR) or just software radio (SR), is a technological innovation that is coming of age for wireless communications of many types. There is no one single agreed definition of SBR/SDR/SR terminology, in part because there are several different perspectives for the technology. In a broad sense, as will be described and illustrated later, the technology involves more than just 'radio' in the classic sense of the word 'radio'. It also involves more than just 'software' in the usual usage of that word.

In this chapter, suggested definitions are presented along with examples of the use of the terminology. Throughout this chapter, the term 'software based radio', as defined, is used as an overarching term that comprises SDR, SR, and adaptive-intelligent SR (AI-SR).

1.1 A Multidimensional Model Sets the Stage

The multidimensional aspects and different perspectives of SBR are illustrated in Figure 1.1. Starting at the bottom of the figure, the *radio implementers' plane* is what many people think about when using the terms 'SR' and 'SDR'. In this plane SDR is simply an implementation technique. At the radio level, SBR can be viewed simply as an efficient technique for the construction of wireless devices. The use of SBR technology is predicted to replace many of the traditional methods of implementing transmitters and receivers while offering a wide range of advantages including adaptability, reconfigurability, and multifunctionality encompassing modes of operation, radio frequency bands, air interfaces, and waveforms. This is the most fundamental perspective and the one that is encountered most frequently in the literature. This perspective is described in more detail in Section 1.3.1.

Ultimately, the full benefits of SBR can be achieved only by modifications to the network level of a wireless communications system. That is why the term 'SBR' refers to more than

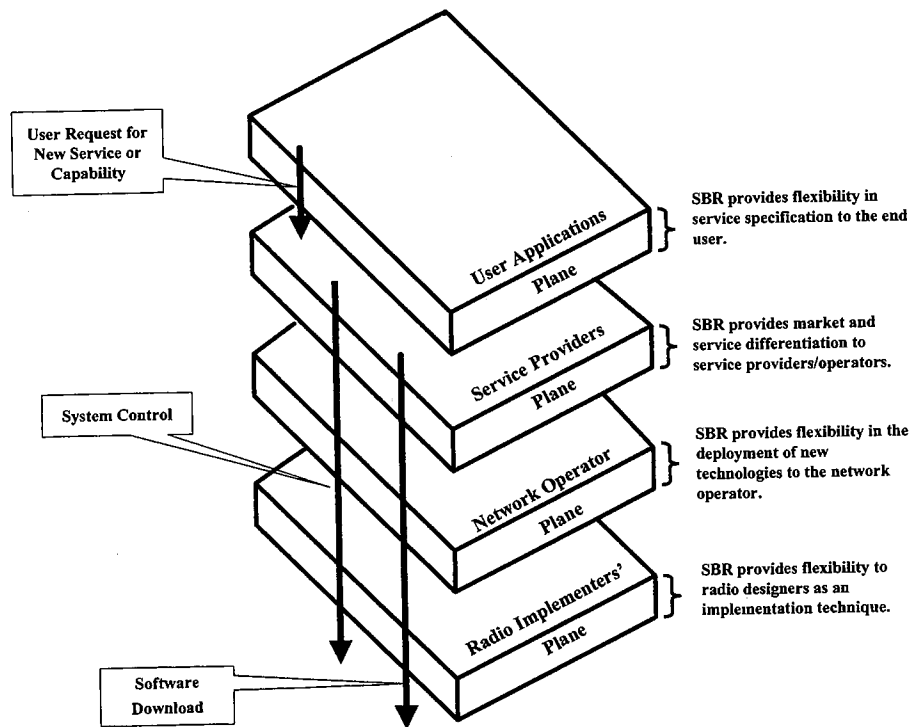


Figure 1.1 The Multidimensional perspectives of software based radio. Reproduced by permission of Cingular Wireless.

the receiver-transmitter pair of a wireless system. It involves the overall wireless system; hence the *network operator plane*. This is discussed in more detail in Section 1.3.2.

The *service provider plane* and the *user applications plane* round out the scope of SBR. Service providers see several advantages to the SBR technology, especially when viewed from a system level perspective. It allows the service provider a mechanism for service differentiation, and it provides a cost-effective means for easily upgrading the network infrastructure as technology evolves. In other words, upgrades to the infrastructure could be achieved through software downloads to the network and handset components instead of costly hardware swap outs of network components and handsets.

Finally, the *user's application* level depicted provides a user's perspective of the advantages of SBR. One could conceive that included in this plane would be the ability for an implementation of an SBR to have excess capacity in the hardware/software combination required to perform the radio functions at any given moment. This capability could be utilized to execute other user applications: for example a music player/codec for downloaded digital music. Thus the potential exists for reducing costs and increasing functionality by not adding into the device a separate dedicated processing system just for applications that might reside within it. The close internal coupling may be useful for users receiving services and capabilities on demand from a number of sources; these are shown by the vertical lines cutting across the planes in Figure 1.1.

The primary focus of the material presented in this chapter is the radio and network operator plane. The service provider and user planes are introduced in order to provide a

complete picture of the multidimensional aspects. These planes will be the subjects of future work as the lower planes are developed and implemented. In many respects the lower planes (radio and network) can be viewed as the critical enabling foundation for the potentials and benefits of SBR to be extended beyond the edges of the traditional 'wireless cloud'. Ultimately, the entire concept can be likened to the internet model but applied from the wireless device core operating system up through the applications themselves.

1.2 What is Software Based Radio?

The terms 'SDR', 'SR', and 'AI-SR', as defined, are utilized throughout this chapter to denote specific implementation stages of SBR. The term 'SBR' is introduced as a generic term for this broad collection of technology and concepts. Usage of this term means that the information will be generally applicable across all manifestations of the technology. SBR includes both software signals to process the radio signal and software control of the radio parameters as illustrated in Section 1.3.4.

1.2.1 *Software Defined Radio and Software Radio*

There are many understandings of what is considered a 'software based radio'. An SBR can be generically defined as a radio that uses software techniques on digitized radio signals. The fundamental intent is to shift from employing a traditional hardware-focused, application-specific approach to radio implementation to using a software application to perform the radio tasks on a computing platform.

For clarification, and to understand the evolution stages possible in SBR as a function of advances in the underlying core technologies, we have chosen two commonly accepted terms: 'SDR' and 'SR'. As technology progresses, an SDR can move to an almost total SR where the digitization is at (or near to) the antenna and all of the processing is performed by software residing in high-speed digital signal processing functionality. Hence, the SDR will occur in the near term, migrating to the SR in the longer term, subject to the progression of core technologies. The need for such progression will be a function of the application. For example, a base station application may require and/or be able by virtue of technology advances and design latitude to move to a software radio, but a handset or portable terminal, because of numerous constraints, may not need or be able to progress beyond an SDR.

1.2.1.1 Definition of the Software Defined Radio

An SDR is defined as a radio in which the receive digitization is performed at some stage downstream from the antenna, typically after wideband filtering, low noise amplification, and down conversion to a lower frequency in subsequent stages – with a reverse process occurring for the transmit digitization. Digital signal processing in flexible and reconfigurable functional blocks defines the characteristics of the radio.

1.2.1.2 Definition of the Software Radio

As technology progresses, an SDR can move to an almost total *SR*, where the digitization is at (or very near to) the antenna and all of the processing required for the radio is performed by software residing in high-speed digital signal processing elements.

A simple example of how these two definitions are related to each other is illustrated in Figures 1.2–1.5. It is evident from inspection of these, in light of the definitions chosen, that there is a key transition stage in the metamorphosis of SDR to SR. This metamorphosis is a function of core technology advances balanced against the full scope of design criteria and constraints applied to the wireless product. Core technology in this instance includes, as a minimum, analog to digital to analog conversion capabilities, digital signal processing advances, algorithms, memory advances, including similar attributes of the fundamental building blocks required for the digitization and manipulation of radio signals in the digital space and any requisite translation in frequency of the analog environment [1]. Design criteria and constraints include such factors as cost, complexity, performance, form factor, size, weight, power consumption, and so on. Additional analysis of these stages is provided in Section 1.3.1.

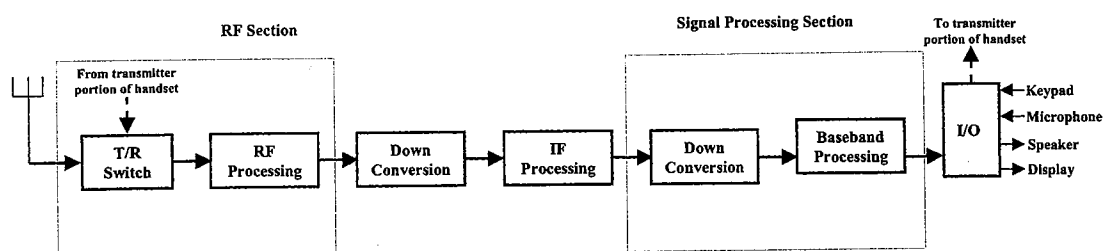


Figure 1.2 SDR evolution – stage 1: cellular /PCS generic single mode, single band handset. This figure is representative of ANY single mode (i.e. AMPS, TDMA, CDMA, GSM, PHS, etc.) and single frequency band (i.e. 850, 900, 1800, 1900, etc.) handset. This is considered to be the traditional design product implementation. Reproduced by permission of Cingular Wireless.

In the simplified example shown of a commercial wireless terminal device (i.e. a cellular or personal communications service (PCS) handset) there is a need to accommodate multiple radio technology interface types and frequency bands into the terminal. In a traditional implementation approach each unique radio interface or band combination would be constructed around a dedicated set of specific application or function integrated circuits. Essentially, the capabilities are hard coded and fixed at the time of design or manufacture. To increase the number of supported modes or bands, additional units of function are added into the terminal. These functional blocks would operate in a matrix arrangement of radio interfaces and frequency bands to provide a set of a priori defined capabilities.

Initial application of SBR results in the SDR as shown in Figure 1.3. At the onset the principal advantage is substitution of technology in the implementation. Subsequent implementations build on this base and engender wider ranging flexibility which can span the gamut from simple updates of radio functionality to complete over-the-air downloads of new radio interfaces. Sharing of processing capabilities by radio functions and applications riding the radio transport is a cost-effective leveraging of SBR radio capabilities that is a tractable step beyond the limitations inherent in the application-specific and unchangeable function blocks available in devices today.

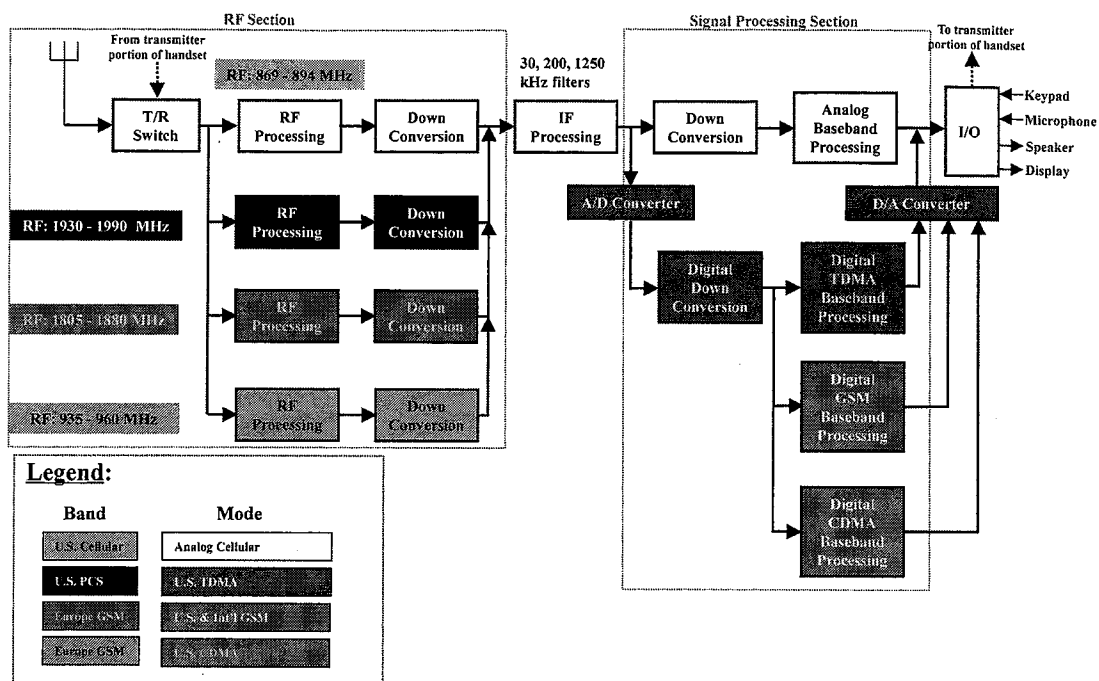


Figure 1.3 SDR Evolution – stage 2: quadruple-band (800, 900, 1800, and 1900 MHz), quadruple-mode (AMPS, TDMA, GSM, CDMA), traditional-design, multiband, multimode handset. Reproduced by permission of Cingular Wireless.

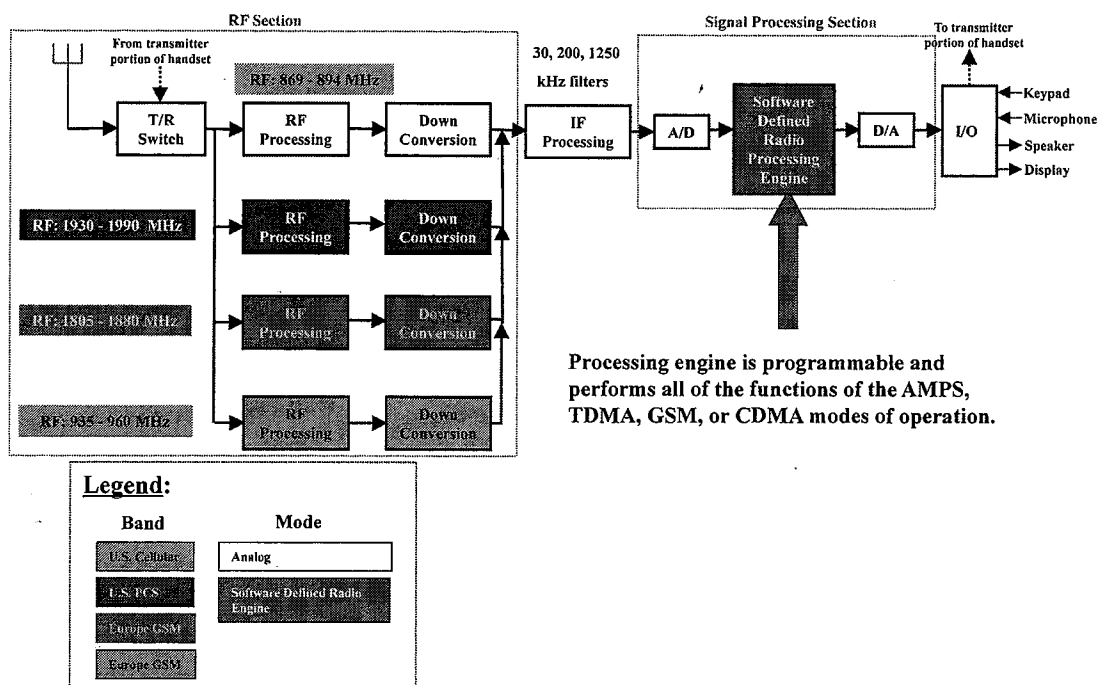


Figure 1.4 SDR Evolution – stage 3: A/D, D/A, and signal processing chips currently have the capacity to perform this IF and baseband processing. Reproduced by permission of Cingular Wireless.

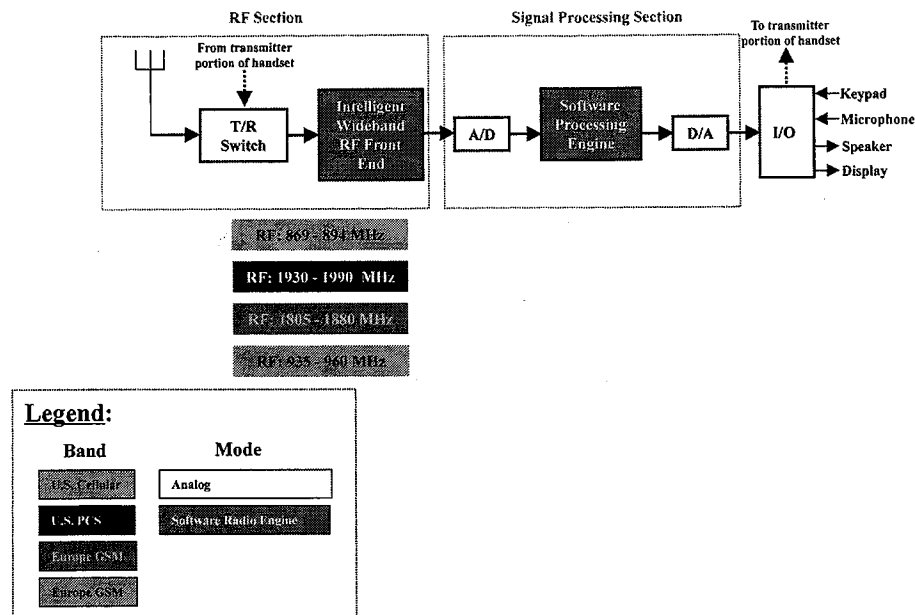


Figure 1.5 SDR Evolution – stage 4: future product as technology evolves in A/D capabilities, etc. Reproduced by permission of Cingular Wireless.

In the above discussion and throughout this chapter the terms ‘digital signal processors’ (DSPs) and the like are used in a broad sense; therefore DSPs include field programmable gate arrays (FPGAs), reconfigurable computing (RC), etc.

1.2.2 Adaptive Intelligent Software Radio and Other Definitions

For the sake of completeness and clarification, some addition terms and concepts should be considered in any discussion of SBR. They are reviewed in this section.

1.2.2.1 Definition of Adaptive Intelligent Software Radio

An *AI-SR* is one which is capable of adapting to its operational environment thereby achieving enhanced performance and/or spectral efficiency.

The basic concept underlying such terminology is the ability of the radio to adapt to its environment by automatically adapting (i.e. without human intervention) its operational mode to achieve enhanced performance and efficiency. This requires the use of artificial intelligence, significant computational power to process adaptive algorithms in real time, and real-time data from a variety of sources including the mobile network infrastructure, radio frequency (RF) bands available, air interface protocols, user needs, applications, minimum performance requirements (which might be subscriber dependent as well as application dependent), the propagation environment, and the capabilities of the SDR platform. Thus, the *AI-SR* radio is an extension of the SDR and SR concepts as defined above. As a simple example of this extension, the radio might adapt in real time to the propagation environment by using a more robust waveform developed dynamically as the propagation environment rapidly deteriorates. Although at first glance this might appear relatively easy to implement,

in reality it is very complex because of the need to interact with the mobile network infrastructure and the need for the radio to process all of the inputs described above.

1.2.2.2 Definition of Digital Radio, Multiband, and Multimode

Digital radio is a radio in which the information is digitized at some point between the antenna and the input/output devices. Digital radio does *not* necessarily mean that the radio is an SDR. A radio may be digital, but if the signal processing that takes place after the A/D conversion is performed by special purpose, application-specific integrated circuits (ASICs) it is not an SDR. *Multiband* is the capability of handsets or base stations to operate in multiple frequency bands of the spectrum. *Multimode* refers to the capability of a handset or base station to operate in multiple modes (e.g. multiple air interface standards, multiple modulation techniques, or multiple access methods). Multiband/multimode capabilities may be implemented using a variety of hardware and/or software techniques, including SDR.

It should be recognized that SBR is applicable to many differing marketplaces for wireless. A common distinction that has been made is to consider three major application universes:

- commercial wireless (e.g. cellular, personal communications services (PCS), land mobile, etc.)
- civil government (e.g. public safety, local, state, and national communications, etc.)
- military

Each of these major markets has a differing set of criteria (e.g. cost, weight, size, performance, features, etc.) that directly impacts the application and definition of SBR as applied to each of these domains. This must be taken into account when understanding the evolution and influence of SBR. There is, however, significant overlap in the applicability of SBR across these market domains and this is a strong driver for the development, and adoption, of SBR.

In this chapter, the focus is on SBR as principally applied to the commercial wireless domain. Increasingly, reconfigurability, flexibility, multiband, and multimode characteristics are required in all types of radio based communications systems including commercial wireless services, military communications, and civil government services. Many of these systems are evolving into their next generation counterparts. As a result, these systems face the problems associated with a deployed embedded base and the need to preserve continuity across both the old and new systems, often over a transition interval that may span many years. There is an increased expectation of usability by those who deploy systems and by the end users of such systems. Manufacturers of systems face related concerns in providing product for these marketplaces. Consequently, broad interoperability among diverse systems using many frequency bands, often on a global basis, forms a baseline requirement now and into the future. In the longer term, the ability to evolve an embedded base of deployed wireless systems, both infrastructure and terminal devices, to accommodate new capabilities on a dynamic basis will become an additional key design factor. Solutions to dynamic reconfigurability include both a versatile hardware and software environment and the ability to provide updated, enhanced, or replacement capabilities via a download mechanism.

1.2.3 Functionality, Capability and SBR Evolution

In light of the above basic definitions and concepts, we emphasize that there are two distinctly different aspects of software functionality that may be incorporated into a radio:

1. software processing of the information signal;
2. software control that provides intelligent adaptation of the radio parameters to achieve higher performance (e.g. lower bit error ratios for data transmission) and/or greater spectral efficiency (e.g. higher bits per second per Hertz) as the radio adapts automatically to its environment.

These two distinctions of software functionality in current and future radio technology are directly related to two of the fundamental questions in wireless:

1. the capability of SBR technology to address interoperability issues resulting from the plethora of wireless communications systems;
2. the ability of SBR technology to achieve greater spectral efficiency and spectrum utilization (including dynamic spectrum sharing and interruptible spectrum).

On one end of the scale, the industry is generally enthusiastic about the ability of SBR to address critical interoperability issues for a wide variety of applications and to provide multimode, multiband capabilities for commercial wireless systems in the near term. On the other hand, industry does not generally support the view that SBR, using adaptive intelligent radio concepts, can solve spectrum efficiency and spectrum management issues for 10 years or more. The adaptive intelligent aspects of SR which could ultimately lead to adaptive, dynamic spectrum sharing can be viewed as an even longer-term evolution of radio technology. This illustrates the divergence of views on how SBR applies in the marketplace as it evolves from niche solutions to a future where it is pervasive throughout the wireless system, with the overall system capabilities offering benefits in all the planes of the multi-dimensional model previously introduced.

Figure 1.6 shows the evolution of technology from SDR to software radio to AI-SR. The triggering mechanisms for the evolution of SDR to SR are advances in signal processing technology including A/D and D/A converters, faster signal processors, memory chips, etc. Advances in these core technologies are necessary to move the digitization of the radio signal from the base band to the intermediate frequency (IF) stage to the radio frequency (RF) stage (near to the antenna in the ideal SR radio). Advances in intelligent network algorithms are necessary to trigger the further evolution of SR to the AI-SR capable of enhanced spectral efficiency through adaptive spectrum sharing and spectrum management. Note that the evolution for commercial wireless handsets is on a different timescale than that of commercial wireless base stations – the latter will evolve more quickly than the former. The fullest benefits of the technology at all levels of consideration (cf. Figure 1.1) require fully deployed software based handsets and software based base stations in order to begin to achieve alternative approaches in wireless communications. In addition to this full deployment, complex adaptive algorithms addressed at a system level must be developed.

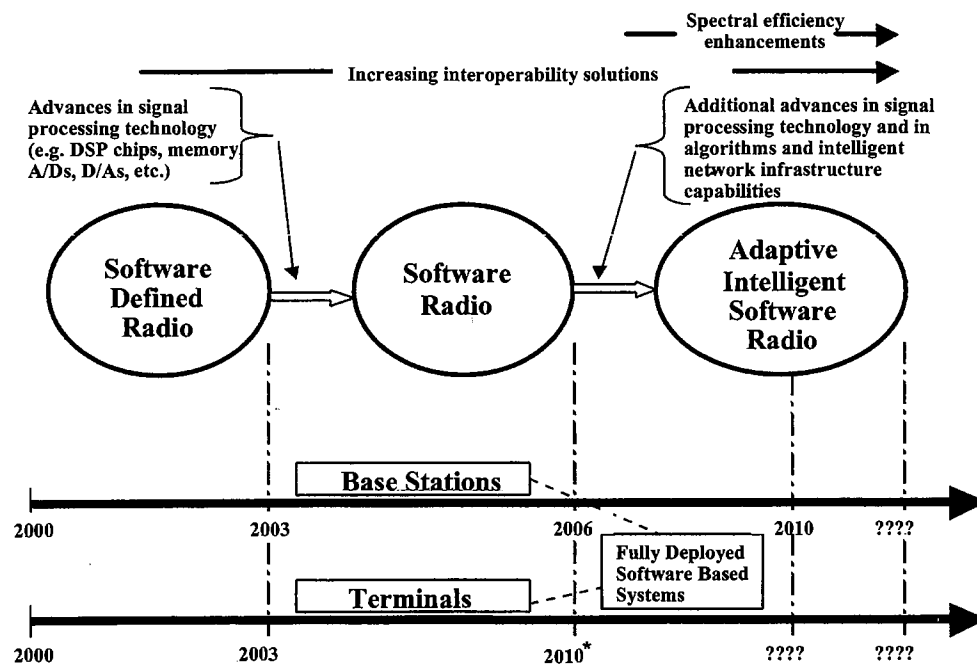


Figure 1.6 SBR – Evolution from SDR to SR to AI-SR. Note differing time scales; evolution is presented relative to stages portrayed in circles. Other wireless services such as military or civil government systems may have differing timelines. *Subject to progress of core technologies (e.g. A/D converter, power consumption, etc.) Reproduced by permission of Cingular Wireless.

1.3 Architectural Perspectives for a Software Based Radio

1.3.1 The Radio Implementer Plane

In considering the architecture [2] of the radio device, whether terminal or base station, it can be simplistically divided into two major functional areas:

- radio front end – the radio frequency aspects, for both receive and transmit
- radio back end – the signal processing functionality

Traditionally, this subdivision leads to an artificial assigning of hardware as the dominant element of the front end, and hardware coupled with software as the dominant element of the back end. These assignments to hardware and software will certainly change over time, with software becoming the dominant driver and the hardware moving to a supporting role, particularly as technology advances [3]. The window of opportunity for these changes overlaps the wireless evolution window from current generations through to full maturity of the next generations.

As has been pointed out in other analyses, a large portion of the complexity of a commercial wireless device today is in the RF section needed to span multiple frequency bands. The multimode functionality (i.e. differing radio air interface technologies) can be effectively realized in software in the processing section as a relatively small portion of the overall complexity, particularly as a wireless device moves through subsequent product generations. It should be understood that it is fully recognized that the front end and back end considera-

tions are very complex in themselves and that the discussion is one of relative complexity in the wireless device as additional frequency bands and modes are accommodated. This becomes even more of an issue when current and future generations must be accommodated in the same wireless device.

Figures 1.2–1.5, taken from the commercial wireless world, are instructive of the functional subdivision into the radio front end and back end concept and also serve as an evolutionary pointer towards the ultimate objective encompassed by SR. This example addresses the product evolution from a traditional design, single band and single mode radio interface perspective, moving to a multicapability, multimode, and multiband digital radio global phone.

- Stage 1 is the baseline single mode, single band case (cf. Figure 1.2).
- Stage 2 is the second generation world phone, which cannot currently meet all requisite objectives of the marketplace when implemented in conventional solutions (cf. Figure 1.3).
- Stage 3 is the first evolutionary stage that retains a conventional approach in the currently more complex portion in the RF front end and employs the SDR solution in the back end, where it makes sense in the near term (cf. Figure 1.4).
- Stage 4 represents what might be considered the ultimate evolution, where there is consolidation of functionality into software relying on technological advances (cf. Figure 1.5).

While all global frequency bands for commercial wireless (in particular the Asia region) have not been shown on the figures, principally due to space limitations, it is evident that these and other frequency bands would and can be accommodated using an SDR design approach. Included is an expectation that the ITU-R IMT-2000 third generation, 3G, radio technologies operating in the IMT-2000 recognized frequency bands (and in existing first and second generation technology frequency bands) will require SBR concepts.

It must be noted that similar examples can be constructed for flexible base stations. These base stations have the potential to achieve Stage 4 sooner than a terminal device because of differing design constraints, i.e. relaxed form factor, ability to support higher power consumption, need to accommodate multiple simultaneous carriers/modes, etc.

1.3.2 The Network Operator Plane

There are two primary roles for software based radios – first, extremely flexible base stations as well as mobile handsets/terminals and second, the creation of adaptive networks. Early discussion focused very much on the former, whereas perhaps the latter has potentially the most profound implications.

Wireless networks are static at the present time. Voice and limited amounts of data can be transmitted and received, and devices at either end are generally fixed in their configurations and capabilities. SBR is making its entrance into the wireless domain to reshape static wireless networks into dynamically adaptive systems. Hence, the entire nature of the network can change rather than being fixed and static.

In the next few years, wireless networks will be a mix of 2G, 2.5G, and 3G, and there will be a proliferation of new variations of the existing standards, and entirely new standards, some of which are currently under development. Each standard essentially requires a new system to be put into place before consumers can use it, and before wireless carriers can begin to make money.

An example of the constant evolution of existing standards can best be seen by examining one of the currently established standards: GSM. In the next few years, about once every 12–15 months a new variation, or what can best be described as a ‘network overlay’, is scheduled to be added to the GSM standard. For example, some of the variations that are occurring for GSM are the new adaptive multirate codec (AMR), 14.4 kb/s data services, and general packet radio service (GPRS) among others. With the GSM evolution to 3G, using either enhanced data rates optimized modulation (EDGE) or wide-band code division multiple access (WB-CDMA), the cycle of deployment and evolution repeats again. Each of these variations requires anything from selected upgrades within the network infrastructure, to an overlay-type build out addition of new technology across the wireless system, to wholesale removal and replacement of essentially the entire system. The situation is similar for technology evolutions such as those associated with code division multiple access (CDMA) (IS-95), time division multiple access (TDMA) (IS-136), and even PHS.

The injection of SBR base stations in conjunction with SBR handsets radically changes the picture. Since SBR base stations and handsets can both potentially evolve along with the currently existing standards, as well as adapting to new standards, a single ‘adaptive’ network can meet these changing requirements. The creation of multiple overlay networks for each new standard may no longer be required, or at the very least reduced.

1.4 Software Radio Concepts

Figures 1.7 and 1.8 provide further illustration of the conceptual definitions. The figures, because of their generality, are equally applicable to either a commercial wireless handset or base station architecture. In the example in Figure 1.7, the analog-to-digital (A/D) converter is placed after the intermediate frequency (IF) processing. As indicated in the figure, the radio baseband processing is under software control and a man-machine interface permits some manual input from the user. The architecture in this example is considered to be a SDR since some, but not all, of the signal processing is accomplished in software. It is of course possible for the digital baseband processing to be implemented in application-specific integrated circuits (ASICs), in which case the radio would be considered to be a digital radio but not an SDR.

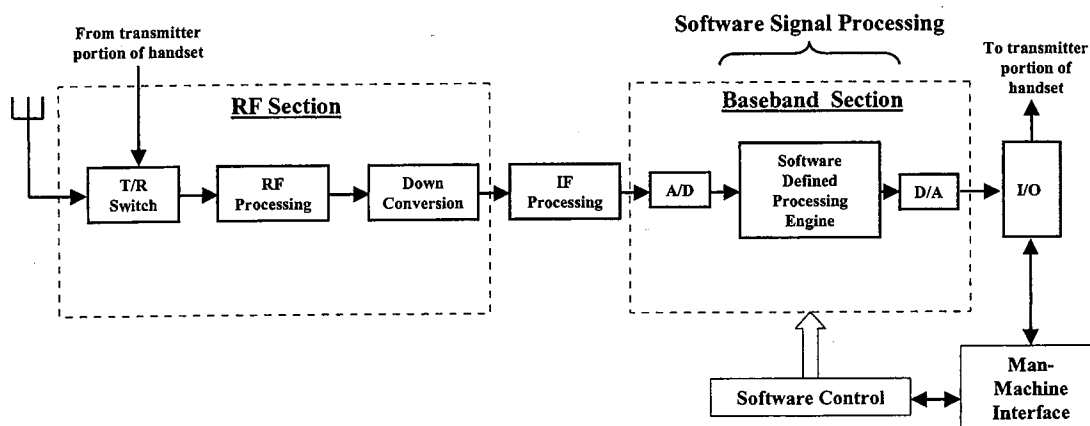


Figure 1.7 Conceptual definition of the software defined radio (SDR). This functional block diagram has been simplified for illustrative purposes. Reproduced by permission of Cingular Wireless.

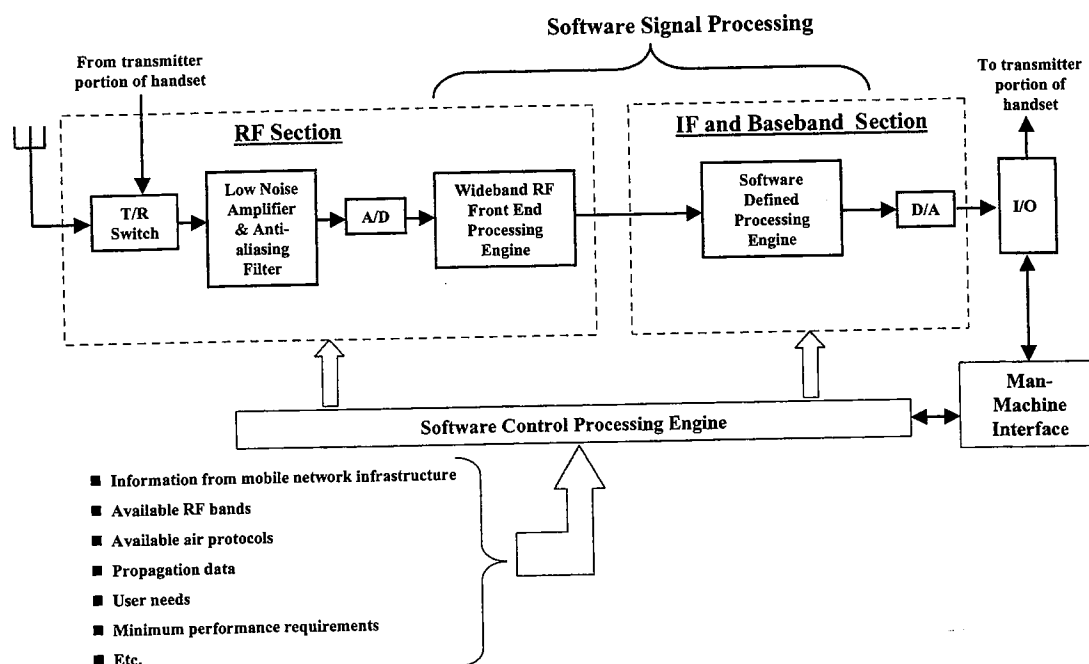


Figure 1.8 Conceptual definition of the software radio (SR) and the adaptive intelligent software radio (AI-SR). This functional block diagram has been simplified for illustrative purposes. Reproduced by permission of Cingular Wireless.

If the A/D converter were to be moved closer to the antenna, the radio moves closer to the ideal, namely the SR as defined above. This is portrayed in Figure 1.8 which illustrates two concepts:

1. the software radio concept in which the digitization is nearly at the antenna;
2. the AI-SR concept.

In Figure 1.8, the A/D converter is immediately after the low noise amplifier and anti-aliasing filter; i.e. the digitization is at RF. The other RF and IF processing is performed by the wideband RF front end processing engine. The intermediate case, which is not shown, would be between the architectures shown in Figures 1.7 and 1.8 in which there is direct digital conversion to baseband from RF, thereby eliminating analog IF processing.

It can be expected that for the foreseeable future there will be a need for an analog low noise amplifier in the RF front end of the receiver and an analog final power amplifier in the RF portion of the transmitter – therefore, the ideal SR of an A/D converter right at the antenna appears to be unattainable in practice, but is certainly seen as an ultimate goal of technology developers.

Figure 1.8 also illustrates the concept of the AI-SR in which the radio has the capability of adapting to the operational environment. The RF front processing engine is under the control of a powerful software control processing engine. This software control processing engine implements the artificial intelligence and the processing algorithms that give the SR a truly adaptive capability. In the commercial wireless world this is the type of processing that would be needed to achieve adaptive spectrum sharing, interruptible spectrum, spectrum-on-demand, or adaptive spectrum management. While this capability is a

highly desirable goal, the research into the component parts needed to implement this capability is in the embryo stage. Note the large number of inputs required, including information from the mobile network infrastructure and other sources outside the wireless system.

1.5 Adoption Timeframes for Software Based Radio

It must be pointed out that given the acknowledgement that second generation (2G) capability will need to be accommodated in third generation (3G) IMT-2000 terminals and also in infrastructure in transition deployments, the impact of SDR as a preferred and dominant design cannot be overlooked. Additional capability is also important when one considers that in addition to the traditional 2G and 3G capabilities there may be significant market drivers for the inclusion of Bluetooth, 802.11a, 802.11b, or Hiperlan networking and similar capabilities in the same devices. The multimode multiband driver of the second generation, which will continue for some time, could be an entry vector for SDR and it is envisaged that such a driver will definitely extend into second generation/third generation infrastructure and terminals. A perspective of the opportunity space for SDR for commercial wireless is presented in Figure 1.9. It is anticipated that commercial wireless will be a principal global driver for SDR in both the terminal and base station [4]. Similar future generation drivers apply to other radio services such as civil government (e.g. public safety, aviation, emergency/disaster communications), and military communications systems [5].

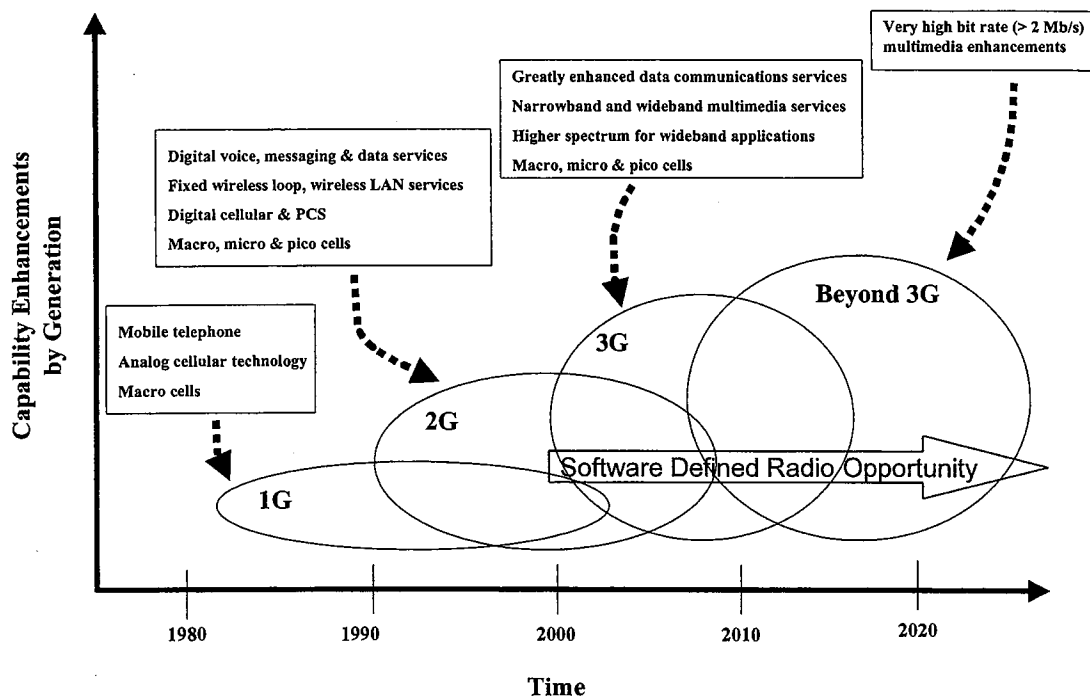


Figure 1.9 The market opportunity for software defined radio; generations of terrestrial commercial wireless systems and the software defined radio opportunity window. Reproduced by permission of Cingular Wireless.

The debate of cost, complexity, performance, size, weight, power consumption, and the like between the traditional implementations and the SDR implementations in the near term and that of SR in the longer term is the subject of ongoing analysis. Such a transition is significantly influenced by the cost/capability advances that are continuously occurring in the underlying core technologies.

Current research indicates that SDR is considered to be a displacing technology in the early implementations in commercial wireless and in the civil government areas of application. That is, the initial applications for SDR must be those where a small premium in cost is offset by the advantages that SDR accrues. As SDR becomes more widespread in its usage, it will become the dominant technology and the preferred design for most applications. This will be due in part to an increased shipped volume of product requiring SDR, the mastery of the learning curve for SDR development, related technology advances, and an increasing general marketplace acceptance of SDR. A projection of commercial wireless adoption of SDR in terminals and base stations is presented in Figure 1.10.

Base Stations:

1999 - 2003	very limited use by infrastructure manufacturers
2004...	increased use due to significant driver of third generation wireless
2005 ...	widespread adoption by most manufacturers as core platform

Terminals:

1999 - 2001	almost no usage
2002-2004	initial usage to meet multimode, multiband, in the 1G & 2G market ("high end" products and "roaming" segment); some niche 3G designs
2004-2005...	increased usage to add 3G capabilities ("information centric" product, embedded application segment for computers, autos...)
2006...	widespread adoption and movement to SDR as baseline design

Figure 1.10 Timeline for Commercial Adoption of SDR. Reproduced by permission of Cingular Wireless.

In a military application, the multifaceted needs of consolidation of radio procurement programs and the desire to achieve a common communication platform across services, branches or for tactical missions sets a different objective. Because SDR can be driven by governmental mandates and procurement program specifications and requirements, it will be chosen as the dominant technology and, subject to funding levels, a rapid introduction of both SDR and SR into this marketplace is anticipated.

In short, the commercial wireless and civil government sectors are market driven, while the military marketplace is likely to have a mandated capability to capitalize on the flexibility and marketplace economics of the commercial environment.

1.6 Realization of Software Based Radio Requires New Technology

Clearly the existence of SBR has overwhelming advantages, some of which are discussed in detail below – advantages that not only allow flexible devices but also imply profound changes to the nature of wireless networks themselves.

The primary question which arises is whether the implementation of SDRs is achievable using today's technology. There currently exist limited first generation SDRs in base stations. These base stations allow the selection of either AMPS or TDMA or another mode on a per channel basis via software. In design today are third generation compliant base stations which will allow the selection via software of companion second generation (2G) and third generation (3G) technologies. Current design technology, in the form of arrays of digital signal processors (DSPs) along with arrays of custom ASICs, can meet the flexibility requirements in the base station arena. However, as with any initial implementations, base stations can benefit from a more accelerated and enriched adoption of SBR.

As we will see below, the crux of the matter is the terminal or themobile handset – the one device where the functional requirements demanded by consumers, the power dissipation requirements, the low cost requirements, and size requirements combine to require new technology. The use of a new technology approach could result in the development of mobile SBR handsets within the next few years rather than the decade long wait that is required if conventional technology must wait for Moore's law to provide the required multiple billions of operations per second capability and flexibility simultaneously. A range of such potentially applicable technologies is described elsewhere in this volume.

It is interesting to note that, while the principal goals and constraints between base stations and terminals may be quite different, there is considerable overlap of the core technology developments between the two applications.

Other developments are underway that may address the complexity of the radio front end. One promising application of technology, not traditionally associated with wireless, is micro-electro-mechanical-systems (MEMS) which is being investigated in this context.

1.7 Power/Performance/Price Limitations of Handsets Dictates Inflexible Networks

So in a cellular network, where is the SBR evolution bottleneck? What is the element that precludes the introduction of SBR? The bottleneck is at the terminals, where the power/performance/price limitations force trade-offs that a base station does not have. In fact the power/performance limitations of handsets currently dictate inflexible networks. For example, the processing requirement in a modern 3G terminal is conservatively estimated to exceed 6–13 billion operations per second [6]. Approaching this in a traditional manner with an ultimately flexible view would suggest placing all of this processing power into a DSP. Please note that the use of the term 'DSP' in this section is intended to be the traditional definition of DSP (digital signal processor integrated circuits), not a far ranging concept of generalized signal processing as utilized previously in the chapter. Unfortunately, modern day DSPs, which are rated at 6 billion operations per second, also have a power dissipation that is measured in terms of many multiples of watts – with the undesirable end result of a battery life that is measured in minutes.

So what happens? To meet the battery performance requirements, the engineering teams then start placing functions which require significantly high operations per second of performance into silicon pieces such as ASICs. The trade-off here is twofold.

First, the number of 'functions' or 'features' required all consume silicon area, which equates to cost. The number of 'features' rapidly increases the size of the silicon die to the point that the price begins to exceed the budgets set. A good rule of thumb for the cellular handset area is that the total area of digital silicon should not exceed 100 mm² to produce a competitively priced handset. Unfortunately, as in most design situations, the number of features wanted exceeds the silicon area (cost effectively) available.

Second, any processing function which is placed in silicon is totally inflexible to change. Any changes to standards, bug fixes, better algorithms, all equate to a new silicon design. For example, in the arena of speech codecs, better performing algorithms, in terms of compression and/or voice quality, are currently being developed almost on an annual basis; e.g. for GSM, it went from RPE-LPC to HR to EFR to AMR. A 9-month silicon design cycle cannot hope to keep pace.

The handset, and not the base station, is the one component (at least at the fundamental radio air interface level) that is most resistant to change in existing networks. The base station component can be upgraded at one time at some cost to the network operator on the operator's desired schedule. However, handsets are owned by individuals and thus have a life span of their own. Even as the average life span of handsets approaches 12 months, it must be noted that many consumers keep handsets for many years.

Current ASIC/DSP technology simply does not have the horsepower and flexibility to implement several multiple standards at a time [7]. Current estimates on the requirements for a 3G system including applications are anywhere from 6 billion operations per second to 13 billion operations per second. A fully software implementation on today's latest DSPs would require anywhere from 20 to 80 low power DSPs on a die to provide this level of processing power. Thus the designer of a cellular system is faced with the prospect of using a few DSPs to provide some level of quick time to market while augmenting the DSPs with hardware accelerators to provide the raw horsepower required and paying the price in terms of silicon area and accompanying cost. The limitations and trade-offs of today's technologies are discussed in greater detail in later chapters of this book,¹ as well as some of the innovative technologies which would appear to offer significant breakthroughs in overcoming what have been seen as longstanding traditional constraints.

1.8 Regulatory Concepts Facilitate SBR Introduction

While a full treatment of the regulatory aspects of SBR is beyond the scope of this treatise,² it is appropriate to introduce some fundamental considerations that regulators globally should be considering as they develop facilitating regulations for SBR.

¹ A comprehensive introduction to the trade-offs of ASICs, DSPs, and FPGAs, including design methodologies, may be found in Chapter 7.

² Regulatory perspectives from North American and European regulators may be found in Chapters 10 and 11, respectively, in the companion volume to this, *Software Defined Radio: Origins, Drivers & International Perspectives*, W. Tuttlebee (Ed.), John Wiley & Sons, Chichester, 2002.

There are four broad areas of inquiry that comprise common areas of foundation information about SBR that are applicable on a global basis of regulatory review:

- state of the technology
- interoperability between radio services
- spectrum efficiency and sharing
- equipment approval process

Within these areas some initial conclusions can be drawn, given the current state of SBR advancement globally.

With regard to the 'state of the technology', it is evident that SBR (at least the early stages of SDR) is sufficiently developed to move forward with the introduction of regulation, or the modification or removal of regulations as the case may be, to allow SBR products to enter the marketplace.

In the area of 'interoperability between radio services', it is understood that SBR is an important tool but that there are considerations beyond just technology development. The practical aspects of the wide scale use of SBR as an interoperability mechanism must recognize that there must be a sufficient 'critical mass' of SDR-capable equipment for all impacted services being deployed in the marketplace over time. There is a learning curve and a transition period that must be acknowledged.

When one considers 'spectrum efficiency and sharing', it is natural to think that SBR as a new radio implementation technology could help radio spectrum users and regulators alike to get the most out of any given set of spectrum allocations. The answer, unfortunately, is quite similar to that for interoperability. Critical deployed mass. These advantages will accrue in time, but once again there must be significant introduction (perhaps even approaching ubiquity) of SBR-capable radio equipment and attendant network and system control capabilities, both at the user terminal and also at the fixed infrastructure level. Additionally, when looking towards a panacea solution, there is the reality about the finite set of flexibility that realistic marketable products can have, especially if they are positioned in a consumer-oriented wireless environment like commercial wireless and public safety. Military application, of course, is a very different case.

Finally, there is the important area of 'equipment approval process'. In this area we have the items that are of the most interest to the regulators. In addressing concerns about frequencies of allowable use, power output authorized, modulations to be used, interference potentials, and the like, the regulators working with industry are able to craft appropriate rules and procedures to allow these characteristics to be managed in a flexible product like SBR – even accommodating the fact that SBR devices have a designed-in flexibility that can accommodate change after manufacture and initial certification [8]. This is truly a new (and welcome) approach on the part of regulatory agencies that fosters a recognition and full use of new technology in what was once a hardware-only dominated implementation space. In SBR, because of the widespread and pervasive use of wireless, we see the potential, as never before, to develop a regulatory perspective that can encompass globally coordinated concepts.

Figure 1.11 takes the multidimension SBR model introduced in this chapter and maps a regulatory perspective onto it in a first attempt at understanding a system view of SBR and regulatory interaction. Similarly, Figure 1.12 provides that same perspective applied to the SBR functional model as initially shown in Figure 1.8.

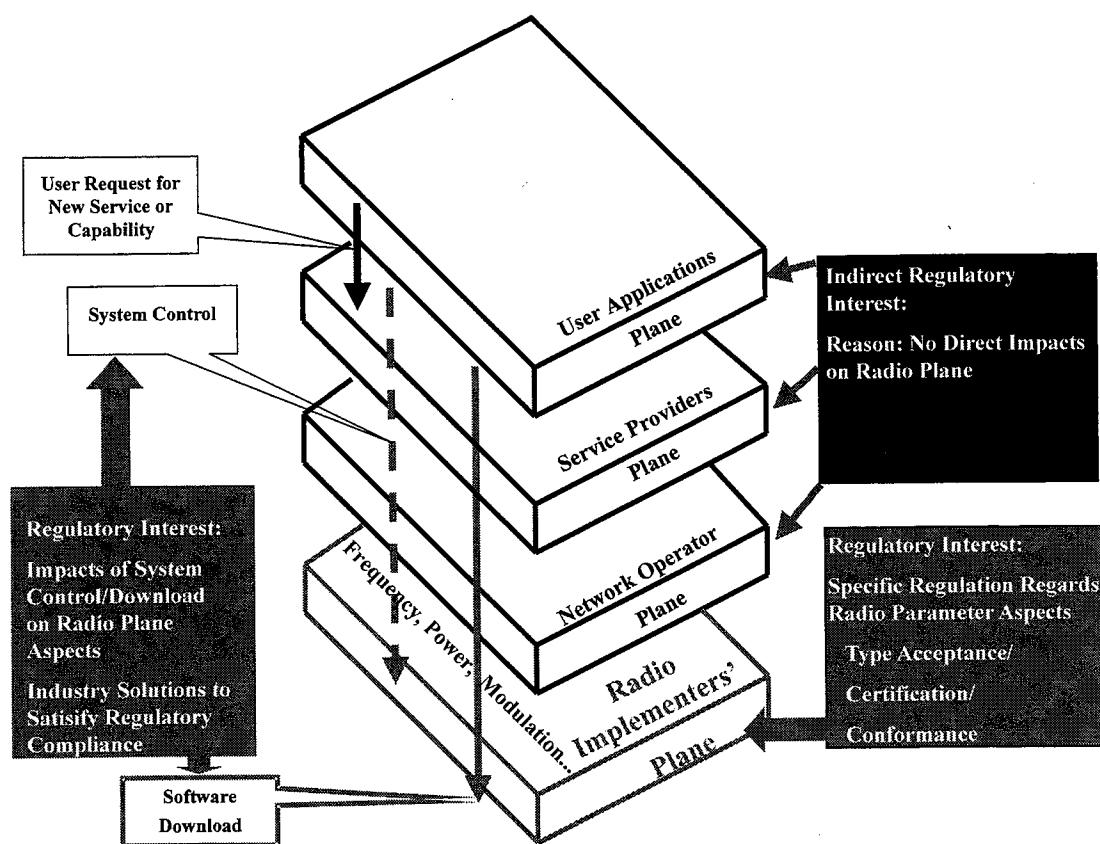


Figure 1.11 Regulatory issues of software defined radio. In this perspective the Radio Implementers' Plane is intended to be more of a physical representation viewpoint that encompasses either a radio base station or a terminal (handset) or both. Reproduced by permission of Cingular Wireless.

It is evident, as SBR moves through its stage from SDR ultimately to AI-SR, that the flexibility and capabilities will grow. As these capabilities grow, so too will the deployed base of products, both those that comprise the infrastructure and those that are considered terminals, handsets, or mobile devices. It is necessary and prudent that industry and regulators continue to work together to understand how the fullest potential of SBR can be brought to the marketplace to facilitate both the goals of industry and those of regulators in light of these increasing capabilities.

1.9 Conclusions

Industry understanding of the potentials of software radio has advanced significantly since the seminal work of the early and mid-1990s. From an initial recognition of the potential of the technology to allow an easier introduction and support of multiple air interface standards, the wider commercial implications of adaptive, evolvable networks have begun to be appreciated. The key constraint to realizing such a vision is the technology requirements of the handsets, illustrated by the trade-offs between traditional DSP vs. ASIC technologies and their capabilities. With such potential benefits, significant investment is now occurring to

Regulatory Scope within the Software Based Radio "Box"

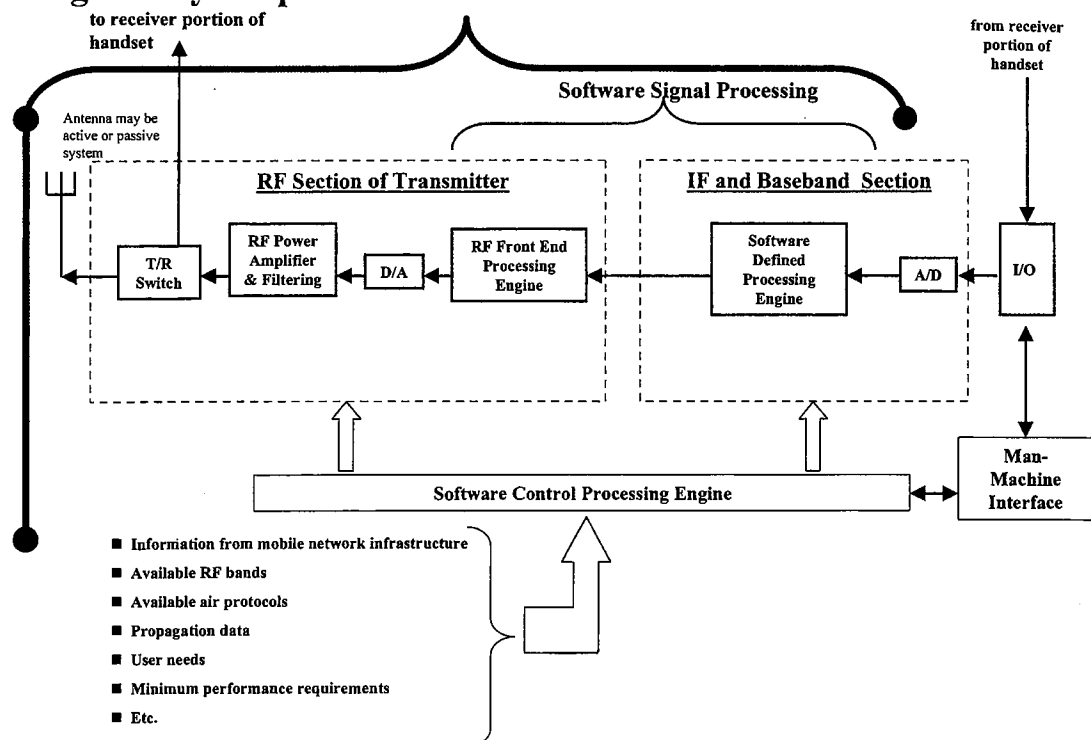


Figure 1.12 Regulatory implications for software based radio equipment. This functional block diagram has been simplified for illustrative purposes. Reproduced by permission of Cingular Wireless.

make software handsets a commercial proposition. The regulators around the world are positively addressing how they might adapt regulations to facilitate the introduction of SBR products. As all the factors come together, within the next decade we can expect to see the advent of adaptive SBR infrastructures.

Acknowledgements

I am indebted to Mr. James Hoffmeyer for his contributions to this work. James A. Hoffmeyer is president of Western Telecom Consulting, Inc. He is under contract to Cingular Wireless and is involved in activities within ITU-R and ITU-T for IMT-2000. He is the rapporteur appointed by WP 8F to liaise between WP 8F and the ITU-T on IMT 2000. Mr. Hoffmeyer is also working on future vision issues associated with IMT-2000 and is also active in SR developments.

References

- [1] Mitola III, J., 'Technical challenges in the globalization of software radio', *IEEE Communications Magazine*, February 1999, pp. 84-89.

- [2] Software Defined Radio Forum, Version 2.1 of the Technical Report on 'Architecture and elements of software defined radio systems', February 2000, see <http://www.sdrforum.org>
- [3] Hentschel, T. et al., 'The digital front-end of software radio terminals', *IEEE Personal Communications*, August 1999, pp. 40–46; see also Chapter 6 in this volume.
- [4] Schneiderman, R., 'Study views software radios as future design element', *Wireless Systems Design*, February 1999, pp. 8–11.
- [5] Software Defined Radio Forum, 'SDR Market Demand Forecast Series: Software Defined Radio: A Window of Opportunity in Wireless Communications', February 1999; see also Ralston, J.D., 'A market perspective: software defines radio as the dominant design,' in Tuttlebee, W. (Ed.), *Software Defined Radio: Origins, Drivers & International Perspectives*, John Wiley & Sons, Chichester, 2002.
- [6] Blust, S. et al., 'Software defined radio moves into base-station designs', *Wireless Systems Design*, November, 1999.
- [7] Salkintzis, A.K. et al., 'ADC and DSP challenges in the development of software radio base stations', *IEEE Personal Communications*, August, 1999, pp. 47–55.
- [8] United States of America, Federal Communications Commission, First Report and Order 'Authorization and Use of Software Defined Radios', ET Docket No. 00-47, FCC 01-264, Adopted September 13, 2001. http://www.fcc.gov/edocs_public/silverstream/pages/edocs.html (search for 01-264 in the FCC number field).

Part II

Front End Technology

Front End design – including RF Architecture, Data Conversion and Digital Front Ends – has emerged as a key issue as SDR techniques are finding themselves increasingly embodied by stealth into today's new products.

The radical solution – 'Pure' Software Radio, with A/D conversion at the antenna – is not yet feasible at GHz carrier frequencies. However, recent technology advances suggest it may be nearer than had been thought.

Bluetooth

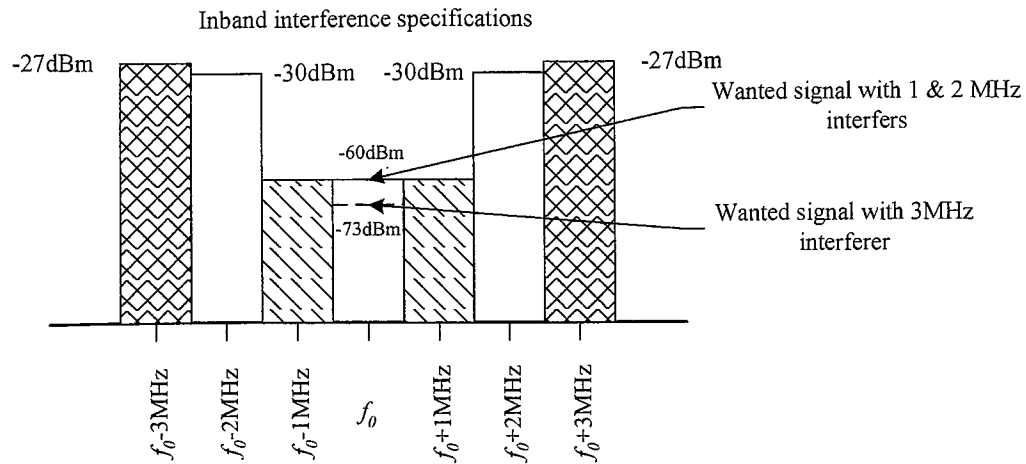


Figure 2.47 Bluetooth receiver blocker specification

HIPERLAN/2

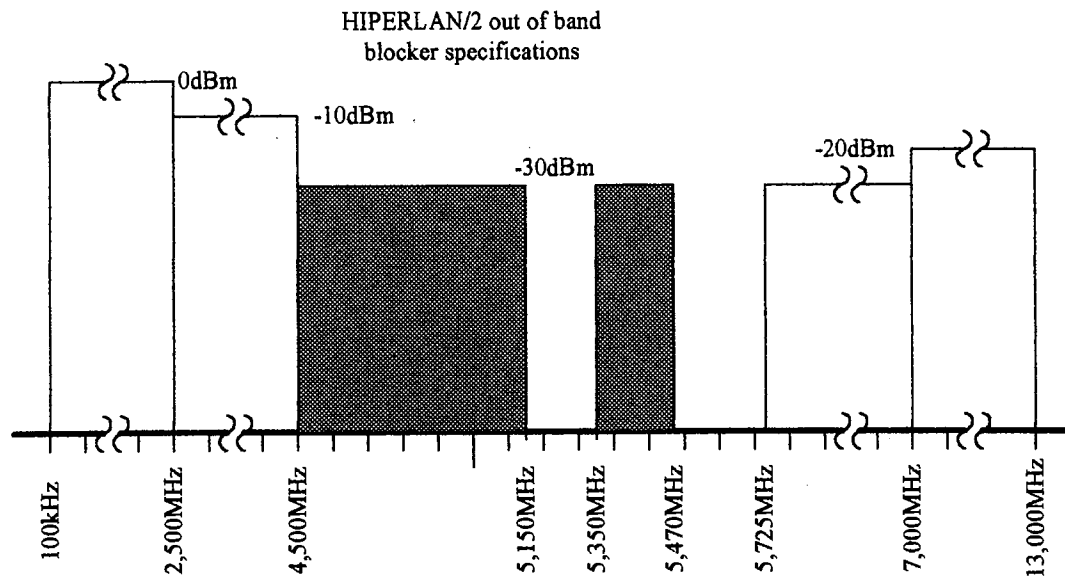


Figure 2.48 HIPERLAN/2 receiver blocker specification

3

Radio Frequency Front End Implementations for Multimode SDRs

Mark Cummings

enVia

All wireless communication systems have traditionally employed a radio frequency front end (RF FE) (see Figure 3.1), located between the antenna and the baseband subsystem, the latter commonly implemented with digital signal processing technology. While 'pure' software radios anticipate analog-to-digital conversion at the antenna, dispensing with the need for this element of the radio, today's 'pragmatic' software defined radios (SDRs), still (and as will be seen later in this chapter, for the foreseeable future) require an RF FE and place stringent demands thereon. The requirement for more cost-effective and reconfigurable RF FEs is one of the major needs of the wireless industry.

The perspective of our discussion within this chapter is that of commercial SDR pioneers and practitioners, seeking economically viable solutions for the commercial wireless marketplace. As such, we describe an early multimode RF FE solution, the AN2/6 product family (a two-band six-mode design), discuss alternative architectural options, and outline constraints and possible directions for the future evolution of SDR RF front ends.

The basic functions of the RF FE are

- down-/up-conversion
- channel selection
- interference rejection
- amplification

Down-conversion is required for receivers. A receiver subsystem takes the weak signal from the antenna, converts [*down-/up-conversion*] the signal from the transmission radio frequency (high – RF) to baseband frequency (low – typically low end of the desired signal will approach 0 Hz), filters [*interference rejection*] out the noise (both from external sources

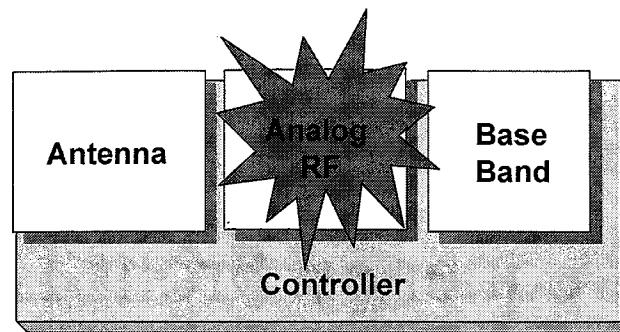


Figure 3.1 Wireless device generic architecture

and internally generated as an unwanted byproduct) and unwanted channels [*channel selection*], amplifies [*amplification*] the signal, and delivers the signal to the baseband subsystems.

Up-conversion is required for transmitters. A transmitter subsystem takes the signal from the baseband subsystem, converts the signal up from baseband frequency [*down-/up-conversion*] to the desired transmission radio frequency, amplifies the signal to the desired transmission level [*amplification*], filters out any noise introduced in the process [*interference rejection*], and delivers the signal to the antenna.

3.1 Evolution of Radio Systems

Since wireless communications systems first began to make significant appearances in the 1890s, evolution has progressed along two axes:

- modulation and encoding schemes
- technology for implementing modulation/demodulation and encoding/decoding

In this chapter, the focus is primarily on implementation technology. The evolution of modulation and encoding schemes plays an important role in the forces that drive the need for SDRs. However, that is considered elsewhere.

By the end of World War II, wireless communication systems had evolved to the point where they could be broken down into the following functions:

- human interface
- local control
- protocol stack
- low speed signal processing
- high speed signal processing
- RF FE
- antenna

Examples of the functions of each, in a cellular handset, include:

- human interface – controlling the speaker, microphone, keyboard, and display
- local control – managing the handset hardware and software resources
- protocol stack – managing such functions as call set up
- low speed signal processing – user data coding such as interleaving

- high speed signal processing – modulation such as FM, QPSK
- RF FE – see above
- antenna – interface to the ether

In these early systems, each function was implemented with discrete analog technology. This resulted in relatively large, expensive, high power consumption systems, which were difficult to design, manufacture, and manage/maintain in the field. The desire to lower cost size and power consumption while making devices easier to manage in the field has driven the technology evolution path we are still on today.

As digital technology entered the beginning of its period of rapid evolution, discrete analog components on complex printed circuit boards were gradually replaced. First discrete digital logic components were used to implement the human interface, local control, and protocol stack functions. With the appearance of the microprocessor, the discrete logic components were replaced with a microprocessor called a 'microcontroller' and software. Then the low speed signal processing (low speed signal processing plus high speed signal processing constitute the baseband subsystem defined above) analog discrete components were replaced with digital logic components. Then special mathematical functionality (such as multiply accumulate, MAC) were added to microprocessors to create digital signal processors (DSPs). Low speed signal processing functions were converted from discrete digital logic to DSPs and software.

Then the high speed signal processing analog discrete components were replaced with digital logic components. The expectation was that the same process would continue and that the high speed signal processing would be implemented by some kind of a microprocessor and software. The dream was that this process would continue to the point where a receiver subsystem would consist of an A/D converter at the antenna and everything else would be done by software. However, a fundamental barrier was found.

Early cellular standards such as AMPS required a baseband bandwidth of 30 kHz. About the time that low speed signal processing was firmly entrenched in DSPs and people were looking for the next step in evolution, second-generation cellular standards (modulation/encoding evolution) such as GSM with a baseband bandwidth in the 200–300 kHz range were beginning to appear. At the same time, second-generation standards such as IS-95 (CDMA/AMPS dual mode air interface standard (AIS)) with a baseband bandwidth of 1.24 MHz were also coming to market.

Although DSPs could theoretically handle the speed of processing required for high speed signal processing of second-generation AISs, practical limitations created an impenetrable barrier. DSPs are by their nature single stream instruction set processors. This means that there is a single processing unit which must load data (single sample of signal data) and a single instruction, process the data, write the result to a buffer, and start over again. The Nyquist theorem requires that a signal be sampled at more than twice its rate of change (in practice 2.5 times its bandwidth in Hertz) in order to preserve the data in the signal. This means that a 1.24 MHz signal must be sampled at approximately 3.25 MHz. In order to avoid quantization error, single samples are typically represented by 13 or more bits in cellular systems. Minimum high speed signal processing requires approximately 100 instructions per sample. Assuming that the processor has a 16-bit word size, this means that a DSP attempting to do high speed signal processing for a second-generation cellular handset would have to operate at a clock speed in excess of 325 MHz. In practice, because of bus delays, the need to

write to buffers, etc., it turns out to be in the GHz clock range. Early DSPs could not run at these clock speeds. As DSP development progressed, it became clear that since power consumption varied directly with processor speed, it would not be practical to operate a DSP at these clock rates for battery powered applications. It also turned out that even for systems run on utility-provided power, there were issues of size, heat dissipation, etc. which would make pure DSP solutions for high speed signal processing impractical.

Discrete digital component implementations avoided the power problem by implementing each of the 100 instructions in 100 discrete digital logic circuits arranged in a bucket brigade. In this way, each circuit can run at the clock rate of the sampled and quantized signal (such as 3.25 MHz), dramatically lowering power consumption. Further power/speed improvements can be made by having each circuit optimized for a single function, avoiding the inefficiencies inherent in a general purpose processor which has to be optimized for a wide range of functions.

As digital technology continued to develop another alternative appeared. Discrete digital logic components could not be replaced by a DSP with software, but they could be combined into a single chip. This combination of discrete digital logic into a single chip came to be called an application specific integrated circuit (ASIC). It achieved the cost, size, and power consumption advantages inherent in integrated circuits, but it did not have the flexibility inherent in software driven general purpose processors.

In the late 1980s and early 1990s a variety of markets for portable two-way wireless communications systems were experiencing rapid growth, and rapid evolution of technology for implementing modulation/demodulation and encoding/decoding resulting in a proliferation of AISs. The markets most prominently impacted were military, civil government, and commercial telecommunications (cellular, personal communications systems (PCS), wireless local area networks (WLAN), wireless personal branch exchange systems (WPBX), personal area networks (PAN), geosynchronous earth orbit satellite systems (GEOS), low earth orbit satellite systems (LEOS), geolocation systems GPS, etc.).

The lack of the flexibility inherent in ASIC processors for high speed signal processing made equipment manufactured to support a given AIS, for the rest of its useful life, limited to that AIS. This made it difficult for commercial users to communicate as they moved in and out of different AIS service footprints. Government users who did not have the lingua franca of the public switched telephone network to fall back on, if they had equipment supporting two different AISs, might not be able to communicate at all. ASICs, then, satisfied the desire to lower cost size and power consumption but did not satisfy the desire to make devices easier to interoperate or to manage in the field.

In the early 1990s, solutions to the high speed signal processing requirements that offered software driven flexibility and the ability to change AIS baseband subsystems to support different AISs through software began to appear [1]. The appearance of these solutions in the company of the market drivers led to the coining of the term software defined radio (SDR). This evolutionary process is graphically displayed in Figure 3.2 [2]. These solutions can be characterized as based on reconfigurable logic. There are a variety of approaches within the general area of reconfigurable logic. Generally, they use software or software-like code to configure digital logic to perform the high speed signal processing at relatively low clock rates, thereby achieving the desired power consumption/heat dissipation while they are running, while being able to be reconfigured between runs to support different AISs [1].

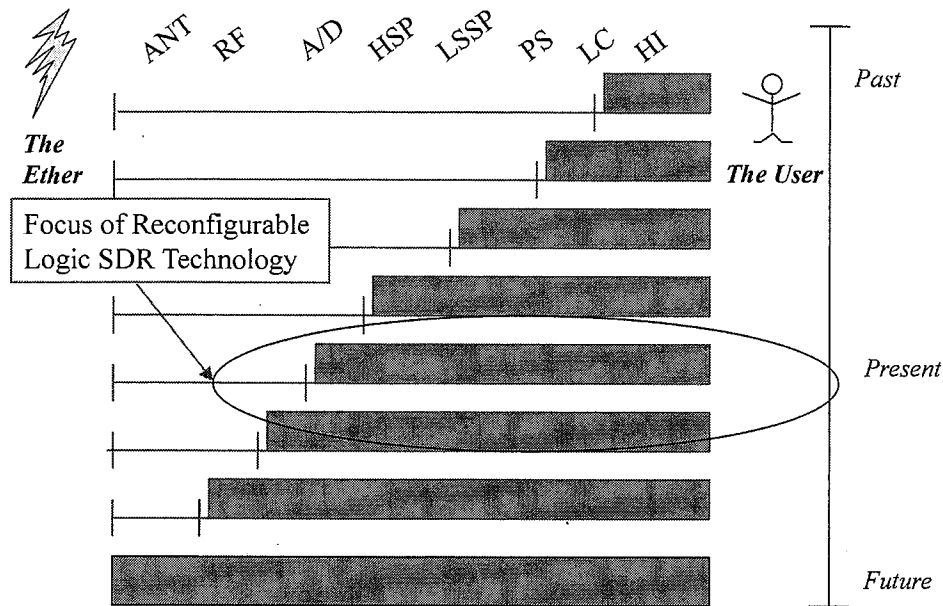


Figure 3.2 Wireless systems evolution

3.2 Evolution of RF Front Ends – Superheterodyne Architecture

At the time of the introduction of reconfigurable logic and the coining of the term SDR, the dominant implementation architecture used for RF FEs was the superheterodyne architecture [3]. The superheterodyne architecture was patented in 1915. It was developed to overcome the problems inherent in the direct conversion or homodyne architecture (sometimes called zero IF) developed in the 1890s. The problems inherent in direct conversion are discussed in detail in Chapter 2. We present a discussion of our own perspectives later in this chapter.

The superheterodyne receiver as shown in Figure 3.3 uses a chain of amplifiers, frequency synthesizers, mixers, and filters to down-convert, limit noise, and select the desired channel. It uses at least two steps of mixing and filtering to achieve the desired result. The first step mixes the signal down to an intermediate frequency (IF) and the second step mixes the signal down to baseband.

The superheterodyne transmitter uses a chain of amplifiers, frequency synthesizers, mixers, and filters to do the converse, i.e. to up-convert, limit noise, and amplify the signal for transmission. Again two stages are used. The first stage mixes the baseband signal up to IF and the second stage mixes it up to the desired RF band/channel.

The superheterodyne RF FEs were implemented with discrete analog components. Although there have been many years of work devoted to refining the superheterodyne architecture, by its very nature it is not easily integrated with contemporary chip technology. The frequency synthesizers and filters required by the superheterodyne architecture require very narrow, very sharply defined band pass characteristics. That is, they must pass the desired frequency and reject all the undesired frequencies as much as possible. This is sometimes described as filter quality or 'Q'. The steeper the filter wall, the higher its 'Q'. Superheterodyne architectures require very high Q components.

These components can be built with arrays of resistors, capacitors, and inductors (R, L, C).

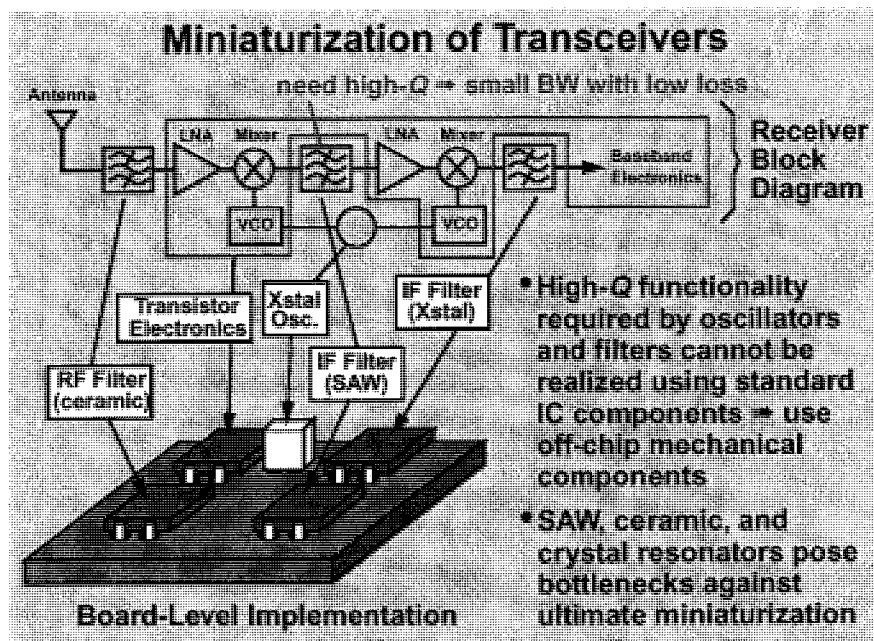


Figure 3.3 Miniaturization constraints with the superheterodyne architecture

To achieve high Q , very precise values of R , L , and C are needed. Integrated circuit technology requires extremely large chip area to implement accurate R s, L s, and C s. A single high precision inductor or resistor can consume the entire chip area of a common size chip, even with today's very small chip feature sizes ($0.13\ \mu\text{m}$ and smaller). One alternative commonly employed is the use of piezo electric devices in resonant chambers (surface acoustic wave (SAW)) devices for filters. Since the required resonant chamber size is a function of frequency of operation, as the frequency goes up, these devices can become smaller. Even so, typical SAW filters in cellular systems are several times the size of conventional chips.

What emerges is a practical limit. For a single mode (AMPS, TDMA, GSM, CDMA, etc.) and a single band (800 MHz, 1900 MHz, etc.) for cellular applications, superheterodyne RF FEs for personal portable two-way devices such as cell phones generally have as many as 300 electronic (passive and active) parts. In the early to mid-1990s this meant that the RF FE on average was 50% of the total cost of a cell phone and consumed up to 80% of the battery life in those phones.

As SDR high speed signal processing technologies emerged from the laboratories and promised multimode multiband systems, military, civil, and commercial users began to wonder about the implications for multimode multiband RF FEs. Each AIS had unique performance characteristics. The most significant were the requirements to tune to a specific RF frequency (band) and to deliver a specific bandwidth to the baseband subsystem. Bands vary widely from country to country, and from market to market. Advances in RF technology are also constantly opening up higher and higher frequencies for use. The baseband bandwidth is increasing as technology evolves. For example, early analog cellular systems required 30 kHz. Early time division multiple access (TDMA) digital systems requirements fell in the 200–300 kHz range. Early code division multiple access (CDMA) systems required 1.2–1.5 MHz. Wideband CDMA systems (WCDMA) currently being deployed require 4–6

MHz. Systems planned for the future have even wider baseband bandwidth requirements. The AISs also had differences in performance characteristic requirements such as noise, phase noise, linearity, etc.

The simple solution was to have a different RF FE for each mode and band envisaged. This meant, for example, that if two bands were required, two separate RF FEs would need to be implemented, each with 300 parts, implying a total parts count approaching 600. In addition the designer would have to take special care to make sure that the two RF FEs did not introduce signals into each other, producing potentially deadly noise. At the time there were two cellular/PCS bands and six modes in use in the United States. In order to have a handset that could work on all these modes and bands, using the above approach could potentially require 4800 parts! This had severe implications for cost, size, and power consumption.

Because of the military market's lack of the lingua franca provided by the public switched telephone network and the monopsony power inherent in military procurements, the military market led the development of SDR in the early stages.

3.3 The AN2/6 Product Family – Dual Band, Six Mode

The US Federal Government's Advanced Research Projects Administration (ARPA) contracted with a small Silicon Valley Company (enVia, Inc.) to, among other things, develop a prototype of a multimode multiband RF FE that would support multiple modes and bands with maximum efficiency. This prototyping effort grew into a product family named the AN2/6.

A prototype system based on five test boards interconnected by coaxial cable was demonstrated in satisfaction of the ARPA contract in 1998. enVia continued development and in 1999 announced a family of single board RF FEs that was awarded the prize for being the most innovative product of 1999 by RF Development and Design Magazine [4].

The AN2/6 supports all the cellular/PCS modes and bands in use in the United States in the 1990s. The AN stands for North America and the 2/6 stands for the two bands:

- cellular 800/900 MHz
- PCS 1900 MHz

and the six modes:

- cellular
 - analog – AMPS
 - TDMA – IS-54, IS-54A, IS-54B, IS-54C, IS-136
 - CDMA – IS-95
- PCS
 - GSM – PCS1900
 - TDMA – upbanded 136
 - CDMA – upbanded 95

Product implementations also supported the iDEN band and were extensible to WLAN and GPS AISs.

A single interface was provided for an antenna connection. The baseband interface was inphase and quadrature (I and Q) analog with analog control interfaces. The product family consisted of a handset RF FE, a microcellular base station RF FE, and a sniffer (the receiver portion of handset RF FE used by a base station to determine what other AIS base stations are within its immediate area) implementation.

3.3.1 The AN 2/6 Architecture

The design objectives were to make the maximum possible reuse of hardware to reduce the part count, size, power consumption, and cost to the minimum. Twenty-one paper designs were executed. The 21st design used an innovative frequency planning approach that allowed a single IF frequency to be employed for all bands and modes. The resulting paper design is shown in the block diagram contained in Figure 3.4.

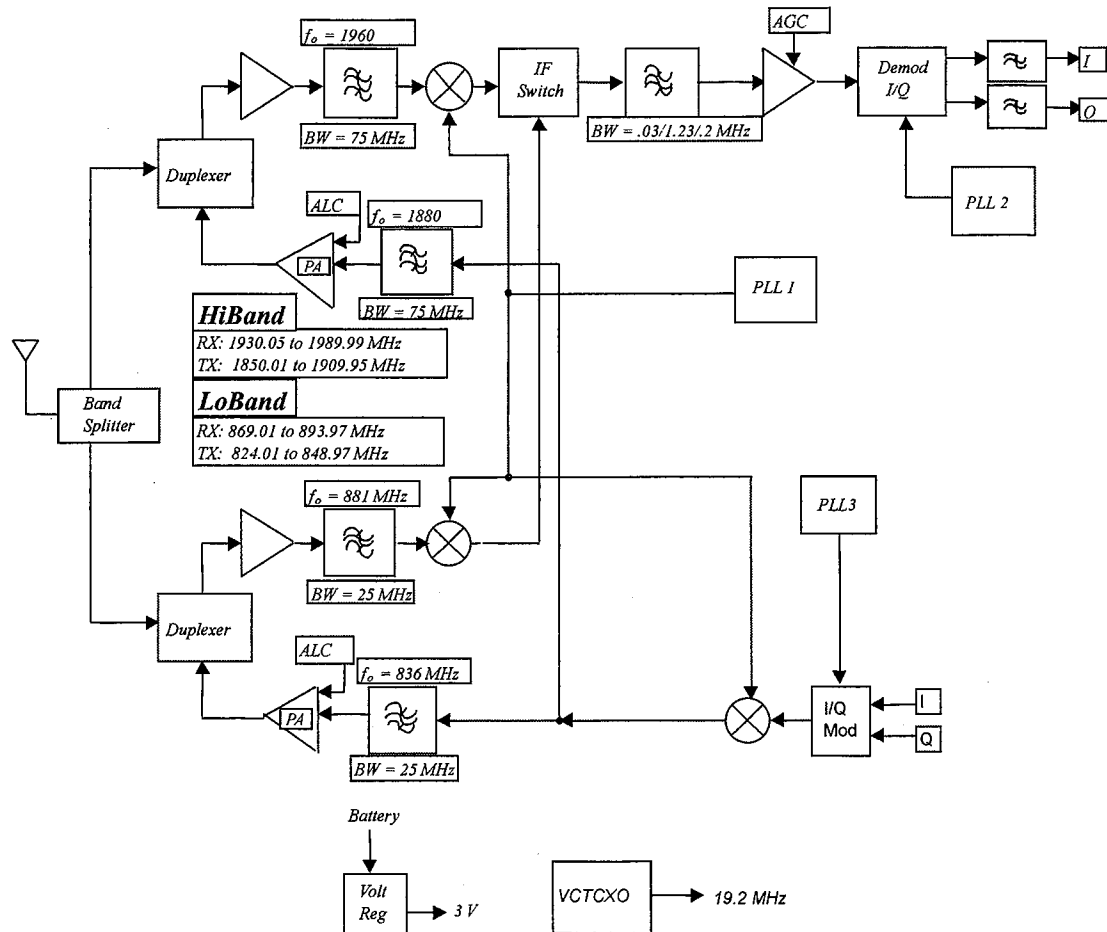


Figure 3.4 Design architecture of the AN2/6

In order to achieve the smallest part count, integrated chip solution products were explored. The most integrated solution in full product delivery was selected. An evaluation board was built. The evaluation board showed that the chip set did not meet the performance specified in

its spec sheet. A second chip set was selected and an evaluation board built. It too, did not meet spec. Finally a preproduction chip set was selected and a third evaluation board built. It too failed to meet its specifications, but with additional off-chip components, the failures could be worked around.

The RF FE design that resulted from the combination of the block diagram and the modified chip set was broken down into its five highest risk portions. Each of these high risk portions was assigned to a test board. Five iterations of test boards were designed, built, and tested before acceptable performance was achieved. The five boards were interconnected via coaxial cable to create a demonstration prototype.

The circuit design incorporated on the five test boards was then implemented in a single board system. The board was sized so as to be able to fit the contemporary handset profile. It was a ten-layer FR-4 printed circuit board. Five iterations of the single board were required before acceptable performance was achieved. The final handset implementation had 492 parts on the board. The primary cause of the need for iterations of the single board was unpredictable RF leakages between components or circuit paths which appeared by schematic to be electrically isolated. After the final layout, which achieved cellular noise performance specifications, a change in any circuit path or component location by even a fraction of a millimeter would probably have created an RF leakage problem.

The resulting product was for a superheterodyne architecture solution, a low cost wireless cellular RF front end module operating in six modes and two frequency bands. It was compliant with all North American standards and enabled rapid time-to-market for cellular PCS handsets, wireless appliances, and some base stations. The product was available in a variety of forms, supporting all or subsets of the six modes and two bands, and designed to be easily converted to other world standards.

The product provided fast digital switching between bands (824–894 MHz and 1850–1990

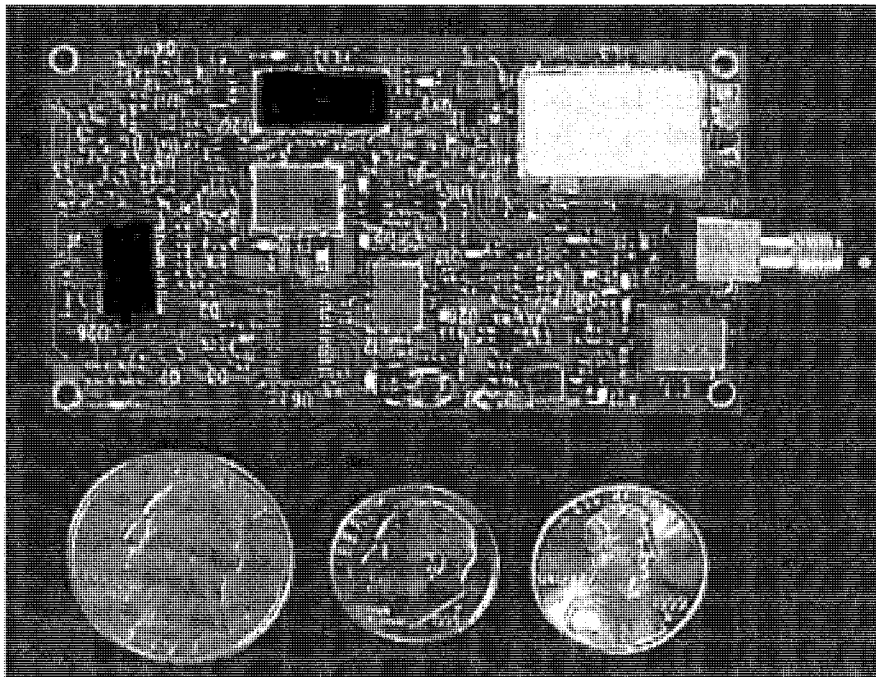


Figure 3.5 Prototype implementation of the AN2/6 handset board

Table 3.1 Specifications: physical size: width 4 cm; height 7.2 cm; thickness 0.8 cm

Receiver		Transmitter	
Frequency range	869–894 MHz	Frequency range	824–849 MHz
Noise	7.9 dB	Power output	
IP ₃	–7 dB	IS 95	26 dBm
IF bandwidth, 3 dB		IS 136	26 dBm
IS-95	1.23 MHz	AMPS	30 dBm
IS 136	30 kHz	Power control range	85 dB
AGC range	>85 dB		
Frequency range	1930–1990 MHz	Frequency range	1850–1910 MHz
Noise	7.9 dB	Power output	
IP ₃	–7 dB	IS 95+	27 dBm
IF bandwidth, 3 dB		IS 136+	27 dBm
IS-95+	1.23MHz	GSM	30 dBm
IS 136+	30 kHz	Power control range	85 dB
AGC range	>85 dB		

MHz) and modes and was compliant with the AMPS, IS-95, IS-95+, IS 136, IS136+, and GSM mode standards. It processed the Rx signal from antenna to analog I/Q outputs and the Tx signal from analog I/Q input to antenna output. The module design, on FR4 PCB material, featured a maximum vertical thickness of 10 mm with a small 61 × 110 mm-form factor. A 20% reduction in size was achievable through the application of automated assembly processes.

The AN2/6 was designed to be extensible to the other major 2G, 2.5G, 3G, WLAN, and WPBX standards in Asia, Europe, and North America. The AN2/6 product was the first implementation of the entire set of North American standards in a single RF FE. It provided a cost-efficient means for a cellular phone or wireless appliance manufacturer to reduce time-to-market with a multimode, multiband phone, wireless appliance, or microcellular base station.

The specifications of the handset version prototype board implementation of the AN2/6 shown in Figure 3.5, had the specifications listed in Table 3.1.

The interface specification shown in Table 3.2 was submitted to the SDR Forum as a recommendation for a standard interface between SDR handset RF FEs and SDR digital sections (baseband and controller).

3.3.2 Lessons Learned From the AN2/6

The ARPA prototype successfully demonstrated that it was technically and economically feasible to build multimode superheterodyne multiband RF FEs that would support SDR digital subsystems. It showed that by innovative design methods and a focus on maximizing hardware reuse, superheterodyne RF FEs with substantially lower part counts – as opposed to separate superheterodyne RF FEs for each mode and band – were achievable, leading to lower size, costs, and power consumption. The AN2/6 proved that products based on the prototype were commercially viable in the late 1990s.

Table 3.2 The interface to the digital section (baseband and controller) for an AN2/6 handset

Signal	Description	Specification
UNLOCK_PLL	Filtered output of OR'ed unlock lines for all PLLs	Positive true 3 V CMOS logic
AGC_1	AGC control voltage for RF input Stages	1–2.2 V increases LNA gain from –20 to 13 dB. $Z_{in} > 20$ K ohm
AGC_2	AGC control voltage for IF stages	0–3 V increases IF gain from –48 to +48 dB. $Z_{in} > 33$ K ohm
TX	Enabling bit for all TX PA functions	Positive true 3 V CMOS logic
ALC_1	Voltage control for power ampl driver stages	0–2.2 V decreases driver gain from +26 to –24 dB. Gain vs. voltage characteristic unique to each unit. $Z_{in} > 10$ K ohm
ALC_2	2nd stage of voltage control for power ampl output power	0–3 V PA (driver Vdd). $Z_{in} = 180$ K ohm. Gain vs. voltage characteristic unique to each unit
ALC_3	3rd stage of voltage control for power ampl output power	0–3 V PA (driver Vdd). $Z_{in} = 180$ K ohm. Gain vs. voltage characteristic unique to each unit
BATT_IN+	Battery voltage for power ampl circuits	3–4.2 V @ TBD mA
BATT_IN+	Battery voltage for power ampl circuits	3–4.2 V @ TBD mA
GND	BATT_PA ground	TBD mA
GND	BATT_PA ground	TBD mA
LO_BAND	Logic level for selecting low frequency band	Positive true 3 V CMOS logic
HI_BAND	Logic level for selecting high frequency band	Positive true 3 V CMOS logic
GND	Ground	TBD mA
QRX	Analog Q output of receiver	AC coupled, 1 V pp. $Z_{out} < 10$ ohms
IRX	Analog I output of receiver	AC coupled, 1 V pp. $Z_{out} < 10$ ohms
GND	Ground	TBD mA
GND	Ground	TBD mA
QTX	Analog Q input to transmitter modulator	1 V pp, $Z_{in} > 100$ K ohm
ITX	Analog I input to transmitter modulator	1 V pp, $Z_{in} > 100$ K ohm
BATT_RF	Battery voltage for all circuitry exclusive of the power ampl	3–4.2 V @ TBD mA
BATT_RF	Battery voltage for all circuitry exclusive of the power ampl	3–4.2 V @ TBD mA
GND	Ground	TBD mA
19_2REF	Buffered 19.2 MHz reference	Approx. 0.4 V pp into 50 ohm

Table 3.2 (*continued*)

Signal	Description	Specification
GND	Ground	TBD mA
GND	Ground	TBD mA
CLK_LO	Clock signal for PLLs	3 V CMOS level
DATA_LO	Data signal for PLLs	3 V CMOS level
GND	Ground	TBD mA
STB_PLL1	Strobe enable signal for LO1 PLL	Negative true 3 V CMOS logic
STB_PLL2	Strobe enable signal for LO2 PLL	Negative true 3 V CMOS logic
STB_PLL3	Strobe enable signal for LO3 PLL	Negative true 3 V CMOS logic
GND	Ground	TBD mA
AFC	Tuning voltage for 19.2 MHz reference oscillator	0–3 V, 1.5 V nominal, ± 7 ppm to 10 $\pm$ control range, positive slope, $Z_{in} > 100$ K ohm
GND	Ground	TBD mA
Mode_GSM	Logic level for selecting GSM mode	Positive true 3 V CMOS logic
Mode_IS95	Logic level for selecting IS-95 mode	Positive true 3 V CMOS logic
Mode_IS136	Logic level for selecting IS-136 mode	Positive true 3 V CMOS logic
GND	Ground	TBD mA

The ARPA prototype took two and a half years to develop. To go from prototype to first product took another year. It might have been possible to shrink the development time by adding more resources to the development team; however, the result would have been in the same order of magnitude. Multimode multiband superheterodyne RF FEs for cellular and similar applications take years, not months, to design.

Superheterodyne architecture solutions for RF FEs for cellular and similar applications lacked the capability to provide the flexibility previously obtained in the other subsystems (baseband and controller) through software driven architectures. Superheterodyne RF FEs had to be designed from the beginning for very specific modes and bands. Although they could be switched between bands and modes, once built the bands and modes supported could not be changed. This expectation of the static nature of RF FEs can be seen in the handheld architecture published by the SDR Forum (see Figure 3.6) [5].

There is a significant risk associated with the design of superheterodyne RF FEs for cellular and similar applications. In most electronic system design the key step is circuit design implemented in a schematic. In the design of superheterodyne RF FEs for cellular and similar applications, the schematic is necessary, but not sufficient, for success. The layout (i.e. physical implementation) and attention to isolation of components is absolutely critical. The schematic preparation is a science, but the layout is closer to art – an art that is learned slowly over many years of practice. Unfortunately, the skill base available to design superheterodyne solutions is limited. The explosion of digital technology has meant that engineering schools have significantly reduced the emphasis on teaching analog RF technology for 20 years. What has resulted is a bimodal distribution in the available skill base: a small and shrinking group of professionals with 30 or more years of experience and a second group with

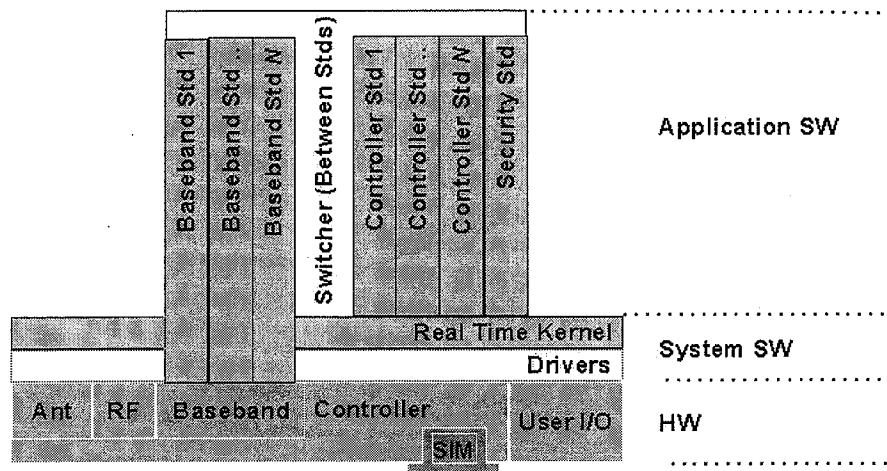


Figure 3.6 SDR forum handheld architecture

less than 10 years experience. The result of the intersection of the 'art' characteristic vector with the 'experience' vector produces high levels of risk.

This risk factor is exposed in several ways. The discovery by the AN2/6 development team that integrated chip sets to support superheterodyne RF FE for cellular and similar applications often do not meet specifications is one example. There are several stories current in the industry about failures. One fairly well substantiated case is the departure from the US market of a major Japanese consumer electronics company because RF FE problems in cell phones were hurting its corporate reputation for quality. There are also several stories of South East Asian initiatives in the cell phone area which have foundered on RF FE failures. In these cases, companies were rumored to have spent tens of millions of dollars on RF FE development over several years, only to abandon projects because they could not make the RF FEs perform adequately.

For some time, certain well-established companies in the cell phone industry saw the high risk factor as a very positive situation. From their perspective, it was a very good barrier to entry of new competition. Similar perspectives have played a role in the military and civil government markets as well.

The most fundamental lesson learned was that the superheterodyne architecture RF FE for cellular and similar applications was on a long term cost/performance curve that was not likely to be changed. In fact, it appeared that the superheterodyne RF FEs cost/performance curve might be reaching an asymptote.

If the cost performance curve for RF FEs for cellular and similar applications (the highest volume market with the most competitive pricing) is plotted over the last 20 years, it can be seen to be declining at a 1–5% annual rate (see Figure 3.7) [6]. This compares with the Moore's Law curve for digital technology of doubling capability and halving price every 18 months.

As conventional approaches began to reach the limits of their ability to integrate components and reduce part count in superheterodyne architectures, some attempted to achieve higher integration levels with exotic materials. Some of these materials have been and still are used for special components such as power amplifiers (PAs). In this case, the attempt was to

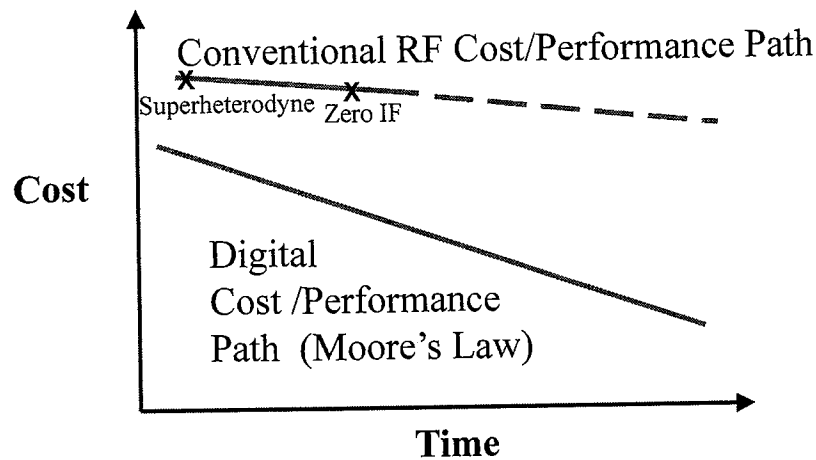


Figure 3.7 Cost/performance of the RF front end over the past 20 years

increase integration levels by using them to implement other parts of the superheterodyne architecture. Examples of materials include:

- GaAs – gallium/arsenide
- SiGe – silicon germanium
- SoS – silicon on sapphire
- SoI – silicon on insulator

These materials are able to increase speed and to increase the isolation of functions. These are very useful properties; however, they do not address the fundamental limiting factor, the need for high Q. Furthermore, application of these materials and their associated processes add significant cost and risk.

By the end of the 1990s, it became clear that the range of part counts achieved by the AN2/6 for superheterodyne architecture RF FE for cellular and similar applications was a practical limit. In the year 2000, 415 million cellular handsets were sold. There were new classes of applications appearing that demanded lower cost SDR cellular type solutions and the RF FE was a barrier. Examples of these new applications include:

- PDA morphing into a general purpose communicator
- wireless remote asset location systems
- wireless metrology (remote parameter measurement) systems
- wireless inventory systems (i.e. vending machines and others)

The forces that had spurred the wireless communications industry to integrated hardware driven by software for the other sections of the wireless communications device were still at work. It was now clear that the imperative could not be met by further refinement of the 80 year old superheterodyne technology.

3.4 Alternative RF Front End Architectures

3.4.1 Direct Conversion RF Front Ends

The first alternative to the superheterodyne architecture that received widespread industry attention was the direct conversion architecture, sometimes called homodyne, or zero IF (ZIF). Direct conversion eliminates one of the two stages of up-/down-conversion (see Figure 3.8) in the superheterodyne architecture. In so doing it can eliminate approximately one-third of the parts used in a superheterodyne architecture.

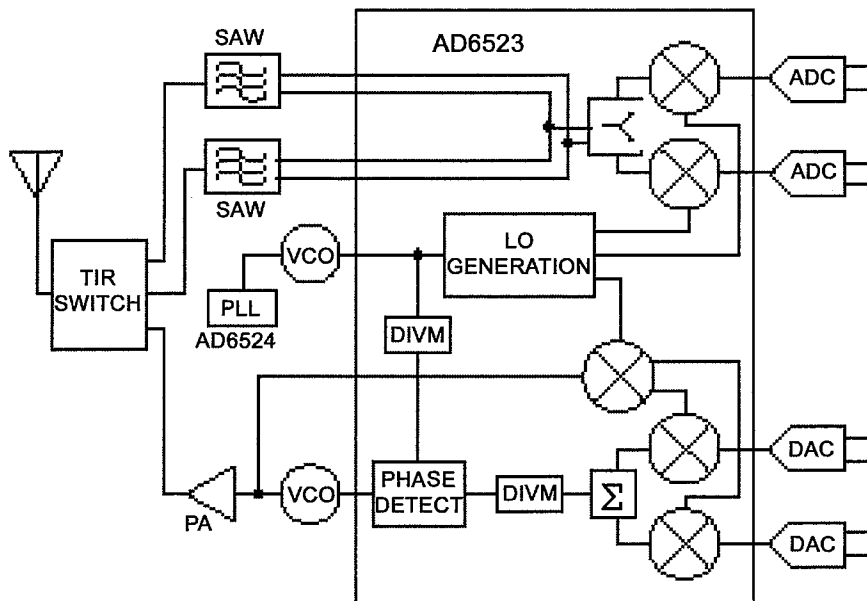


Figure 3.8 The Othello approach to the direct conversion architecture

In an industry highly motivated to find dramatically improved price performance RF FE solutions this sounded like a good deal, at first. However, as stated earlier in this chapter, the direct conversion architecture has fundamental performance problems that the superheterodyne architecture was created to overcome. In using direct conversion architectures to lower the part count, we inherit these problems. It is true that for some AISs these problems can be overcome, but only at a price. A brief review of the problems inherent in direct conversion architectures is presented below.

RF FE s must implement a receiver and a transmitter. However, the receiver is the more difficult. The receiver in today's typical cell phone consumes 60% of the battery life. For this reason, this discussion will focus primarily on the receiver.

Direct conversion receiver architectures perform the RF to baseband frequency translation in a single step. The RF signal is mixed with a local oscillator at the carrier frequency. There is therefore no image frequency, and no image components, to corrupt the signal. Direct conversion receivers offer a high level of integratability but also have several significant problems. So far, direct conversion receivers have proved useful only for modulation methods that do not place appreciable signal energy near DC.

In direct conversion architecture systems, noise near baseband ($1/f$ noise) corrupts the desired signal. It is well known that one of the largest components of noise in CMOS circuits is $1/f$ noise; and the smaller the f , the larger the noise introduced. This places the greater part of circuit noise at the exact location of the down converted image (i.e. zero frequency) and reduces sensitivity.

The local oscillator can be viewed as a large interferer exactly on the carrier frequency. If components of its energy leak into other portions of the receiver, it will be an in-band disturber. By definition, any phase noise in the local oscillator will be in the center of the desired band and located at zero, or near zero, frequency. This problem can be mitigated somewhat by using an extremely pure and costly frequency source as the local oscillator. But doing so adds more cost and potentially more components, thereby defeating the original purpose of going to a direct conversion architecture. Another consequence is that use of this architecture may be limited only to modulation schemes that have little information at zero frequency compared to the rest of the band, thus ruling out support for many AISs.

Since the local oscillator is at the in-band channel frequency, if it leaks out of the Rx portion of the direct conversion system into the Tx portion, it can result in unauthorized emissions by the RF FE. This can lead to problems in obtaining regulatory certification for products using this approach.

There are other problems associated with the local oscillator at the in-band channel frequency. If the local oscillator is just slightly different in frequency from the in-band channel frequency, a beat frequency product is created. This beat frequency can be introduced into the desired signal delivered to the baseband subsystem by the Rx, but it also can get into the Tx portion of the RF FE, resulting in unauthorized emissions. There are two very common ways that the local oscillator frequency once set at in-band channel frequency could become slightly different. They are:

- Doppler shift
- base station transmitter frequency drift

Studies have shown that a cellular handset in use in a sidewalk café in a typical metropolitan area, although stationary, experiences a Doppler shift of between 15 and 20 miles per hour because of the motion of all the objects around the user combined with the multipath nature of the RF environment. Systems in use in moving cars, buses, trains, elevators, etc. will experience an even higher Doppler effect. These shifts are combined with the natural frequency drift due to temperature, etc. in base station transmitters. This effect is compounded by hand offs between base stations whose transmitters may have drifted in opposite directions. These problems in local oscillator offset can be mitigated by having a local oscillator which is frequency agile and which can track the changes in the incoming carrier frequency. However, here again, doing so adds more cost and potentially more components, thereby defeating the original purpose of going to a direct conversion architecture.

Large on-chip capacitors or high order DSP filters are sometimes required to remove unwanted noise and signal energy near DC. These filters are typically placed between the mixer and the low pass filters but increase the overall bit error rate and power dissipation.

Direct conversion architecture systems are also prone to errors in generating the quadrature signals due to inaccuracies in the 90-degree phase shifter at RF frequencies.

The problems with direct conversion architectures are well documented. The fundamental proposition of those seeking to use them to reduce part count is that there are baseband

solutions to the direct conversion RF FE problems. There are a variety of implementation strategies and design approaches. They all are based on cleaning up the signal in the baseband section by employing knowledge about the noise conditions in the RF FE. The simplest just use the fact that there is a direct conversion architecture RF FE. The more sophisticated add special interfaces between the baseband and RF FE to tell the baseband about error-causing conditions in the RF FE such as local oscillator frequency mismatch, characteristics of the DC offset, etc.

The result is that for some AISs, direct conversion can be used to lower the part count, cost, and size of the RF FE. The size improvement is limited, as may be seen from Figure 3.9, which shows a single mode, single band ZIF RF FE implementation. Furthermore, the resulting increases in the complexity, size, power consumption, and costs of the baseband section partially offset the savings in the RF FE. When taken on a system level, the net effect is to remove the price/performance asymptote facing the superheterodyne architecture solutions and return to the shallow slope of the previous RF FE cost/performance curve (see Figure 3.7). Furthermore, there is a large class of AISs that is very difficult-to-impossible to support with direct conversion architectures.

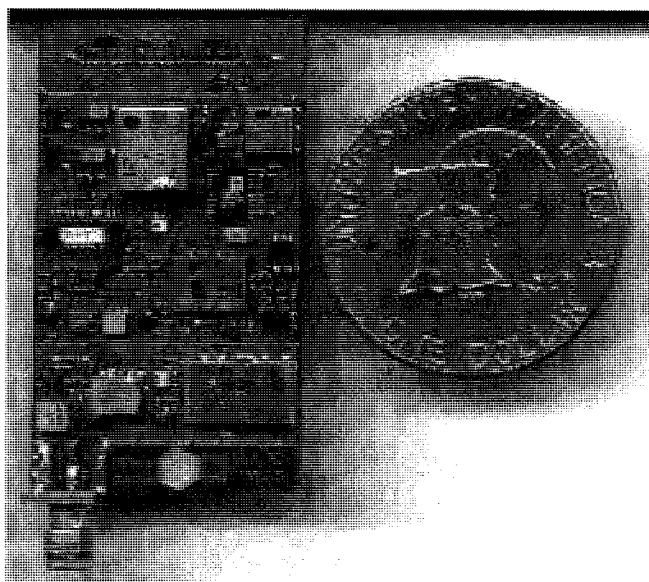


Figure 3.9 Prototype RF front end based on the Othello chipset

It should also be pointed out that, like the superheterodyne architecture, the direct conversion architecture lacks the capability to provide the flexibility previously obtained in the other subsystems (baseband and controller) through software driven architectures. Direct conversion RF FEs have to be designed from the beginning for very specific modes and bands. Although they can be switched between bands and some modes, once built the bands and modes supported cannot be changed.

3.4.2 Pure Digital RF Front Ends

As the limitations of integrated direct conversion architectures became clear, attention once again returned to pure digital solutions. On the one hand, pure digital architecture solutions potentially offer all the advantages of flexibility offered by software driven integrated hardware. On the other hand, this approach still seems to face insurmountable challenges. If the high speed signal processing functions in the baseband section created very difficult power consumptions/speed challenges, then the frequency/speed/bandwidth requirements in the RF FE – being several orders of magnitude higher – move from very difficult to impossible.

Such a solution puts an analog-to-digital converter (ADC) at the antenna and does all down conversion and filtering in a digital signal processor (DSP). The ADC and DSP functions can be implemented in CMOS, yielding the theoretical possibility of a three-, two- or one-chip RF FE solution with a resulting decrease in cost and risk; and the solution would be agile/flexible under software control. Unfortunately, the digital solution requires the ADC and the DSP to run at extremely high speeds, thereby consuming extremely large amounts of power.¹

It is very difficult to operate an ADC at today's typical cellular/PCS frequencies in the 2 GHz range. Low cost, low power ADCs for these frequencies are not generally commercially available today and show little likelihood of being available in the near future.

Even so, assuming that an ADC is available, a simplified Rx example solution with an incoming signal at 2 GHz would sample the signal at 5 GHz and encode each sample in 16 bits. An optimistic estimate of the processing required in a single stream instruction set processor for the filtering and down conversion process would require 500 instructions per sample. Therefore the DSP would have to run at 2500 GHz. The most optimistic estimate is that a full custom ASIC solution might be able to get the clock rate down to as low as 10 GHz. Given that the receiver is on all the time, and that chip power consumption is a direct function of clock speed, the power consumption for the full digital solution is so great that it is not practical for cell phones for as far into the future as we can see. In fact, not only are there no commercially announced full digital solutions for handsets, there are none for base stations where power consumption is less constrained.

3.4.3 Analog Digital Combination Solutions

Some have tried to combine analog mixing and down-conversion with digital down-conversion and filtering.² This architecture uses the first stage of a superheterodyne architecture solution (one stage of synthesis, mixing, and filtering) to bring the received signal down to IF. Then instead of the second analog stage (second stage of synthesis, mixing, and filtering), a digital down-conversion and filtering stage is employed. By reducing the frequency of the signal, the speed at which the digital section must run is also reduced. In this way it is theoretically possible to avoid the noise burden the ZIF solution places on the baseband section while eliminating the parts associated with the second down conversion stage. This approach can provide performance comparable to a superheterodyne system.

Unfortunately, the power consumption in the digital section (A/D converter, digital down converter and digital filter) of this solution is still very large and the digital parts can be

¹ Chapter 5 by Brock addresses a revolutionary approach to the problems of the pure digital architecture, based upon the use of superconducting Josephson junctions, rather than traditional CMOS technology.

² Chapter 6 on the digital front end, by Fettweis and Hentschel, discusses a range of issues related to such concepts.

expensive to manufacture, thus negating the cost and power reduction goals. The power consumption comes directly from the speed of the signals being processed (still several orders of magnitude higher than baseband). The cost of the digital chips comes from the size of the ASICs (die area) required to implement the digital processing. Furthermore, some implementations seek to meet legacy baseband interfaces by adding a digital to analog converter at the end of the digital chain so as to provide analog I & Q to the baseband interface, thus further increasing the costs, size, and power consumption of this solution approach.

3.4.4 Directions for a Completely Successful SDR RF Front End

The SDR RF FE alternatives discussed above define a solution space (see Figure 3.10). On the upper left hand corner is the superheterodyne architecture. It provides good signal quality but has relatively high power consumption and very high costs and size. Finally, it lacks the desired characteristic of the type of flexibility offered by software driven standard hardware architectures.

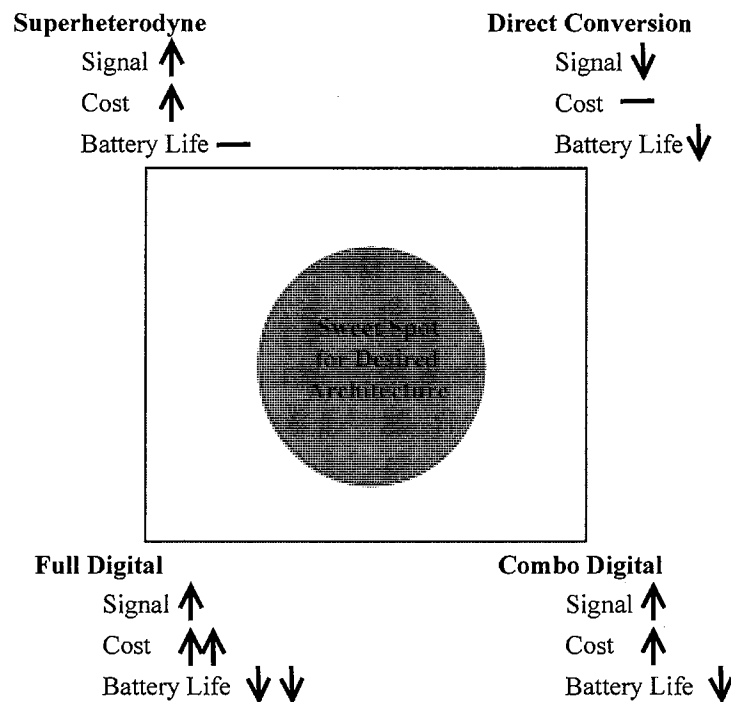


Figure 3.10 The SDR RF front end solution space

The upper right hand corner shows the direct conversion architecture. It provides lower cost and size in the RF FE, but the signal quality is degraded and the for those AISs where it can be corrected sufficiently to operate, large amounts of power must be used in digital filtering added in the baseband (in some implementations additional ADC and DSP hardware are added to the RF FE itself to perform the filtering). It also lacks the desired flexibility offered by software driven standard hardware architectures.

The lower left hand corner shows the pure digital architecture. It promises the type of flexibility offered by software driven standard hardware architectures. It is also theoretically very small in size and part count. However, while theoretically possible, its power consumption/processing speed requirements put it beyond practical reach, let alone cost effectiveness.

The lower right hand corner shows the combination analog digital architecture. It overcomes the direct conversion architecture's signal quality problems while keeping its size lower than the superheterodyne architecture. However, its power consumption and costs are high.

What these architectures suggest is that there is a solution within this space (the sweet spot) which has the type of flexibility offered by software driven standard hardware architectures, while putting the RF FE on or near the Moore's Law curve for power consumption, size, and costs. To find this solution will likely take a creative approach not widely pursued by the industry at this time.

3.5 Conclusion

In this chapter we have reviewed recent experience with the design and development of RF FEs for multimode, multiband, and SDRs. We have seen that the superheterodyne architecture is capable of providing multimode multiband RF FEs that can support SDR digital sections. While providing good performance these superheterodyne architecture solutions can not provide the desired flexibility offered by software driven standard hardware architectures. We have also seen that costs and size decreases of superheterodyne architecture solutions are unlikely to meet the needs of future cellular and other mobile markets. We have examined the three other approaches receiving broad industry attention as alternatives to the superheterodyne architecture for SDRs and found that although each lacks the full set of desired characteristics, taken together they define a solution space in which an architecture with the desired characteristics is likely to be found.

Acknowledgements

The author would like to acknowledge assistance in building the author's understanding of the subject matter from Jim Kubinec, Nathan Silberman, and Rich Walsworth, and editorial help from Bill Ribble.

References

- [1] Cummings, M. and Haruyama, S., 'FPGA in the software radio,' *IEEE Communications*, March, 1999.
- [2] Cummings, M., 'The need for coordination in standards for nano, pico, micro & macro cellular systems in a software defined radio context', *Proceedings of the Keynote Address to PIMRCR '99*, Osaka, Japan.
- [3] Skolnik, M.I., *Radar Handbook*, 2nd Edition, McGraw-Hill, New York, 1990.
- [4] 'Technology awards', *Wireless Design and Development*, December, 1999.
- [5] TR2.1, *Software Defined Radio Forum*, November, 1999.
- [6] Cummings, M., 'Time for software defined radio,' *EE Times*, March 11, 2001.

8

Parametrization – a Technique for SDR Implementation

Friedrich Jondral

University of Karlsruhe

Mobile communications is mainly a service driven business. But there are important marginal conditions from physics and technology that may not be ignored when developing a mobile communications system; the frequency spectrum is a scarce resource. Therefore terminals, mobile as well as base station transceivers, have to work efficiently with respect to spectrum. Mobile radio channels are complicated due to multipath propagation, reflections, scattering, time, or frequency dispersion, etc. Sitting in an office environment surfing the Internet calls for a different solution for mobile communications than traveling in a train at a velocity of 200 km/h while engaged in a phone call. This is one of the reasons for the multitude of air interfaces which have been developed in recent years.

Mobile communications is based upon transmission standards, e.g. the air interface as well as the protocol stack of transmissions are at least agreed upon by the actual communications partners. A standard completely describes how a signal looks on air and which procedures are used to set up, maintain, and to close down a connection. Therefore it seems natural to ask whether there are construction principles that are common to all standards and, if so, to describe the standards according to these principles.

If a terminal in mobile communications has sufficiently flexible hardware at its disposal, it can serve different standards, i.e. it can operate as a *software defined* or *reconfigurable radio* if the hardware can be used to perform the processing for the different standards. It is this flexibility that will be used in future devices to meet different transmission needs.

Starting from definitions of several radio types this chapter discusses the specific features of a software defined radio transceiver architecture and the structure in which such a transceiver has to be represented in order to become 'parameter controlled'. We then briefly compare different methods for achieving adaptability of mobile wireless terminals. Parametrization of mobile standards is then investigated in some detail since on one hand this method can presently be used for adaptability and on the other hand it may be a first step in the direction of a radio knowledge representation language. We consider a specific example of

- [25] Lund, D., Honary, B. and Darnell, M., 'Adaptive nested codec', *ISPLC99: 3rd International Symposium on Power Line Communications and Its Applications*, Lancaster, UK, 1999.
- [26] Lund, D., Honary, B. and Darnell, M., 'Efficient FPGA hardware usage applied to the design of trellis Viterbi decoders', *PREP99*, Manchester, UK, 1999.
- [27] Mitola, J., *Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering*, John Wiley & Sons, 2000.
- [28] Mitola, J., 'Software radio architecture evolution: foundations, technology trade-offs, and architecture implications', *IEICE Transactions in Communications*, Vol. E83-B, No. 6, June, 2000, pp. 1165–1172.
- [29] Mitola, J., 'The software radio architecture', *IEEE Communications Magazine*, May, 1995, pp. 26–38.
- [30] Mitola, J. and Maguire, G., 'Cognitive radio: making software radios more personal', *IEEE Personal Communications*, August, 1999, pp. 13–18.
- [31] Mitola, J., 'Software radio architecture: a mathematical perspective', *IEEE Journal on Selected Areas of Communications*, Vol. 17, No. 4, 1999, pp. 514–538.
- [32] Pereira, J.M., 'Re-defining software (defined) radio: re-configurable radio systems and networks', *IEICE Transactions in Communications*, Vol. E83-B, No. 6, June, 2000, pp. 1175–1182.
- [33] Perez-Neira, A. et al., 'Smart antennas in software radio base stations', *IEEE Communications Magazine*, Vol. 39, No. 2, February, 2001, pp. 166–173.
- [34] Prasad, R., *CDMA for Wireless Personal Communications*, Artech House, Boston, MA, 1996.
- [35] Quicklogic Corporation, 'Quicklogic devices'. <http://www.quicklogic.com/devices>, 2001.
- [36] Ramos, R., Madani, K. et al., 'Reconfiguration resource control in a distributed intelligent re-configurable mobile network', *European Conference on Wireless Technology*, London, 2001.
- [37] Ramos, R. and Madani, K., 'A demonstrator system for an intelligent re-configurable physical layer in software radio applications', *International Symposium 3rd Generation Infrastructure and Services*, 3GIS, July, 2001, Athens, Greece.
- [38] Ramos, R. and Madani, K., 'A novel generic disturbed intelligent re-configurable mobile network architecture', *VTC-Spring Conference*, Rhodes, Greece, May, (2001), pp. 6–9.
- [39] Sandor, I. and Kovacs, J., 'Resource control and reconfiguration in software radio environment', *International Symposium 3rd Generation Infrastructure and Services*, 3GIS, Athens, Greece, July, 2001.
- [40] Tang, X., Aalsma, M. and Jou, R., 'A compiler directed approach to hiding configuration latency in chameleon processors', *FPL*, 10th, 2000.
- [41] Tuttlebee, W., 'Software radio technology: a European perspective', *IEEE Communications Magazine*, February, 1999, pp. 118–123.
- [42] Xilinx Inc., <http://www.xilinx.com>
- [43] Xilinx Inc., 'Virtex-II 1.5v field programmable gate arrays: advanced product specification', DS031-1, April, 2001.
- [44] Xilinx Inc., 'Virtex configuration architecture advanced users' guide', XAPP 151, v1.2, Xilinx Inc., September, 1999.

parametrization, demonstrating the application of the concept as an SDR implementation approach for second- and third-generation air interfaces GSM, IS-136, and UTRA (UMTS terrestrial radio access). We conclude by considering some hardware issues for such an SDR.

8.1 Definitions

Software defined radio technology is used to support the service and quality of service (QoS) requirements of the users. However, flexibility can be achieved in many ways and thus the goal of flexibility has also to be discussed. Is it useful to have radios that can serve different channels, on different frequency bands with different standards, at the same time?

One possibility is to work with 'plug ins'. For example, a network subscriber could be equipped with a basic hardware module that especially contains a radio frequency front end by his provider. To complete the device, i.e. to make it a functioning transceiver, the subscriber has to buy an additional *hardware plug in* on which the digital part of the signal and information processing is implemented. So, if he wants to use his terminal at home as a HiperLAN 2 station, he buys a HiperLAN 2 plug in. If, on the other hand, he travels to North America and wants to use his terminal in an area covered by cdma2000, he has to purchase a cdma2000 plug in. Of course, such a solution is by no means a software defined radio.

To continue our discussion, it is necessary to introduce some definitions [1]. One most important development toward software defined radio technology was the introduction of digital signal processing to communications which began on a broad basis around 1980.

The first product was the *digital radio* (DR), which is a transceiver that has implemented the baseband processing in digital hardware (digital signal processors, DSPs). Digital radios are broadly used in all second-generation cellular networks. Of course, digital radios can be programmed, but essentially they are constructed to serve only one specific standard, i.e. they can be multiband but not multimode. So digital radios are not necessarily adaptive terminals.

An ideal *software radio* (SR) digitizes the signals directly at the antenna, i.e. no analog mixing or intermediate frequency (IF) processing is employed. The problem here is that the frequency band used for cellular mobile communications extends to around 2.5 GHz, which calls for sampling rates of ~6 Gsamples/s. This is completely beyond the capability of present technology, if one considers that a dynamic range of at least 60 dB is necessary for the analog-to-digital (A/D) converter. Moreover the sampling rate would further increase if wireless local area networks (WLANs), which mostly will use frequencies above 5 GHz, are taken into consideration.

The technological compromise that will provide a practical solution for the future is the *software defined radio* (SDR) that, after a first band selection filter, uses an analog mixing process, most probably directly to the complex baseband, and then digitizes the I/Q signal components. It is under discussion whether complete bands allocated to a standard, or only physical channels out of this band which have to be processed in the receiver, should be digitized. So SDR is a technology that will be available in the foreseeable future and that, as we shall see, is quite necessary for achieving, at least, the service quality required by the family of IMT-2000 air interfaces. However, although this contribution mainly deals with signal processing at the physical layer (MODEM level), it should be pointed out that, for an SDR, adaptability of the protocol layers is also a very important feature [2]; this is discussed in a later chapter of this volume.

In his doctoral dissertation [1], Joe Mitola introduced the concept of *cognitive radio* (CR), which brings together the technologies of SDRs, personal digital assistants (PDAs), and intelligent networks (INs) to form an ‘electronic butler’, who is always worrying about the welfare of his owner. A cognitive radio will not only act as a terminal that connects you to a network, it will at the same time observe the environment and its owner’s habits. It will learn about its owner’s preferences for preparing for or for making decisions, should the occasion arise. Of course, with its requirements for automatic speech recognition, reasoning, planning, decision making, etc., cognitive radio presently only represents a vision, but this vision may become a reality much sooner than we might think today.

So far, we have been engaged in considering general concepts, but now let us concentrate on discussing the physical layer of SDRs that should be applied in mobile cellular networks of second or third generation, or in cordless environments. To be at least a little bit more precise at this point in our discussion, we should point out that all the standards considered in this chapter use carrier frequencies between 800 MHz and 2.3 GHz.

8.2 Adaptability

For a software defined radio as discussed in this chapter adaptability means that the radio is able to process signals of different mobile standards. In the following we discuss several methods by which adaptability can be achieved.

Software download is a procedure by which the software relating to a specific standard may be downloaded to the hardware of the radio in the same way that a PC user’s program is downloaded to the processor of a general purpose computer. For software download there are two alternatives: the download can be done via the air or from a smart card. Several problems have to be solved in this connection. The download has to be performed error free, since otherwise there would be incorrect functions for the radio. Error free download can be guaranteed much more easily if the software is downloaded from a smart card than if it is downloaded over the air. Another problem concerning over the air download is the definition of the relevant air interface. If it is not possible to agree upon an air interface for worldwide mobile radio coverage, why should it be any easier to agree upon an air interface for software download that can be used all over the world? In future mobile networks interstandard handover will become necessary (e.g. from UMTS-FDD to GSM). How can this be done with software download? Must the different standards be loaded onto the hardware in advance?

One way to circumvent the drawbacks of software download is the *parametrization* of standards. This method is discussed in [3] in great detail and is described in the remainder of this chapter.

The final goal of adaptability will be the definition of a radio knowledge representation language (RKRL) [1], by which only a service that has to be supported by the radio has to be specified. From a toolbox of modulators/demodulators, encoders/decoders, as well as transmit/receive protocols, a transceiver is constructed, the signal processing of which is optimal for the given channel, the chosen quality of service (QoS), etc. There are indications that parametrization of standards may lead to the definition of such a RKRL. This topic has to be investigated in the future.

8.3 Parametrization of Standards

As a communications standard we define a set of documents that describes the functions of a system in such a way that a manufacturer can develop terminals or infrastructure equipment on this basis. Standardization is one necessary condition for making a communication system successful on the market as exemplified by the global system of mobile communications (GSM). Standardization encompasses all kinds of communication networks. In this chapter we concentrate on *cellular mobile* systems.

Of course a standard has to contain all the functions of the system; especially for a mobile communications standard at least the air interface, as well as the protocol stack, have to be specified. For our further discussion we focus on air interfaces and use the notion of *standard* more or less as a synonym for *air interface*. Moreover, in the following we look at GSM and IS-136 as well as at UTRA-FDD as examples of second- and third-generation standards respectively.

8.3.1 Second Generation – Global System for Mobile Communication (GSM)

The global system for mobile communication (GSM) was introduced as a radio standard in Europe in the early 1990s. The most important features of the GSM air interface are summarized in Table 8.1.

The modulation mode used in GSM is gaussian minimum shift keying (GMSK). GMSK is a special form of frequency shift keying, i.e. this modulation mode is nonlinear. Because of the gaussian shaped transfer function of its lowpass filter, the sidelobes of its spectrum are

Table 8.1 Key air interface parameters of the GSM system

Uplink	890–915 MHz (1710–1785 MHz) ^a
Downlink	935–960 MHz (1805–1880 MHz)
Channel width	200 kHz
Channel access	FDMA/TDMA
Duplex mode	FDD/TDD
Duplex distance	45 MHz/1.73 ms (95 MHz/1.73 ms)
Users per carrier frequency	8
Speech encoder	RPE-LTP ^b
Net speech rate	13 kbit/s
Modulation	GMSK (BT = 0.3)
Error correction coding	CRC, convolutional
Number of carrier frequencies	124 (374)
Bit duration	3.692 μ s
Number of bits per time slot	156.25
Burst duration	576.9 μ s
Channel bit rate	270.833 kbit/s
Max. cell radius	35 km (10 km)

^a Data in parentheses refer to GSM1800.

^b Regular pulse excitation–long term prediction.

reduced as compared to (pure) MSK. Gaussian shaping smoothes the phase diagram and therefore realizes soft transitions between signal states, leading to a reduction in bandwidth. The bandwidth time product of GSM is specified as $BT = 0.3$. One drawback of GSM is that this modulation mode is nonlinear. Fortunately a linearization method for GMSK exists, but the physical price for this linearization is that the modulus of the complex signal envelope is no longer constant. In [3] the linearization of GMSK is discussed in detail and it is shown that the linearized GMSK meets all the requirements of the GSM standard.

As a channel access mode, GSM employs FDMA and TDMA with eight timeslots per carrier. Uplink and downlink are separated by FDD and by TDD, which uses a time delay of three slots. Therefore the terminal transmits and receives signals at different times. In GSM-900 there are two transmission bands, each of 25 MHz bandwidth, that are used for FDD. Thus GSM has 124 frequency channels in total. Since guard bands have to be introduced at the beginning and at the end of the frequency intervals, 122 frequency channels remain for the network providers. On each (duplex) frequency we have eight time slots, i.e. there are 976 speech channels available. Several different slot types are used – the normal burst, the frequency correction burst, the synchronization burst, the dummy burst, and the access burst.

The normal burst, which is the carrier burst of the speech (or data) transmission, is shown in Figure 8.1. At the beginning of the burst there are three tail bits followed by 57 information bits. In the middle of the burst we see one (out of eight possible) training sequences of 26-bit length, preceded and followed by one stealing flag bit, respectively. The training sequence is used in the receiver as a reference signal for channel estimation and bit synchronization, the stealing flag bits indicate whether the burst carries user or control data. The next bit block is made up of the second 57 information bits, followed by three tail bits and a guard time that corresponds to the duration of 8.25 bits. The guard time is necessary to avoid time slot overlap caused by the switching of the transmitter's amplifiers. In total, a burst is built up of 156.25 bits that are transmitted within 576.9 μ s. The equalization of a slot is supported by the tail

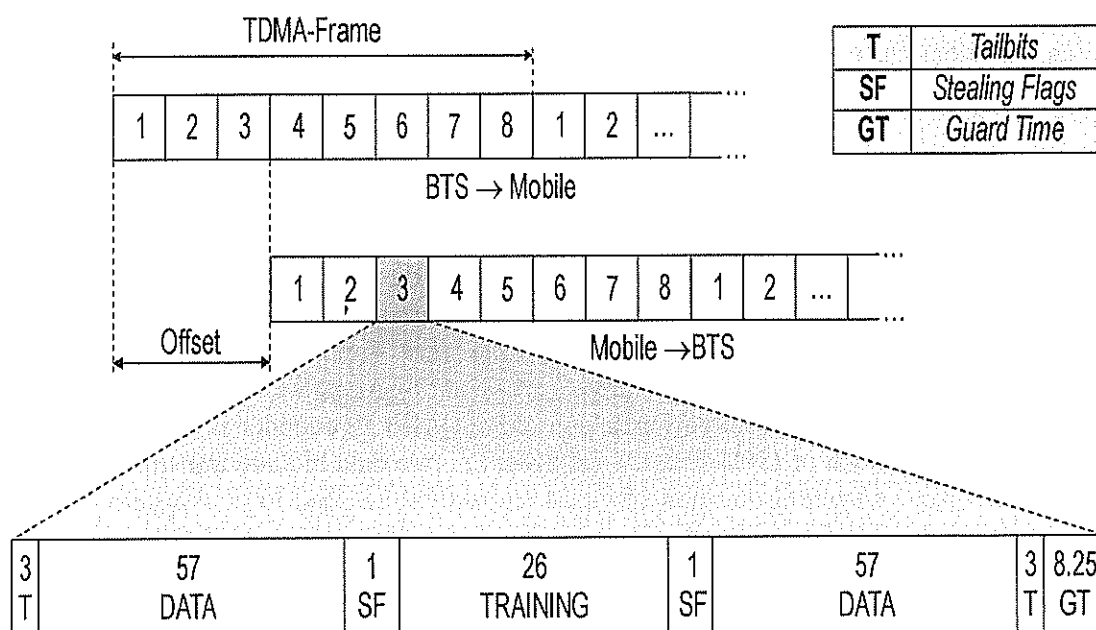


Figure 8.1 The GSM normal burst

bits. If the maximum likelihood sequence estimation (MLSE) method is used for equalization, the tail bits serve for the termination of the sequence estimator in a predefined final state. Equalization starts from the midamble in such a way that the first data bit block is equalized from its end, while the second data bit block is equalized from its beginning. Eight bursts fit into one TDMA frame.

As an example of channel coding in GSM we look at the full rate speech traffic channel TCH FS. The speech encoder produces 260 bits within 20 ms, but there are differences between the bits with respect to their importance. Some are more important for the reconstruction of the speech signal than others. Therefore, the precoded bits are put into an order according to their importance. In GSM there are two bit classes:

- class 1 and class 1a (with 182 bits in total)
- class 2 (with 78 bits)

The 50 most important bits are assigned to class 1a, encoded by a cyclic block code with three check bits. The generator polynomial of this systematic cyclic block code is given by

$$g(x) = x^3 + x + 1$$

All class 1 bits (i.e. also the class 1a bits) are error protected by a convolutional encoder of rate 1/2. According to the GSM standard the generator polynomials used are

$$g_1(x) = x^4 + x^3 + 1$$

and

$$g_2(x) = x^4 + x^3 + x + 1$$

In total the encoding procedure transforms 182 class 1 data bits (plus three additional check bits and four bits for the termination of the convolutional code) to 378 code bits. The class 2 bits are transmitted without coding. This results in a total of 456 bits, which corresponds to a gross data rate of 22.8 kbit/s within a 20 ms speech frame. Besides convolutional and block coding, interleaving of the traffic data is also performed, implemented via a rearranging matrix.

8.3.2 Second Generation – IS-136 (DAMPS)

IS-136 is the North American equivalent of GSM. This system is also called digital AMPS (DAMPS), which indicates that IS-136 is a derivative of the earlier analog advanced mobile phone service (AMPS) standard. IS-136 specifies the air interface of a digital, terrestrial cellular mobile radio standard. Table 8.2 shows the most important technical data of this system. The access is done via FDMA/TDMA. Up- and downlink are separated by FDD. For speech coding a vector sum excited linear predictive (VSELP) encoder is used.

The modulation mode of IS-136 is $\pi/4$ -DQPSK, i.e. two data bits are transmitted by each symbol. Therefore a symbol duration of 41.14 μ s results, i.e. IS-136 is a system that is in the border area between broadband and narrowband systems. Unlike GSM, equalization is not absolutely necessary. In contrast to GMSK, the modulation mode $\pi/4$ -DQPSK is linear. The envelope of a $\pi/4$ -DQPSK signal is not constant, but due to the phase offset of at least $\lambda = \pi/4$ for two consecutive symbols the complex envelope avoids zero crossings.

Within one TDMA frame three user signals may be transmitted. A user's downlink burst is

Table 8.2 Key air interface parameters of the IS-136 system

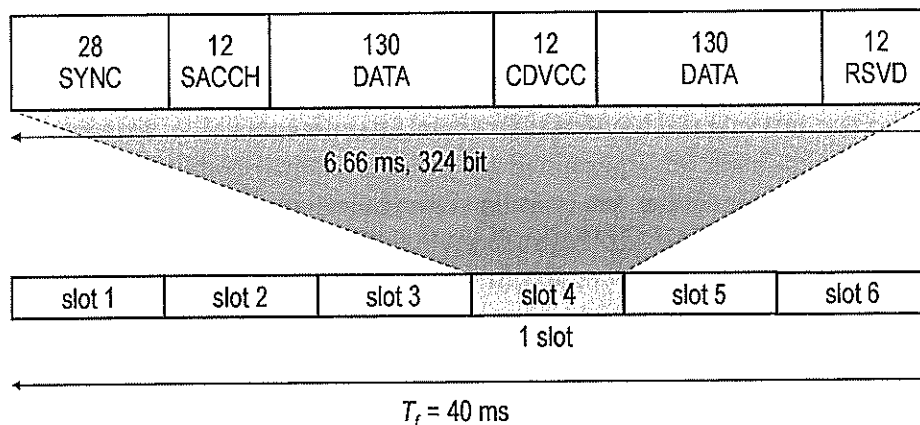
Uplink	824–849 MHz
Downlink	869–894 MHz
Channel width	30 kHz
Channel access	FDMA/TDMA
Duplex mode	FDD/TDD
Users per carrier frequency	3
Speech encoder	VSELP
Net speech rate	7.95 kbit/s
Modulation	$\pi/4$ -DQPSK
Error correction coding	CRC, convolutional
Number of carrier frequencies	832
Bit duration	20.57 μ s
Burst duration	6.67 ms
Channel bit rate	48.6 kbit/s
Maximum cell radius	20 km

shown in Figure 8.2. Data for clock synchronization as well as for channel estimation are located at the beginning of the burst. In front of the first half of the user data, control data are inserted. The user data are followed by the colour code, which identifies the base station. Afterwards, the second half of the data, as well as the attached guard bits for the burst's transmission time synchronization, is transmitted.

The VSELP speech encoder provides a data rate of 159 bits every 20 ms. These bits are, as in GSM, arranged into two classes. There are 77 class 1 and 82 class 2 bits. Class 2 bits are transmitted without error protection. The 12 most important class 1 bits (class 1a) are protected by seven bits of a cyclic block code. The corresponding generator polynomial is

$$g(x) = x^7 + x^5 + x^4 + x^2 + x + 1$$

The class 1 bits as well as the check bits are fed into a terminated convolutional encoder of rate 1/2 with generator polynomials

**Figure 8.2** IS-136 downlink burst

$$g_1(x) = x^5 + x^3 + x + 1$$

and

$$g_2(x) = x^5 + x^4 + x^3 + x^2 + 1$$

The 77 class 1 bits, seven check bits, and five tail bits (for the termination of the convolutional encoder) produce 178 code bits at the channel coder's output. Moreover, there are 82 uncoded class 2 bits. Related to the speech frame of 20 ms duration, this leads to a gross data rate of 13 kbit/s.

8.3.3 Third Generation – Universal Mobile Telecommunication System (UMTS)

The universal mobile telecommunication system (UMTS), the European version of IMT-2000, provides two air interfaces, UTRA-FDD and UTRA-TDD. Within the spectral interval allocated to UMTS there are 12 (duplex) bands for UTRA-FDD and seven bands for UTRA-TDD. Two out of the seven UTRA-TDD bands are for self-provided applications (similar to the cordless phone system DECT). The two air interfaces share several similarities: for example, the bandwidth of 5 MHz, the chip rate of 3.840 Mchip/s, the QPSK-modulation, as well as the root raised cosine roll-off impulse shaping filter $g_s(t)$ that employs a roll-off factor of 0.22. On the other hand, the two air interfaces use different access modes for the frequency resource. The TDMA component of UTRA-TDD requires synchronization of the users in the down- as well as in the uplink. Therefore, multiuser detectors [4] that are able to eliminate multiple access interference (MAI) (which occurs because of nonideal cross-correlation properties of the spreading codes) can be implemented in the base stations. Moreover, asymmetric data traffic can flow efficiently with TDD. The UTRA-FDD air interface uses different frequency bands for up- and downlink as well as higher spreading factors. The cell radii of UTRA-FDD are bigger and the users can move at higher velocities compared with UTRA-TDD.

UTRA can usually provide several transport channels per user. These transport channels are separated by time multiplex or physically. A physical channel in UTRA-FDD is defined by a spreading code, in UTRA-TDD by a spreading code and a time slot on a specific frequency band. One particular property of UTRA is that many individual data rates are possible. The different transport channels may employ different data rates and their data may be transmitted with different error protections. Speech coding (adaptive multirate (AMR)) is very flexible and leads to data rates between 12.2 and 4.75 kbit/s. For each transport channel, data can be channel coded, rearranged, and transmitted in sets of transports blocks within transmission time intervals (TTIs) of 10, 20, 40, or 80 ms (Figure 8.3). The channel data are adapted to one of the data rates permitted by flexible channel coding and rate adaptation. First, a systematic block coding for transmission quality control is employed, i.e. to each transport block a number of CRC bits (between 0 and 24) is appended.

For error correction coding we find the following options:

- convolutional coding with a rate of 1/2 or 1/3, constraint length 8, and maximum code block length $Z = 504$ (for example with speech transmission);
- turbo coding with rate 1/3 and maximum code block length $Z = 5114$ for BERs of at most 10^{-6} ; or

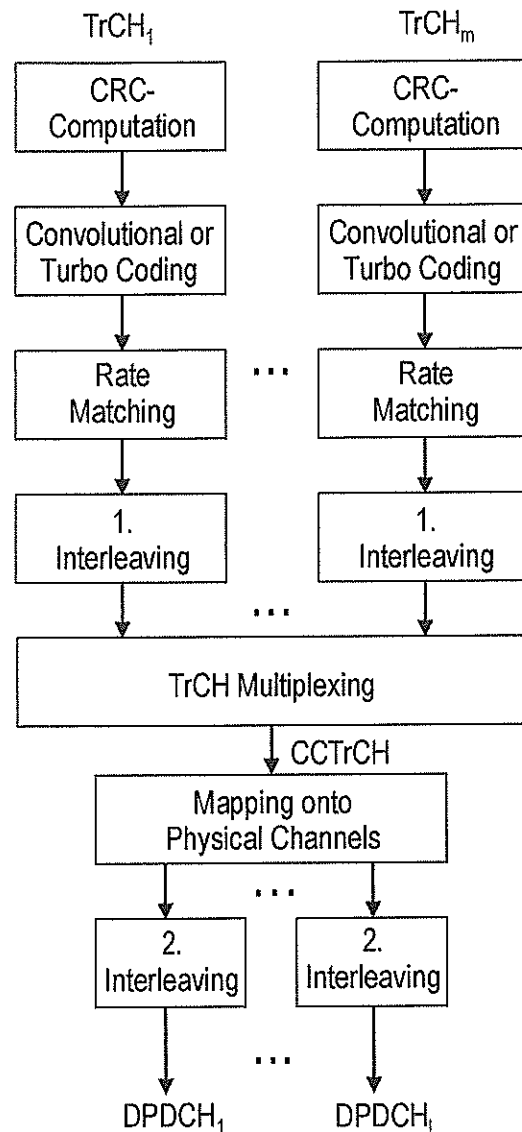


Figure 8.3 UTRA transport channel data

- no channel coding, if the data to be transmitted are already channel coded by the user, or if the transmission conditions are ideal.

The number of bits to be transmitted (user data plus CRC) within a TTI is set to the coding block length N , if it is less than Z ; otherwise, the bits have to be divided into several coding blocks. For termination of the convolutional coding eight zeros are appended at the end of each coding block. Then a variable block interleaving with column re-arrangement over the total TTI duration is performed followed by a rate adaptation which depends heavily on the transmission conditions and upon the cell occupancy. The rate adaptation punctures or repeats specific bits to achieve the nearest data rate of a physical channel. In the UTRA-FDD downlink the rate adaptation is applied first, followed by the first interleaving. Then all transport channels of the user are divided into 10-ms blocks and multiplexed into one single data stream (coded composite transport channel (CCTrCH)). This data stream is then distrib-

uted to one or several physical channels (dedicated physical channels (DPCHs)), again in blocks of 10 ms in length, and afterwards rearranged once more by a second block interleaver with column rearrangement. This means that for UTRA-FDD a CCTrCH of one user is transformed on to physical channels of an equal transmission rate or spreading factor, respectively. For UTRA-TDD, different physical channels (characterized by time slot and spreading factor) can contain different numbers of data bits. With the exception of UTRA-FDD uplink several such CCTrCHs may be built per user and transmitted. The simultaneous transmission of several physical channels, by using different codes, leads to increased amplitude variations of the mobile station's transmission signal in the uplink.

Each physical channel (DPCH) consists of information data (dedicated physical data channel (DPDCH)) and connection specific control data (dedicated physical control channel (DPCCH)). The control data contain:

- pilot sequences (with a length of 2, 4, 8, or 16 bits) used for channel estimation and for the estimation of the carrier to interference ratio C/I for power control;
- feedback information (FBI) bits which are used for transmission of mobile station information to the base station and therefore are sent in the uplink only;
- transmit power control (TPC) bits that carry instructions for power control (2, 4, or 8 bits)
- transport format combination indicator (TFCI) bits that give information about the composition of the data stream (0, 2, or 8 bits).

Pilot sequences are separately sent within each connection. This results in the benefit that the pilots can be employed for channel estimation even if adaptive antennas are used. Moreover, power control in the downlink and phase control in the uplink become possible too. In UTRA the transmit power is controlled very quickly, so that even fast fading can be accommodated. The distribution of control and data bits for the signals I- and Q-components is different for the UTRA-FDD up- and downlink. Therefore, we have to discuss them separately. Pure control channels like the random access channel (RACH) or the broadcast channel (BCH) are not further discussed within this text.

The UTRA-FDD downlink's slot and frame structures are presented in Figure 8.4. A slot is built up of 2560 chips and therefore its duration is 0.667 ms. Within a slot the DPDCH and the DPCCH bits are sent successively. The number of bits transmitted by a slot depends on the spreading factor N_s , which can take values $N_s = 2^k$ with $k \in \{2, 3, \dots, 9\}$. Therefore a slot contains, because of the QPSK modulation, $2 \cdot (2560/4) = 1280$ control and data bits if the spreading factor N_s equals 4. For $N_s = 512$ the number of control and data bits is $2 \cdot (2560/512) = 10$. Because of this concept of variable spreading factors, different data rates can be transmitted within the same bandwidth of 5 MHz. Accordingly, low data rates are transmitted with a high spreading factor due to the advantageous properties of the spreading code's cross correlation functions (CCFs) and autocorrelation functions (ACFs). The signals are then resistant against multipath and MAI. For high data rates, low spreading factors apply and the resistance against multipath and MAI is diminished.

An UTRA frame consists of 15 slots or 38,400 chips and has a duration of 10 ms. A superframe is composed of 72 UTRA frames. If a single user occupies several DPCHs, the corresponding DPCCH bits are transmitted only once and the control data are suppressed in all the other DPCHs.

From Figure 8.5 we see how the modulated and spread transmitter signal $s(t)$ is constructed from the bit stream.

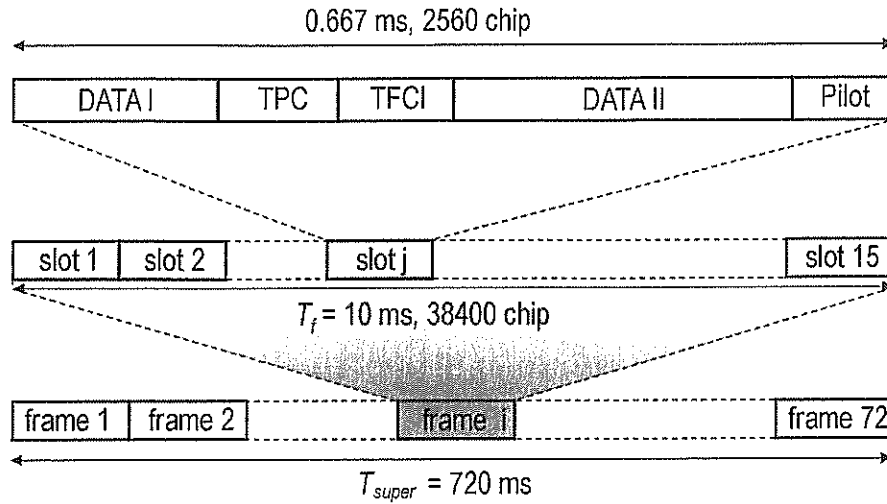


Figure 8.4 The UTRA-FDD downlink structure

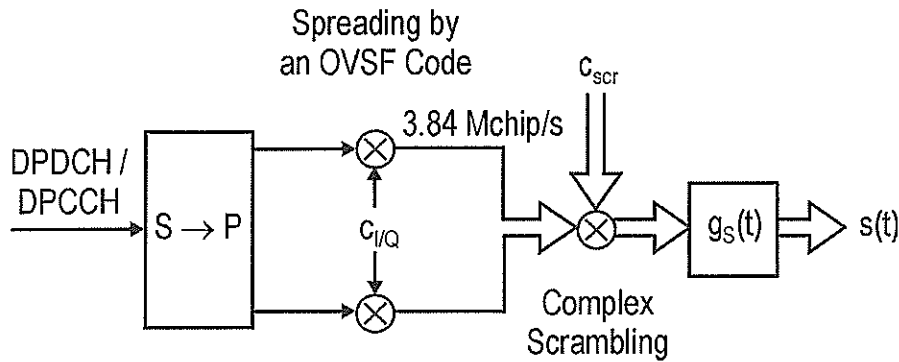


Figure 8.5 Construction of the UTRA-FDD transmission signal

First, the DPCH bits are distributed on to the I- and Q-branches by serial to parallel transformation. Afterwards, the data are spread at the chip rate of 3.840 Mchip/s, applying the same orthogonal variable spreading factor (OVSF) code to the I- and the Q-branch. Finally, the chip sequence is scrambled a second time by a cell specific spreading sequence. This procedure is necessary for cell identification and improves the signal's CCF and ACF properties. The complex scrambling sequences of a length of 38,400 chips are formed from sections of Gold sequences. In UMTS the base stations are not synchronized, but each cell has its own scrambling sequence allocated to it. The cost for the missing base station synchronization is an increased effort at the connection setup, because the cell search can be time consuming for the mobile, since in total 262,144 scrambling sequences, which are divided into 512 groups, can be applied for cell identification. The scrambling sequences are deduced from Gold sequences of length $2^{18} - 1$, whose generator polynomials are given by

$$g_1(x) = 1 + x^7 + x^{18}$$

and

$$g_2(x) = 1 + x^5 + x^7 + x^{10} + x^{18}$$

The generation of the different Gold sequences is controlled by the initial state of the linear feedback shift register defined by $g_1(x)$. The initial state chosen corresponds to the index of the generated Gold sequence and can take values between 0 and 262,143. As indicated in Figure 8.5 the scrambling sequences are complex. If $\{a_n; n = 0, \dots, 2^{18} - 1\}$ is a Gold sequence with index $\leq 262,143$, the corresponding complex scrambling sequence of period 38,400 is given by

$$\{c_{scr,n}\} = \{a_n\} + j\{a_{n+131072}\}$$

$$0 \leq n \leq 38,399$$

On the downlink the same scrambling sequence is applied for all users, only their separation is done by the OVSF codes. It can be shown that the family of OVSF codes can be transformed into the family of Walsh codes by applying bit reversal. All OVSF codes of the same spreading factor N_s are mutually orthogonal. For different spreading codes orthogonality is not always guaranteed. Therefore the selection of the OVSF codes, applied in a single cell, in which user signals of different spreading factors are transmitted, is done by using the code tree shown in Figure 8.6. Only those codes may be chosen which are not located on the same path from the root to the outermost branches of the tree. If the codes are chosen by this procedure, orthogonality is preserved for different spreading factors too.

The UTRA-FDD uplink's slot and frame structures are presented in Figure 8.7. Here the control bits (DPCCH) are transmitted in the Q-branch, while the data (DPDCH) bits are sent in the I-branch (Figure 8.8).

Usually the data rates of the two branches are different in the uplink. To achieve the same chip rate, the spreading factor 256 is always used in the Q-branch, and for the I-branch spreading factors $N_s = 2^k, k \in \{2, 3, \dots, 8\}$ are applied, similarly to the downlink. This means that between 10 and 640 data bits can be transmitted per slot. The assignment of control and data bits to the Q- and I-branches, respectively, was introduced with regard to the increased electromagnetic compatibility (EMC). Since the control data have to be transmitted in every slot, a continuous signal is guaranteed in spite of discontinuous transmission (DTx). Transmission breaks, which would lead to an on and off switching of the power amplifier, are

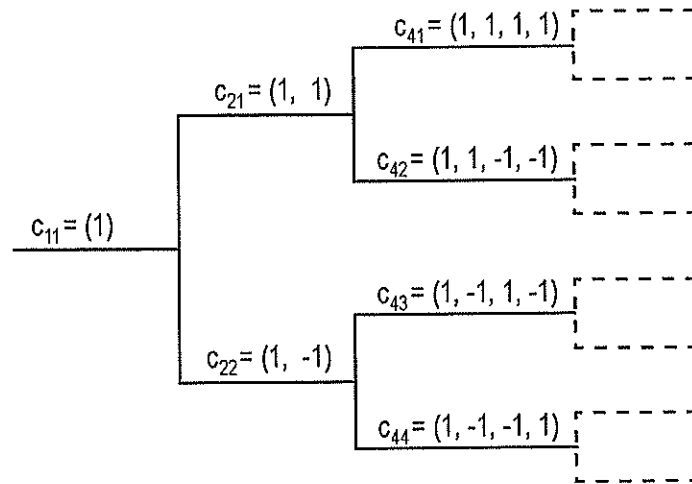


Figure 8.6 The OVSF spreading code tree

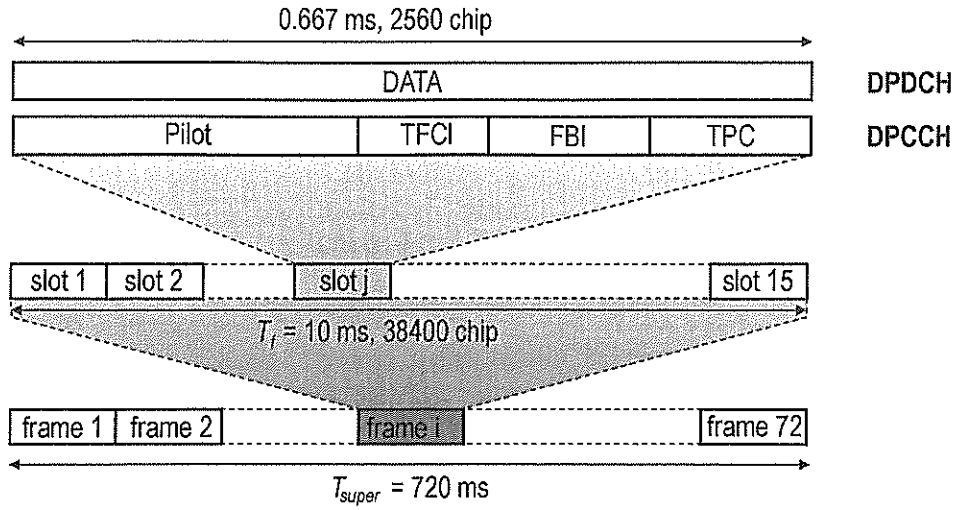


Figure 8.7 The UTRA-FDD uplink structure

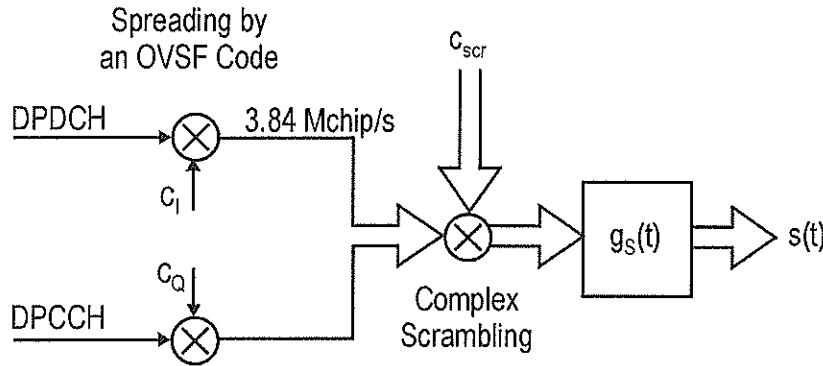


Figure 8.8 The UTRA-FDD uplink I- and Q-branch

avoided. On the downlink, some signals (e.g. the BCCH) are continuously sent. Therefore, no transmission breaks occur in this case.

Since in the uplink the users are not only separated by the OVSF codes but also by the scrambling sequences, the same OVSF code may be assigned to several users within the same cell.

For the uplink the complex scrambling sequences can be chosen from two families: the first family is constructed of quaternary sequences, 256 chips long, that are generated from the following polynomials:

$$g_1(x) = 1 + 2x + x^2 + 3x^3 + x^5 + x^8 \text{ mod } 4$$

$$g_2(x) = 1 + x + x^5 + x^7 + x^8 \text{ mod } 2$$

and

$$g_3(x) = 1 + x^4 + x^5 + x^7 + x^8 \text{ mod } 2$$

Table 8.3 Complex scrambling symbols

z_n	a_n
0	$1 + j$
1	$-1 + j$
2	$-1 - j$
3	$1 - j$

Subsequently, the symbols $\alpha_{n,j}$ ($j = 1, 2, 3$), generated by the three polynomials, are combined:

$$z_n = (\alpha_{n,1} + 2\alpha_{n,2} + 2\alpha_{n,3}) \bmod 4$$

From the last equation a number $z_n \in \{0, 1, 2, 3\}$ results, leading to the complex scrambling symbols a_n , via Table 8.3. These short scrambling sequences are intended to reduce the computational load in the base station if multiuser detection is applied.

The second family consists of sections of length 38,400, out of Gold sequences $\{a_n\}$ of length $2^{25} - 1$ and generator polynomials

$$g_1(x) = 1 + x^3 + x^{25}$$

and

$$g_2(x) = 1 + x + x^2 + x^3 + x^{25}$$

The complex scrambling sequences are constructed from the sequence $\{a_n\}$ by

$$\{c_{\text{scr},n}\} = \{a_n(1 + j(-1)^n a_{2[n/2]+1677232})\}$$

$$0 \leq n \leq N - 1$$

By using this compound procedure for constructing the scrambling sequence for the uplink, variations of the modulus of the signal's complex envelope are avoided. The design of the scrambling sequences guarantees that within N_s chips, i.e. within one code period, maximal one-phase rotation of 180° can occur; furthermore only phase rotations of 90° and maximal one 0° rotation appear. Since every user within the network is assigned its own scrambling sequence, the number of scrambling codes is extremely large; there are 2^{24} long as well as 2^{24} short scrambling sequences. The most important technical data of the UTRA-FDD air interface are summarized in Table 8.4.

8.4 Parametrization Example

In Section 8.3 we described the air interfaces of the standards GSM, IS-136, and UMTS-FDD in some detail. Of course, there are substantial differences between the second-generation (TDMA) standards and the third-generation (CDMA) standard: GSM and IS-136 are on one side while UMTS-FDD is on the other. Here, spreading in the transmitter and despreading in the receiver have to be additionally realized.

Table 8.4 Key air interface parameters of the UTRA-FDD system

Access mode	Direct sequence (DS) CDMA
Duplex mode	FDD
Chip rate	3.840 Mchip/s
Net data rate	8 kbit/s to 2 Mbit/s
Spreading sequences	OVSF codes
Spreading factor	2^k ($k = 2, 3, \dots, 8$); 512 for downlink only
Bandwidth	5 MHz
Channel coding	Convolutional, turbo, CRC codes, interleaving
Modulation mode	QPSK
Chip impulse shaping filter	Root raised cosine, roll-off factor 0.22
Power control	Two closed and one open loop, up- and downlink
Specific properties	Orthogonal spreading sequences for variable spreading factors
Frame duration	10 ms
Slot duration	0.677 ms
Number of slots per frame	15

We will consider the whole signal processing chain, starting with channel coding. There is only a little difference in the coding algorithms within the set of second-generation standards. For speech encoding some combination of block coding for the most important speech bits and convolutional coding for the greater part of the speech bits is applied. For data transmission a stronger convolutional coding is employed. UTRA-FDD as a third-generation air interface offers higher data rates of up to 2 Mbit/s and should guarantee a BER of 10^{-6} for specific applications. Here, turbo codes [5] are implemented which have to be integrated into general encoder and decoder structures. For speech coding convolutional codes are used in UTRA-FDD in general. Since it is not too difficult to find general encoder/decoder structures, we will concentrate here on the more interesting problem of presenting a general modulator structure.

8.4.1 A General Modulator

According to Tables 8.1, 8.2, and 8.4, GSM, IS-136, and UTRA-FDD use GMSK, $\pi/4$ -DQPSK, and QPSK as modulation modes. Additionally, in UTRA-FDD the signal has to be spread to the chip rate of 3.840 Mchip/s. QPSK and its derivative $\pi/4$ -DQPSK are linear modulation modes while GMSK is nonlinear [6]. One object of our investigations was to find a general modulator structure that can process signals of all the standards mentioned above. Therefore we decided to introduce the linearized version of GMSK into an SDR [7,8].

MSK is a two-valued (linear) frequency shift keying (FSK) modulation with modulation index $h = 0.5$. If the rectangular impulse shaping filter of MSK $\text{rect}(\cdot)$ is replaced by a gaussian-shaped impulse $g_{\text{GMSK}}(\cdot)$, a (nonlinear) GMSK modulation results.

The insertion of the gaussian impulse shaper (Figure 8.9) leads to a bandwidth reduction but also leads to a (controlled) inter symbol interference (ISI).

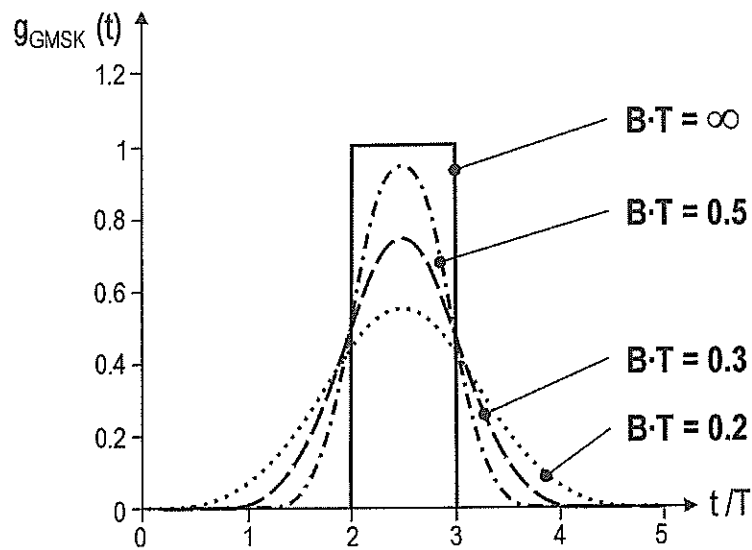


Figure 8.9 Gaussian impulse shaping

The strength of the ISI, or equivalently the bandwidth reduction, depends heavily on the bandwidth time product BT . The GSM standard defines $BT = 0.3$, which leads to a reasonable reduction in bandwidth, but at the same time results in an ISI that has to be equalized in the receiver. But this may be tolerated since the multipath channel also introduces ISI. Figure 8.10 shows the power spectral densities (PSDs) corresponding to different bandwidth time products.

In the complex plane the n th symbol d_n to be transmitted causes a phase variation which has to be added to the phase variations of the previous symbols. For MSK the signal pointer rotates exactly by $\pi/2$ or $-\pi/2$ within a symbol clock (Figure 8.11).

For GMSK the complex envelope can be split up into two summands: one represents the linear, the other represents the nonlinear part of the GMSK signal [9,10]. Moreover, the linear part contains about 99% of the signal's energy. Therefore the GMSK signal may be well

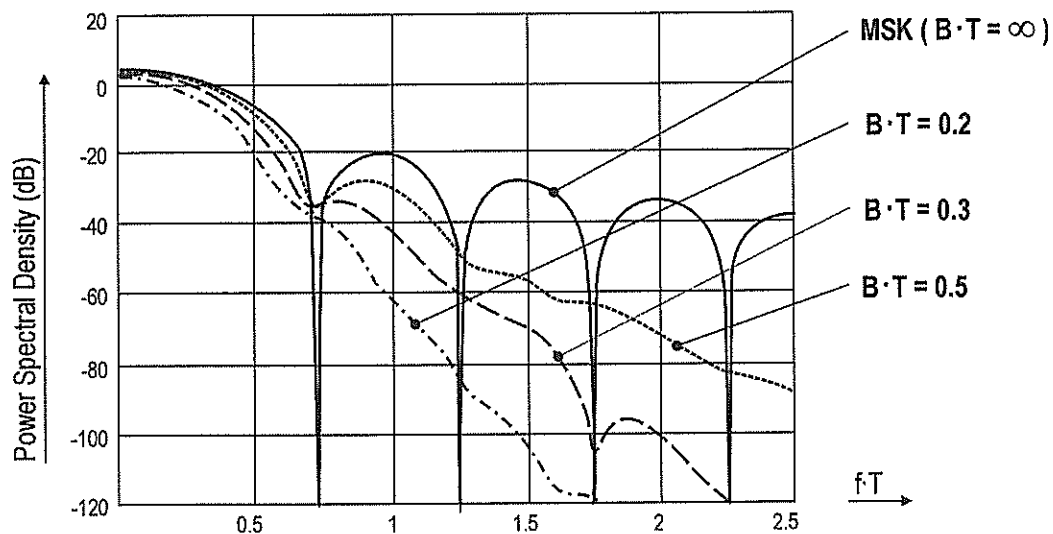


Figure 8.10 PSD variation with bandwidth time product

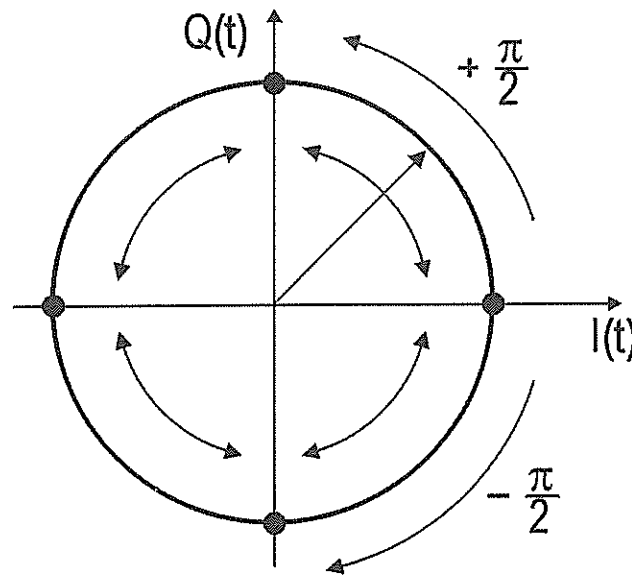


Figure 8.11 MSK phase rotation

approximated by its linear part, i.e. in a software defined radio the symbols to be transmitted by a GMSK signal may be pulse shaped with an impulse $C_0(t)$ that leads to a linear modulation. Using this linear approximation, GMSK signals can be produced by the same linear I/Q modulator employed for PSK signals. This general modulator structure [3] is shown in Figure 8.12. Construction and parametrization of this modulator are discussed in more detail within the following paragraphs.

Figure 8.12 shows a general modulator structure that can produce GSM, IS-136, UTRA-FDD, and even DECT signals. (This structure can be extended to cope with UTRA-TDD and with cdma2000 signals, which are not shown here for reasons of clarity). For UTRA-FDD the modulator shown in Figure 8.12 represents the uplink structure.

We should note here that in GSM a differential precoding is necessary. For the bits $b_k \in \{0, 1\}$ the precoding is defined by

$$\beta_k = b_k + b_{k-1} \bmod 2.$$

The initialization with $b_{-1} = 1$ results from the dummy bits which are sent in GSM before the first user bit during the starting phase of a GSM burst.

The block *Precoder* within the functional diagram of the modulator is switched by the parameters *BurstLength* and *Precoder_On_Off*. The parameter *BurstLength* is used for the initialization, while with *Precoder_On_Off* the precoding may be switched off since it is only required for GSM.

In the next step, the signal is nonreturn to zero (NRZ) coded by the functional block NRZ. For GSM this leads to $d_k = 1 - 2\beta_k$.

The functional block *NRZ* is controlled by the parameter *NRZ_On_Off*. If *NRZ_On_Off* equals one, a bit $\beta = 0$ is transformed into $\beta' = 1$ and $\beta = 1$ into $\beta' = -1$. For *NRZ_On_Off* = -1 the reverse transform results. *NRZ_On_Off* = 0 indicates that no transform is performed.

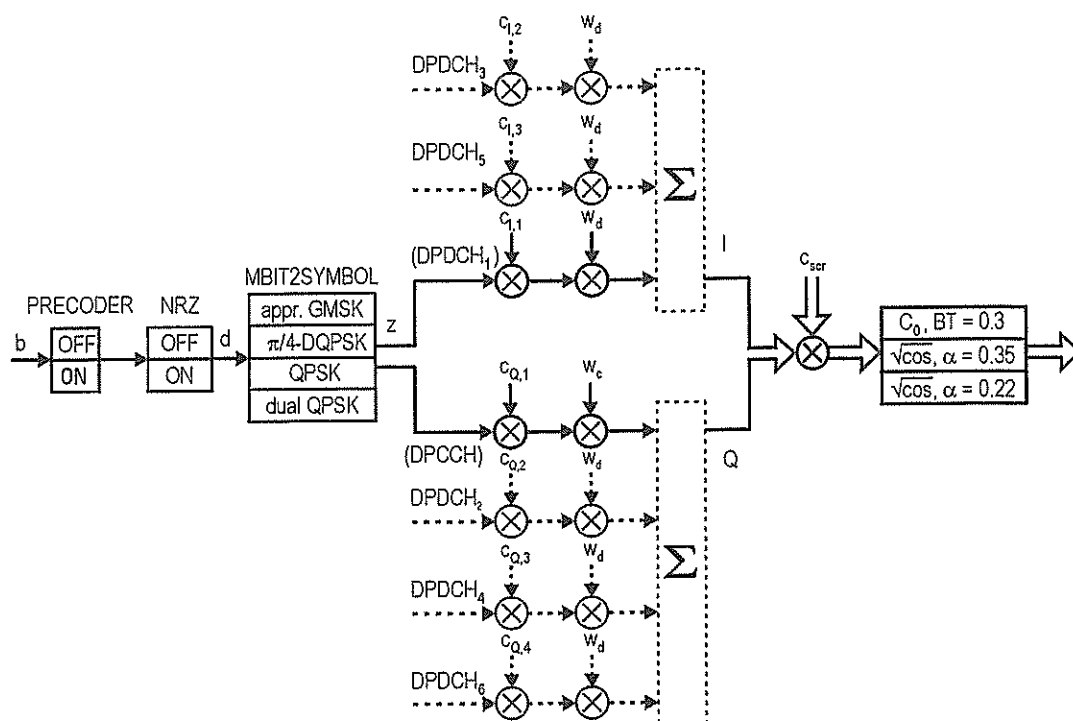


Figure 8.12 Generalized parametrizable modulator

In the next functional block, *MBIT2Symbol*, the bits are mapped to complex symbols z_k . The modulation mode is chosen by the parameter *ModulationNumber*, whose function is specified in Table 8.5.

Since some modulation modes employ memory, an initial state has to be defined. Therefore the parameter *BurstLength* is used in this block too. For UTRA-FDD uplink the control data of the DPDCH are transmitted in the Q-branch and the information data of the first DPDCH are sent in the I-branch of DPDCH₁. This modulation may be interpreted as dual QPSK, where a serial to parallel transform is applied by directing *I_Length* bits to the I-branch and *Q_Length* bits to the Q-branch. The last two parameters are not used for GSM and IS-136. An other option for UTRA-FDD is to switch more DPDCHs onto the I- and Q-branches. These channels are indicated in Figure 8.12 by dotted blocks, since this option can only be chosen with a spreading factor $N_s = 4$.

For UTRA-FDD, spreading is performed with the next processing step. This function is controlled by the parameter *Spreadingfactor*. The spreading sequences are stored and can be

Table 8.5 *MBIT2Symbol* parameters

Modulation mode	<i>ModulationNumber</i>
GMSK	1
$\pi/4$ -DQPSK	2
QPSK	3
Dual QPSK	4

selected by the sequence number. For $Spreadingfactor = 1$ no spreading takes place. The following weighting of the data and control channels with the weighting factors w_d and w_c , as well as the complex scrambling, is again performed only for UTRA-FDD.

The next step is the complex impulse shaping, for which different finite impulse response (FIR) filters can be employed by the parameter *Filter_Number* (Table 8.6). For GMSK the (linear) impulse $C_0(t)$ is used; for PSK modulations root raised cosine roll off filters are applied.

Table 8.7 presents the parametrization of the general modulator required for GSM, IS-136, and UTRA-FDD. For UTRA-FDD the *BurstLength* depends on the spreading factor. In Table 8.7 we used the example $Spreadingfactor_I = 8$. For the DPCCH, on the other hand, $Spreadingfactor_Q = 256$ is always applied.

If, for example, UTRA-TDD should be integrated into this modulator structure, an additional modulation mode has to be realized in the block *MBIT2Symbol* (Table 8.5). Moreover, other scrambling sequences have to be used and finally the complex midamble has to be inserted.

8.4.2 Effects of GMSK Linearization

The linear approximation of GMSK causes a nonconstant amplitude of the transmitted signal. Figure 8.13 shows the effect of GMSK linearization to the amplitude sketched in the complex signal plane for the bandwidth time products $BT = 0.5$ (DECT) and $BT = 0.3$ (GSM).

It is evident that the effect of linearization is stronger for $BT = 0.3$. This can be explained by the fact that for $BT = 0.5$ the GMSK modulation itself is ‘almost’ linear and the effect

Table 8.6 Pulse shaping filters

Filter	<i>Filter_Number</i>
Main impulse $C_0(t)$ of linearized GMSK with $BT = 0.3$	1
Root raised cosine, roll-off factor $\alpha = 0.35$	2
Root raised cosine, roll-off factor $\alpha = 0.22$	3

Table 8.7 General modulator, parametrization

Parameter	GSM	IS-136	UTRA-FDD
<i>BurstLength</i>	148	312	330
<i>Precoder_On_Off</i>	1	0	0
<i>NRZ_On_Off</i>	1	1	-1
<i>ModulationNumber</i>	1	2	4
<i>Spreadingfactor_I</i>	1	1	8
<i>Spreadingfactor_Q</i>	1	1	256
<i>Filter_Number</i>	1	2	3
<i>I_Length</i>	–	–	320
<i>Q_length</i>	–	–	10

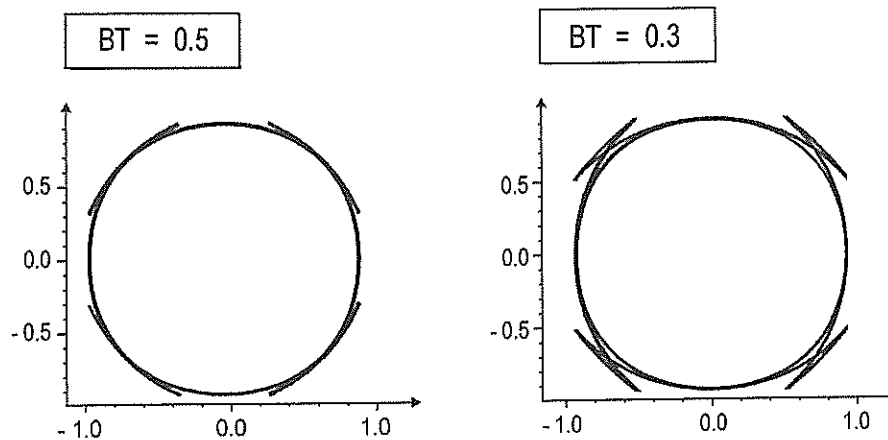


Figure 8.13 Effects of GMSK linearization

remains minimal in this case. On the other hand, it is the constant envelope that makes GMSK very attractive for mobile radio, since power efficient C-class amplifiers can be employed in this case without producing severe intermodulation products.

We can now consider the effects of GMSK linearization if the corresponding signal is power amplified. For our simulations we used a power amplifier model described in [11] that well approximates the measured properties of a real Gallium Arsenide field effect transistor (GaAs FET). Figure 8.14 shows a characteristic AM/AM amplifier of this model.

In Figure 8.15 the PSDs of exact and linearized modulated GMSK signals are compared. Without nonlinear amplification the PSD of the linearized GMSK signal meets the spectrum mask given by the GSM standard better than the PSD of the exact GSM signal. If power

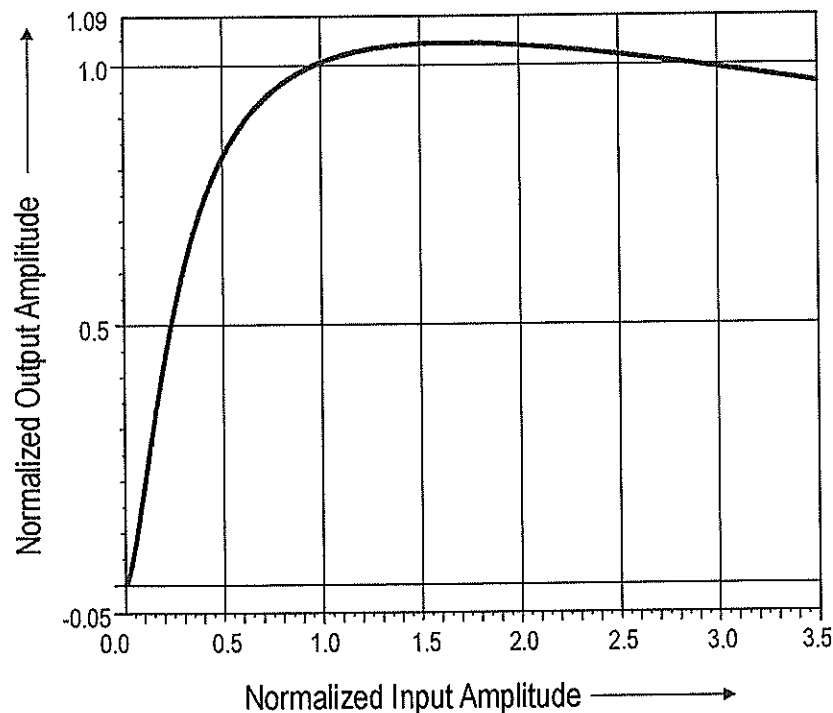


Figure 8.14 Characteristic PA amplifier

amplification is applied, the PSD of the linearized GMSK signal closely approximates the GSM spectrum mask in the region around ± 350 kHz distance from the carrier frequency. But the GSM standard is still met. Since within an SDR the final power amplifier has to meet the advanced demands of $\pi/4$ -DQPSK and QPSK modulations, the linear GMSK approximation does not represent a loss in efficiency. For linear power amplifier characteristics we expect that the PSD of the linearized GMSK signal will meet the GSM standard's requirements more closely than for the example amplifier used in our simulations.

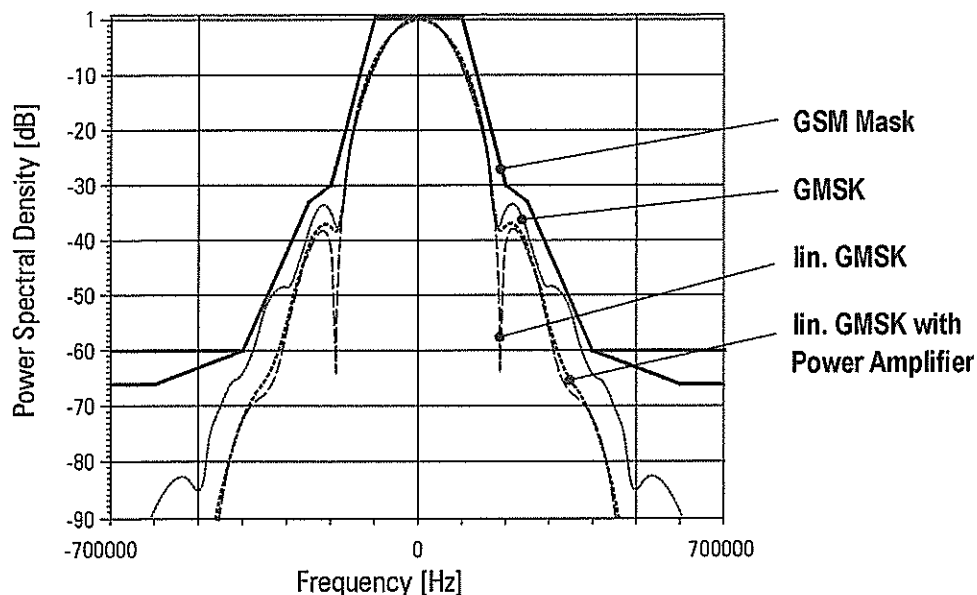


Figure 8.15 Comparison of the PSDs for the original and linearized GMSK

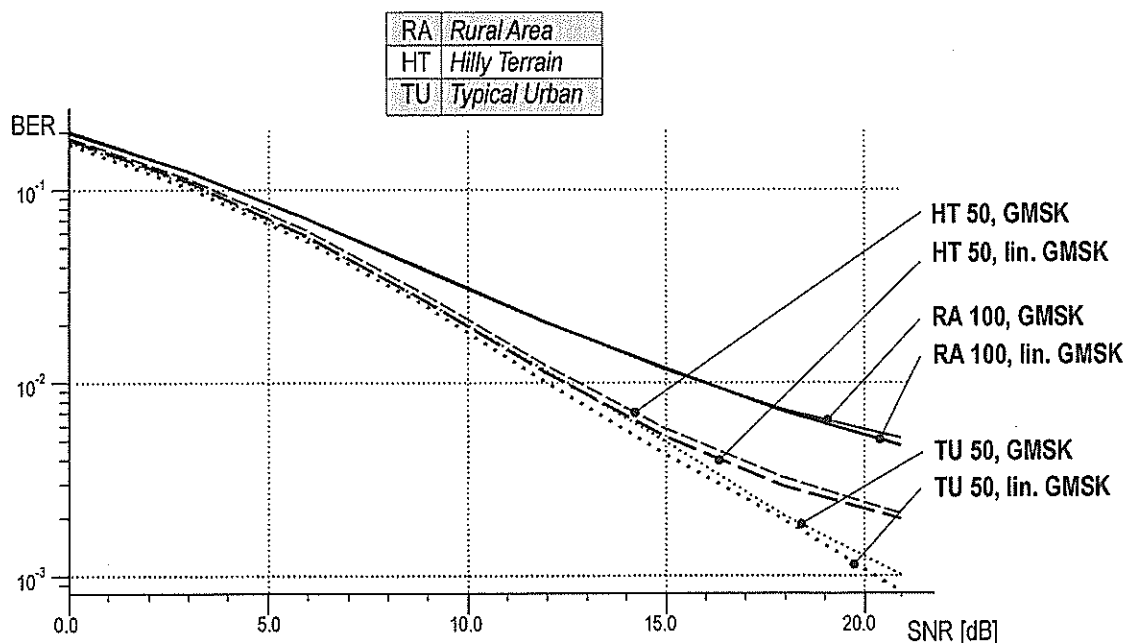


Figure 8.16 Effect of linearized GMSK on BER

We now consider the effect which GMSK linearization has on the bit error rate (BER). The BER heavily depends on the receiver algorithms as well as on the mobile radio channel. For GSM this channel usually is frequency selective. The GSM channel models acquired by COST 207 suppose a maximum multipath delay of 20 μ s, corresponding to an ISI of six symbols. Since the equalizer and the channel estimation algorithms usually applied in the receiver are based on the model of a linearly modulated signal, the application of the linearized GMSK seems to be advantageous with respect to the bit error rate. In this consideration the nonlinear part of the GSM signal is looked upon as an interference term. The results of this discussion can be deduced from the simulation results of Figure 8.16.

The simulations were performed using the COST 207 channel models. The differences between exact and linearized GMSK almost vanish and cannot be detected if channel coding is applied, since the energy of the nonlinear part of the GMSK signal is very small compared with the rest. However, it is very important that the linearization of the GMSK signal does not cause performance degradations.

8.5 Signal Processing Issues

Within this section we consider hardware implementation aspects of SDR. The goal here is not a detailed discussion; our objective is to consider the applicability of programmable signal processing components to software defined radios. A wider discussion of hardware signal processing technology may be found in Chapter 7 by Lund and Honary.

8.5.1 DSP Capabilities and Limitations

We shall restrict ourselves to a discussion of the software defined radio receiver, since the computational power of a receiver is always larger than that of a transmitter. From the signal processing viewpoint it is helpful to discriminate between base station receivers, which have to process several signals simultaneously, and mobile terminal receivers. For a base station receiver it may be advantageous to digitize the whole system bandwidth at the same time and to perform the channel selection by digital means (e.g. by a fast Fourier transform (FFT) receiver). For present purposes, we focus on mobile terminals. In our present example the total processing requirement of the receiver is determined by UTRA and its channel bandwidth of 5 MHz.

The baseband signal processing of a software defined radio has to perform the following tasks:

- estimation of the channel impulse response
- synchronization
- specific receiver algorithms like RAKE or multiuser detection
- equalization
- demodulation
- burst decomposition
- channel decoding
- source decoding

It makes no sense to concentrate on the efficiency of specific signal processors here, since within a terminal these devices are integrated into a system environment. Here, we discuss

some properties that seem to be advantageous in connection with SDRs. This is especially important with respect to future developments that will require sampling rates of more than 100 Msamples/s, which will have to be processed.

Software defined radios require parallel as well as sequential partitioning of algorithms such that the necessary computational power can be made available. For this task specific DSP solutions are on the market. One setup combines a central processing unit (CPU) and a DSP on the same chip (e.g. Motorola 68356). A different solution is provided by DSPs with integrated Viterbi processors that can be employed for all applications that need an equalizer or a convolutional decoder. Of course, sufficiently large memories must be integrated on the DSP chip to guarantee short access times.

8.5.2 FPGA Capabilities

It is not possible to produce mobile communication terminals or software defined radios solely using DSPs since they are optimized for sequential programs and only offer restricted support of parallel processing. In many realizations specific tasks are performed by application specific integrated circuits (ASICs). However, ASICs employ hard-wired logic which does not possess the flexibility necessary for software defined radio implementations.

An FPGA is an array with programmable connections and logic functions, which can be redefined, even after production. There are two kinds of programmable arrays, programmable logic devices (PLDs) and FPGAs. Frequently, PLDs implement sums of products of input signals. Unfortunately all signals then have to be represented in this form (sums of products), leading to a waste of chip area. On the other hand, FPGAs are built of an array of blocks that are built of logic blocks with corresponding connections.

Normally, the area occupied by the logic blocks of FPGAs is smaller than that engaged by PLDs, i.e. the chip area efficiency of FPGAs is higher. A logic block of an FPGA usually contains look up tables with several data ports and flip-flops for data storage.

For circuit development, hardware description languages like VHDL or Verilog are presently used. Therefore it is not necessary to worry about internal structures, even though the processor architectures may be rather complex. Software tools automatically perform mapping, placing, and routing of electronic circuits on an FPGA. However today, to achieve the highest efficiency, hard-wiring is also still required.

Configuration of an FPGA typically happens when the system is switched on. During operation the configuration usually remains fixed. Advanced FPGAs can be reconfigured dynamically while operating; this is necessary, for example, if a system built with FPGAs is used alternately as a transmit and receive module.

8.6 Conclusions

In this chapter we have presented the technique of *parametrization of the baseband signal processing for a software defined radio*. After a short introduction, the terms ‘digital radio’, ‘software radio’, ‘software defined radio’, and ‘cognitive radio’ were discussed. We then explored the meaning of adaptability within a software defined radio and explained the rationale for parametrization of standards. In the next section we presented the most important air interface properties of the standards GSM, IS-136, and UTRA-FDD in some detail as a basis for the development of a general modulator structure that can be used to realize the

different modulation schemes of these standards in a parametrized SDR implementation. We concluded with a brief consideration of the hardware implementation requirements of SDR.

Parametrization as an SDR technique represents more than simply programming the base-band signal processor with the appropriate parameters – it represents an important first step towards the development of new design tools and methodologies for SDR, needed to allow SDR to transition from its current status of emergent to mainstream technology.

References

- [1] Mitola, J., 'Cognitive Radio, An integrated agent architecture for software defined radio', PhD Dissertation, Royal Institute of Technology, Stockholm, Sweden, Department of Teleinformatics, 2000.
- [2] Siebert, M., 'Design of a generic protocol stack for an adaptive terminal', *Proceedings of the 1. Karlsruhe Workshop on Software Radios*, 2000, pp. 31–34.
- [3] Wiesler, A., 'Parametergesteuertes Software Radio für Mobilfunksysteme', PhD Dissertation, Universität Karlsruhe, Institut für Nachrichtentechnik, 2001.
- [4] Verdu, S., *Multisuser Detection*, Cambridge University Press, Cambridge, 1998.
- [5] Heegard, C. and Wicker, S., *Turbo Coding*, Kluwer, Boston, MA, 1999.
- [6] Proakis, J., *Digital Communications*, 4th Edition, McGraw-Hill, New York, 2000.
- [7] Wiesler, A. and Jondral, F., 'Software radio structure for second generation mobile communication systems', *Proceedings of the IEEE Vehicular Technology Conference*, Vol. 48, May, 1998, pp. 2363–2367.
- [8] Wiesler, A., Machauer, R. and Jondral, F., 'Comparison of GMSK and linear approximation of GMSK for use in software radio', *Proceedings of the 5th IEEE International Symposium on Spread Spectrum Techniques & Applications ISSSTA '98*, Vol. 2/3, September, 1998, pp. 557–560.
- [9] Laurent, P.A., 'Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulses (AMP)', *IEEE Transactions in Communications*, Vol. 34, 1986, pp. 150–160.
- [10] Jung, P., 'Laurents' representation of binary digital continuous phase modulated signals with modulation index 1/2 revisited', *IEEE Transactions in Communications*, Vol. 42, 1994, pp. 221–224.
- [11] Ghorbani, A. and Sheikhan, M., 'The effect of solid state power amplifiers nonlinearities on MPSK and M-QAM signal transmission', *Proceedings of the IEE Sixth International Conference on Digital Processing of Signals in Communications*, 1991, pp. 193–197.

12

Protocols and Network Aspects of SDR

Klaus Moessner

University of Surrey and Virtual Centre of Excellence in Mobile and Personal Communications

Visions for systems beyond 3G foresee software defined radio as one of the core enablers of seamless service provision across the boundaries of different wireless transmission platforms. Such visions (e.g. [1]), assume the use of different types of network [2] ranging from fixed (e.g. xDSL) connections via short range wireless (PAN/VAN/HAN) schemes, wireless LANs and cellular radio networks to broadcast systems and beyond. Accessing such a variety of different networks will require highly flexible mobile terminals – a capability expected to be provided by software defined radios. However, not only do the air interface transmission schemes of the aforementioned network access technologies differ, but so do the protocol stacks deployed in each of these access networks.

The visions of future communication systems also foresee interconnection of these different access networks using IP-based core networks, the concept being to support every variety of network–air interface with a common transport, signaling, and service provision platform. Apart from connectivity via such an IP-based core network, the single access networks will need to provide a support infrastructure for the reconfiguration of software defined terminals. Reference [3] outlines three major requirements to be provided through such ‘access support and core-network’ integration:

- connectivity between access and core networks
- internetworking between different access schemes (i.e. inter- and intrasystem handoffs, QoS negotiation, security, and mobility support)
- the ability to interface with new and existing radio interfaces.

The core and access networks in such a future integrated environment also will need a means for secure and trustworthy provision of reconfiguration software.

Reconfigurability of software defined radios will be one of the main enablers of such future visions. However, an examination of the issues mentioned above quickly leads to the conclusion that seamless service provision based on software defined radio will require more than

simply software defined terminals – it will require reconfigurability of the networking software (i.e. protocols and protocol stacks) and new terminal reconfigurability support capabilities embedded within the network infrastructure. These are the issues addressed by this chapter.

The first section of the chapter briefly introduces the idea of reconfigurable protocol stacks, contrasting them with the rigid, currently used, service access point (SAP) scheme. The second part reviews the basic concepts of protocols and protocol stacks in general, followed by an introduction to composable and adaptable protocols. The third part outlines possible support networks that are designed or proposed to support reconfiguration software download and reconfiguration of software defined radios. A description of, and rationale for, distributed reconfiguration management and control are contained within the fourth section of the chapter. The final section summarizes the network requirements for effective support of software radio technology.

12.1 Protocol Stacks: SAPs vs. Reconfigurability

Protocol stacks are mere aggregations of several single protocols, whereby each of the protocols within a stack implements certain functionality and serves a distinct and clearly specified task. Protocol frameworks in general (i.e. protocol stacks) use stratification (layering) as their composition mechanism. One advantage of this approach is that the functionality within one layer is impervious to the properties of the layers below or above. Each protocol layer can thereby be treated as a ‘black box’ and no mechanism exists to identify or bypass any functional redundancies which may occur if protocols within a stack have been implemented by different vendors. In general, functions of protocols or the protocol services within the various layers are *offered to* the next upward layer or *expected from* the layer immediately below, implying that each layer acts as both service provider and service requester [4]. This stratified principle for service access and service provision is used in most mobile and fixed line telecommunication networks. Gateways are therefore defined to act as bridges between the different protocol stacks to enable interworking between different networks.

Communication between the different layers of a single protocol stack is accomplished via so-called ‘service access points’ (SAPs). These SAPs provide access via sets of primitives that enable access to a given layer by its neighboring layer (i.e. the next layer up or down in the hierarchy). Within the remainder of this section, these service access points are compared with open platform implementations (i.e. based on application programming interfaces (APIs) rather than SAPs); the different philosophies of these approaches are also described and compared within the context of software defined terminals and communication nodes.

12.1.1 Service Provision via Service Access Points

Protocol services are sets of operations that are performed within a particular protocol layer; layers above (i.e. the neighboring layer one level up in the protocol stack hierarchy) can request layer inherent protocol services via SAPs. The protocol service descriptions define only the types of operation that a particular protocol layer may offer to the next layer up, as shown in our model (see Figure 12.1), where layer 2 (L2) offers link related services to the upper neighboring layer (L3), via the inherent SAP.

SAPs not only provide the services for their corresponding protocol layer, but they are also

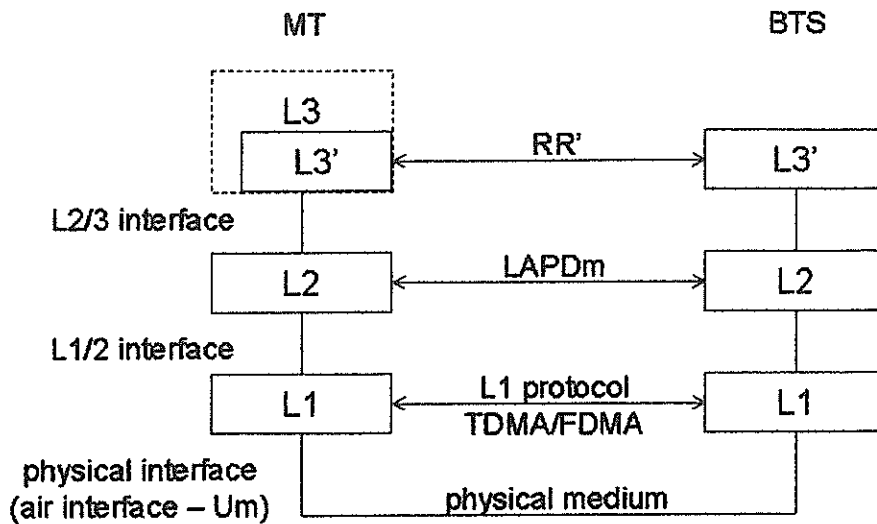


Figure 12.1 Layers and protocols

used to capture the complexity of the layers they represent. SAPs hide the complexity of the layer implementation and describe, uniquely, the interface for the functionality that a particular layer provides, i.e. SAPs identify which functions an upper protocol layer may request from the current layer. Introduction of SAPs and their standardization was a big step towards open platforms; this is because such standardization has allowed proprietary implementations of protocol functionality to be hidden, enabling protocol implementations from different vendors to be used together in one protocol stack implementation.

One of the major drawbacks of SAPs, however, is their lack of flexibility, i.e. they do not support changes, alterations, or improvements within the protocol stack. If even only minor changes in terms of functionality or access to this functionality become necessary, the SAPs and protocols must be restandardized. An example of such restandardization is the gradual evolution of the GSM standard and, finally, the step (incorporating a completely new protocol stack design) towards the provision of generalized packet radio services (GPRS).

Within a protocol stack, each of the SAPs can be identified by a unique address (i.e. a port). Each of the incorporated protocol layers possesses its own SAP; this means that, for example, a transport layer offers its services via a 'transport layer SAP', while the session and presentation layers use 'session layer SAP' and 'presentation layer SAP', respectively.

12.1.2 Protocol Configuration and Reconfiguration

Adaptive, composable, and reconfigurable protocols are introduced and briefly described within a later section. Each of these three technologies offers the ability to customize protocols and protocol stacks for the system requirements of the current settings of a software definable terminal; this ability should be regarded as one of the main features needed to support real software reconfiguration of software definable terminals. However, while the adaptive and composable approaches offer their services to applications via additional adaptation layers only, the reconfigurable approach is based on programming interfaces without such additional adaptation layers.

The main features of programming interfaces are not completely dissimilar to those offered

by SAPs; this means they also offer the services of underlying implementations via clearly defined interfaces, whereby the actual implementation of functions is hidden beneath these APIs. However, using programmable interfaces between protocols offers the complete range of flexibility and features to software developers, similar to that offered by commercial application software development packages. Applying these open programming platform principles to protocol stacks introduces not only increased programmability but also facilitates putting object oriented development features into protocol stack design – thereby simplifying design.

12.1.3 Interfaces vs SAPs

At first sight one might conclude that service provision via SAPs and via programming interfaces are not dissimilar; however, there are a number of differences between the two techniques that must be understood. While SAPs are defined as ports, for accessing the primitives defined within a layer, programming interfaces are defined in interface classes that are linked to the actual implementations of the layers. Interfaces, i.e. their definitions and implementation classes, can be organized hierarchically and their general structure also offers the possibility of applying object oriented programming techniques and methods [5]. This principle enables both the extension and inheritance of interface definitions. Figure 12.2 shows the principle: the left column depicts the definition of a generic interface (called 'layer_n'), its extension in 'layer_nx', and finally its implementation in 'prot_n'. The right column shows the same generic interface; however, this time 'layer_nx' inherits its functionality from 'layer_n', complements it, overrides some of the definitions from the original interface, and, finally, the interface is again implemented in 'prot_n'. Either of the techniques may be applied to facilitate the use of generic interfaces and to enable customization and extensibility of interface definitions.

Use of such a concept makes interfaces extensible in several ways; this includes both specialization by inheritance or by addition of a simple extension to the original interface

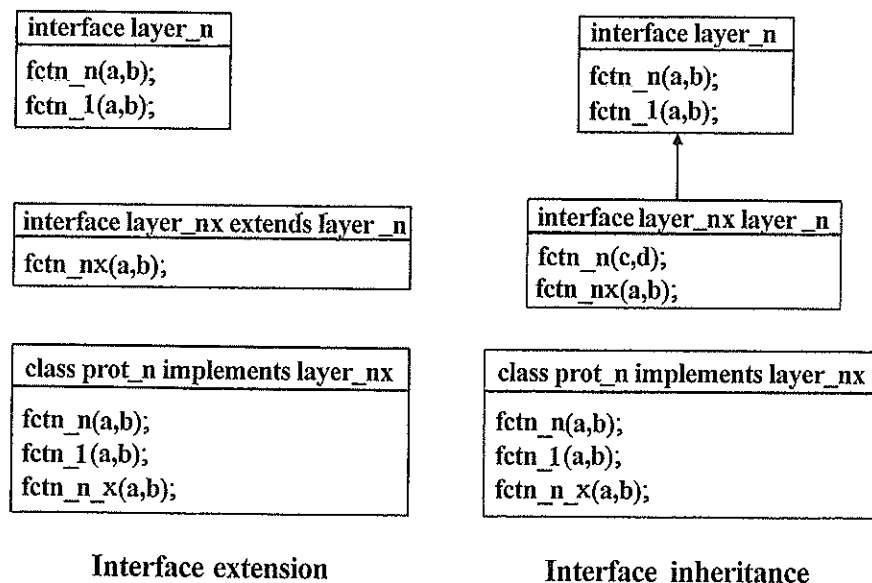


Figure 12.2 Interface extension and inheritance

definition. This latter feature also enables the compatibility between different versions of the same interface, i.e. the functionality will still be given if a protocol implementation that had been extended with some additional feature has to be accessed by a neighboring protocol layer (i.e. using the original interface implementation).

Apart from this ease in the redefinition of protocols and their interfaces, the opening up of application or protocol programming interfaces has the potential to trigger an effect similar to the introduction of application development for personal computers, whereby the provision of open programmable interfaces (and later of complete development environments) triggered the boom of a whole industry. A similar effect may be anticipated for application specific protocol programming in the networking and wireless networking arenas.

12.2 Approaches to Protocol Stack Reconfiguration

Protocols and protocol stacks of a variety of types lie at the core of all data and communication networks. They are the basis not only for the open systems that enable the Internet but also for public switched telephone networks, cellular communication systems, and other networked data/information exchange platforms. It can be regarded as almost certain that most of these (both public and privately owned) networks will become interconnected and will have to be, if possible seamlessly, interworked in the coming years.

Multimedia communication and transmission of data will increasingly take place across the boundaries of different wired and wireless networks using different transmission protocols – indeed this is already happening to a degree. There are several possible ways to adapt between the various different protocols, ranging from gateways that strip off the protocol specific parts of data packets (i.e. the headers) and ‘repacketize’ the information (i.e. the content/data) using different protocols, to adaptive and composable protocols, and, finally, to the use of customizable network nodes capable of interpreting the protocol information attached to a packet as it arrives.

The first part of the next section describes the basics of protocols and protocol stacks; this is followed by a description of composable and adaptive protocols, and, finally, by a description of active networks and other solutions that support the use of multiple protocols within networks.

12.2.1 *Protocols and Protocol Stacks*

The core of any communication system, independent of the transmitted traffic type, is a set of agreements about how a communication has to be made that ensures that all the communication partners will be able to understand and correctly interpret the contents transferred. Protocols are the actual implementations of these agreements, and the sending and the receiving nodes, at least, have to apply the same protocol, or set of protocols, to establish a sensible communication between them.

The complexity of networks (as described in [4]) and network interconnection, however, has led to the implementation of stratified protocol structures whereby different layers offer their particular services to the next higher layer. This ‘protocol stack’ model not only reduced the complexity of protocol implementations but it also introduced different levels of abstraction. Each of these layers within a protocol stack ‘communicates’ with a peer (i.e. the same layer of the stack within a remote partner or host) according to the rules and conventions

defined in the protocol for this particular layer. Figure 12.1 illustrates this approach for a three-layer network (i.e. similar to the GSM protocol architecture at the Um interface), whereby each of the hosts depicted contains the same layers. The congruent layers are the two peers between which a protocol is valid (i.e. within the GSM stack, LAPDm is the protocol used for the communication between L2 of hosts a and b, respectively).

The rationale to separate protocols into several distinct layers is based on a number of principles:

- a layer should be introduced where an additional layer of abstraction is required;
- each layer within a protocol stack should perform a clearly defined function;
- functionality of layers should be defined as generally as possible (to enable open system development);
- layers should be defined to provide minimized interfaces to neighboring layers (i.e. encapsulation of complexity);
- the number of layers should reflect the number of identified protocol functions, but should not blur the clarity of the protocol stack architecture.

These principles are the basis of the open systems initiative(OSI) reference model for protocol stacks and have been applied for most of the commonly used protocol stacks (e.g. GSM).

Open system protocols are rigidly defined and standardized, with the result that they lack any reconfigurability or capacity for adapting to changing communication requirements. More versatile protocol stack constructs are thus required to enable and simplify terminal reconfigurability and to meet the needs of integrated communication environments. Possible approaches for this are described in the subsequent parts of this section.

12.2.2 Modular Approaches: Adaptive, Composable, and Reconfigurable Protocols

As described, the OSI reference model modular principle for protocol stack implementation does not suffice to meet the requirements of reconfigurable data and communication platforms. Protocol stack implementations for such platforms require versatility to such an extent that a software (definable) radio can assume not only *many* but *any* possible protocol configuration.

Adaptable and composable protocols are techniques that can provide such versatility. The basis of these two technologies is the fact that protocols, or rather the layers within the protocol stacks, are mere agglomerations of numerous single protocol functions, which also may be implemented independently from their assignment to a layer. The OSI reference model merely suggests which protocol function should be implemented within which layer; it does not mandate further implementation details.

Adaptive and composable protocols are based on the principle of basic protocol functions being combined in a generic protocol stack (together with the adaptation element necessary), or assembled during boot time, respectively. Using the OSI model as a rough guide for the classification of the functionalities to be implemented within such a pool of protocol functions, the following allocation of functions may be applied, whereby physical layer and application/presentation layer functionality are not regarded as part of a pool of generic functions because of their strong hardware platform and operation system dependencies, respectively. The functions of the other protocol layers are distributed according to the particular task of each layer.

- *Link layer*: protocol functions include framing, sequencing of frames, traffic regulation, error control/handling, and for broad- and multi-cast channel access control (implemented in the media access control (MAC) of the link layer).
- The *network layer* should implement functions that control the operations of the subnet, that manage the various routing issues (i.e. how to route packets from source to destination), and deal with broadcast and multicast transmissions. Furthermore, it should implement congestion control and deliver the support for accounting.
- Functions of the *transport layer* include assembly and disassembly of data units, isolation of the higher layers from changes within the hardware technology, and the creation of distinct network connections for each transport connection (clearly this latter point is not applicable for connectionless transport protocols).
- Finally the *session layer* should implement functions that allow the establishment of 'sessions' between different machines. It should provide enhanced (application dependent) services like management of dialog control, token management, and synchronization of data streams to cope with unexpected connection aborts.

Most of these functions are applicable for any protocol stack, with differences occurring mostly for single parameters such as timer values, or through the use of different algorithms, etc. Furthermore, for some protocol stacks (e.g. the GSM stack) it may only be necessary to have parts of these generic functions implemented (i.e. GSM does not support broad- or multicast, etc.). Nevertheless, identification of generic protocol functionalities is the core of the following protocol frameworks.

12.2.2.1 Adaptive Protocols

Adaptive protocols consist, in principle, of a generic protocol layer that implements a set of common protocol functions; and of a second part that implements a customized extension (to this 'generic, or common, protocol') and provides the protocol functions required to bridge the differences between the generic and standardized protocols. Figure 12.3 depicts the principle of these adaptive protocols, using GSM, DECT, and UMTS stacks as examples. The practical implementation of this scheme results in a protocol stack framework consisting of all the common elements of those protocol stacks that may be implemented within such a framework. The generic part of this adaptive structure contains merely the common elements of the various protocols (i.e. combined and implemented in the generic stack) and can adapt to the target protocol stack by extending this generic part by means of the the protocol functions that are specific for the intended target protocol.

The process for building such an adaptive framework starts with the determination of the common elements needed within the protocols that are to be included in the framework; this is followed by their implementation in the generic part of the structure. Meanwhile, the noncommon, or protocol specific, elements have to be combined and implemented in different, protocol specific, extensions. These extensions will eventually complement the generic/common implementations.

Reference [6] proposes such an adaptable and generic protocol stack, whereby the complete structure is seen as an inheritance tree, starting from the most generic system description (i.e. the notion of an OSI protocol stack) which becomes more specialized to

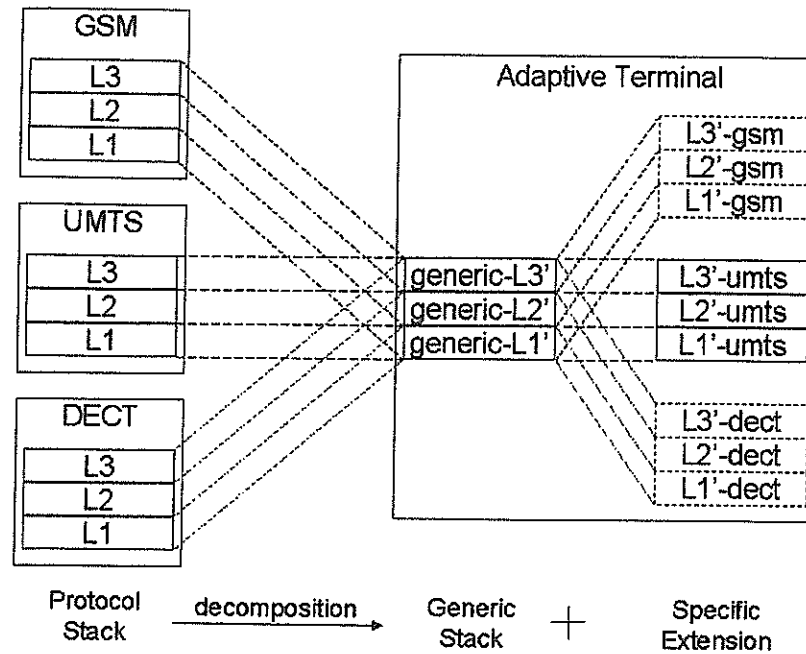


Figure 12.3 Adaptive protocol stacks

implement the 'generic' stack that, in turn, becomes extended to implement the target protocol stack.

Adaptive protocol stacks are indeed a viable approach to generic protocol stack definition and for tackling configurability in software (defined) radio systems. The major shortcoming appears when very different or diverse protocol stacks are to be implemented within one adaptive protocol framework. In such a situation the generic parts of the framework are bound to shrink, and the different extensions required become too extensive to provide a real advantage over discrete protocol stack implementations.

12.2.2.2 Composable Protocols

Composable protocols represent another alternative approach. The functionality of protocols and complete protocol stacks can be split into single protocol functions, and a pool of these functions may then be used to construct customized protocol stacks during boot time. Various projects have been initiated to explore and implement this principle of composable protocols, one of them being 'DaCaPo'.

DaCaPo is a public domain framework that implements protocol configuration during runtime (boot time) rather than during compilation. The intention of this framework is to create customized protocols that provide the QoS parameters necessary for the current/intended connection. The basis of the architecture is a stack structure with a reduced number of layers (when compared to the OSI 7 layer model): only three layers are defined within the DaCaPo framework [7]:

- layer A – the application layer
- layer C – communication support layer
- layer T – the transport infrastructure layer

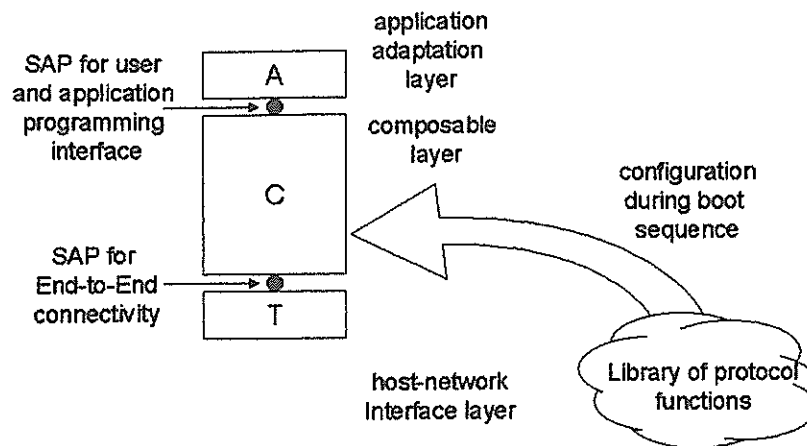


Figure 12.4 Composable protocols (DaCaPo)

While layers A and T, the adaptation layers, are dependent on the applications and underlying transport mechanisms (e.g. ATM, LAN, MAC, etc.), respectively, layer C is the configurable/composable protocol layer. It is comprised of agglomerated granular protocol building blocks, each defining a single protocol task. Figure 12.4 shows the principle of the DaCaPo framework, and of composable protocol stacks in general; the framework uses decomposed protocols, i.e. their ‘atomic’ functionality, implementing single protocol functions and properties.

The DaCaPo framework uses four cooperating entities to control messaging between these building blocks and binding them (i.e. binding them into protocol components) within the C-layer. The four entities include [7]:

- CoRA (a method for configuration and resource allocation) – determining appropriate protocol configurations at runtime;
- Connection management – controlling establishment, error management, and release of connections to peers;
- a runtime environment coordinating execution (linking, initiation, packet forwarding) of the processing within the layer;
- an entity to monitor the other components and control the resource availability within the communication end systems (i.e. message originator and message sink).

Although developed to implement complete protocol stacks, DaCaPo tackles reconfiguration and customization within protocols during implementation/boot time. The framework is currently being used as a support system to enable the development of a QoS ensuring middleware in the MULTE project (multimedia middleware for low latency high throughput environments) being undertaken at the University of Oslo [8].

12.2.2.3 Reconfigurable Stacks

The reconfigurable protocol stack approach is based on the redefinition of interfaces between protocol layers, classification of interactions between different layers within the protocol stack, and provision of an architecture to support protocol stack and protocol reconfiguration. This approach introduces and implements active programming interfaces in the form of

objects that become part of the protocol stack, using object-oriented design methods to define this protocol stack architecture and to replace protocol implementations during run time (following the 'class bloater' principle of the Java virtual machine).

Application programming interfaces (APIs) have already been in use for decades in the field of computing to simplify high level application development. Meanwhile, within the fields of networking and telecommunications the need for and potential benefits of common open interfaces on the application layer has been widely acknowledged. Several research projects have been initiated to explore the implementation and application of open programming interfaces and platforms and their use in mobile terminals and network nodes. However, programming interfaces applied in mobile environments commonly reside on top of standard traffic channels to provide virtually open access to wireless communication systems traffic channels or to specifically designed wireless application execution environments. Examples include MExE, 'On the Move', IEEE P1520 and 'the Radio API' [9–12].

Going beyond this conventional approach, using standardized active programming interfaces at all layers introduces an additional degree of freedom to reconfigure protocol stacks in both terminal and network [1]. A major function required for SDR terminals is the ability to exchange protocol software 'on the fly', implying the dynamic reconfiguration of the protocol stack. The OPtIMA framework [13] was developed to address this need.

OPtIMA is based on the separation (decomposition) of protocol stacks into a number of functional entities. These entities include protocol (pro-) layers, (pro-) interfaces and threads, described in generic 'classes' organized in class libraries which enable dynamic binding during runtime. The architecture can also support composable protocols as those introduced in [14], whereby single protocol layers can be replaced by a collection of components, each of which implements a particular function.

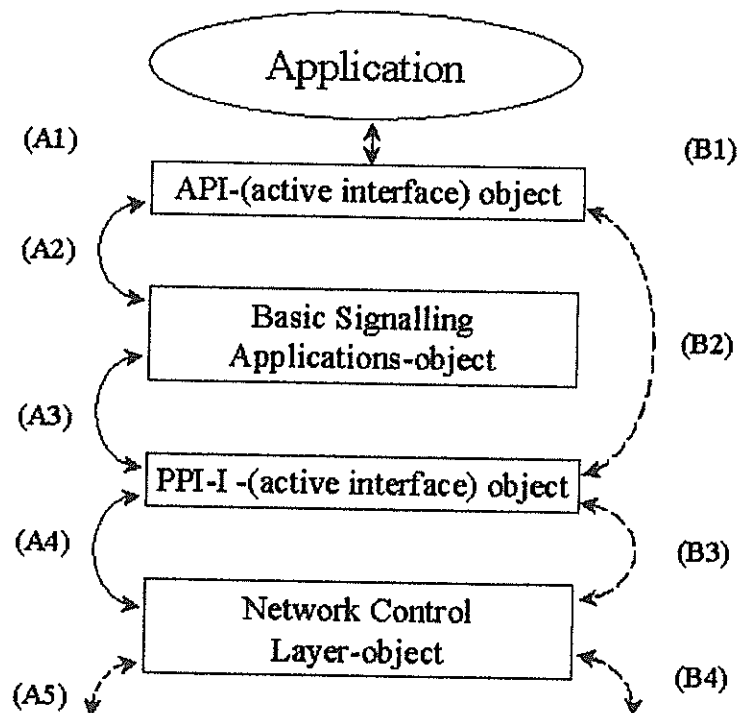


Figure 12.5 Active protocol interface objects

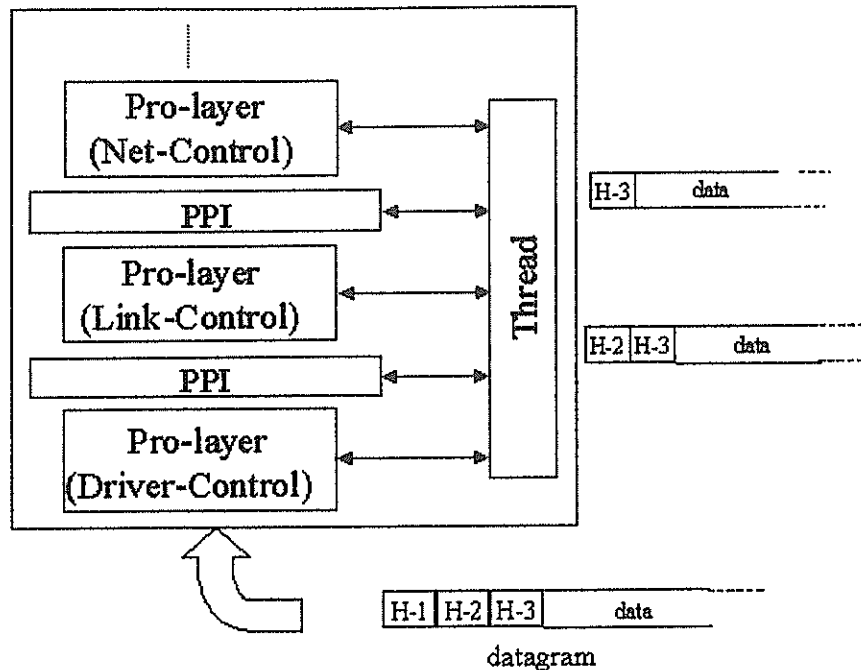


Figure 12.6 Message passing within OPTIMA

The OPTIMA framework is centered on the paradigm of active programming interfaces rather than using the straightforward approach of defining a broad set of APIs and PPIs which contain all the functionality of certain protocols. The use of active protocol interfaces adds more complexity to the systems but also provides the advantage of protocol exchange during runtime rather than during compile or boot time. Active protocol interfaces are objects which embody a set of message interpretation and message distribution methods; the structure is depicted in Figure 12.5. The active interface objects retrieve information required for the processing of the messages from the incoming message headers or, depending on the instantiation of the framework, from the (thread) object in which the incoming message is wrapped. The active interface object processes/interprets the message and passes it on to the target 'pro-layer'; Figure 12.6 shows the principle of message passing within the framework.

12.2.3 Active Networks

Another approach to accommodate different protocols and protocol stacks within a single network node is the IEEE P1520 proposal for active networks. This standardization effort aims to define a set of APIs for networks. The APIs are defined to interpret protocol tags that are attached to the single messages passed throughout heterogeneous network environments. A standardization working group, comprising several research labs in the United States, Singapore, and Sweden, is currently preparing the recommendations and specifications for programming interfaces of service, signaling, and switch control in future active networks. A major objective is to enable the development of open signaling, control, and management applications and also of higher level multimedia services on networks. The working group has already proposed a reference model that identifies a framework of interfaces and functional levels for the protocol stacks within 'active' network nodes [11,15].

12.2.3.1 P1520 Architecture

The introduction of open control and signaling interfaces for the various functional levels of network nodes has its rationale in the functional distribution between these layers, whereby a set of standardized APIs facilitates the access to the various functional levels and thereby enables the interpretation of protocol information contained within arriving packets. As the Microsoft foundation classes (MFC) and the Java developers kit (JDK) have shown for the area of high level application development, the open provision of standardized APIs can enable software developers to write control and signaling software so that even nonstandardized networking protocols can be used within such (active) networking environments. The eventual implementation of such an approach is expected to deliver widespread programmability for data and telecommunication networks. The protocol stack reference model for active network nodes, as depicted in Figure 12.7, defines four different levels for application programming interfaces that implement the various protocol stack features (services and functionalities) necessary to provide internetworking functionality.

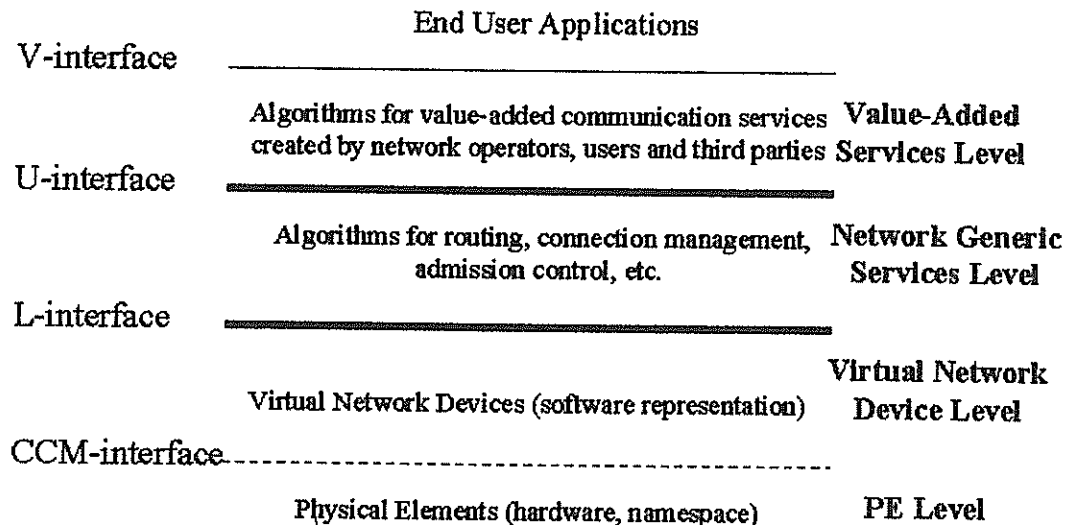


Figure 12.7 The P1520 reference model

The four protocol levels are the value added services level, the network generic services level, the virtual network device level, and the physical element level (PE level). Each of these levels implements the functions that are offered by the corresponding interface. This means in detail:

- *Value-added services level:* functional entities within this level implement end to end algorithms which add value to services provided by all lower levels. These value added services may be user-oriented features and capabilities, e.g. management of real time streams, synchronization of multi- media streams (video, voice, and data).
- *Network generic services level:* the functions implemented within this level include algorithms that define the general functionality of networks; these are mainly routing and virtual circuit/virtual path algorithms. This level also contains distributed object interfaces to service control points (SCPs) for intelligent networking (IN) applications using the

intelligent network application part (INAP); the means to carry out these distributed IN-related algorithms are currently not addressed within IEEE P1520.

- *Virtual network device level*: the functionality herein includes the logical and abstract representations of state variables of hardware (and soft coded) entities in the PE level.
- *Physical elements level*: PE-level entities are the actual physical elements of the network, such as switches, routers, or communication end points. Functional elements within this level are accessed and controlled by (open) protocols such as general switch management protocol (GSMP) and other, proprietary, management protocols.

The interfaces between these functional levels denote the separation between end user applications, value added services, the network service provide, and the underlying network resources. The P1520 reference model identifies four interfaces:

- *V-interface*: the V-interface is a user level interface providing APIs to write personalized end user software (i.e. applications); it also provides access to the value added services level.
- *U-interface*: this interfaces allows users to access the generic network services such as requesting or terminating connections or bandwidth. It furthermore allows the configuration of connections, ranging from establishment, maintenance, and termination of point to point, point to multipoint to virtual private networks, depending on user demands and physical availability of resources.
- *L-interface*: the L-interface provides a set of APIs that enables the direct access to, and manipulation of, the states of the local network nodes and resources. It also allows the implementation of any communication service.
- *CCM-interface*: the connection control and management (CCM) interface consists of various protocols that enable the exchange of state and control information between the physical hardware elements.

Altogether, the IEEE P1520 framework offers the potential to customize communication channels and traffic types across networks; the technology delivers a viable solution for bridging between different legacy access schemes.

Active networks and the range of described modular solutions for protocol stack implementation, although being partial solutions, provide the means for supporting the customization of protocols in software based communication environments. The use of such techniques in software defined radios will enable flexible protocol stack implementations instead of having to host multiple protocol stacks together within one terminal.

12.3 Reconfiguration Management and Control

Reconfiguration is a matter not only for the air interface between mobile terminals and base stations but also for other parts of the access network, such as the nodes along a communication path. Open communication systems and their flexible configurability will, eventually, facilitate data transport with a minimum of overhead across networks and, in particular, via the wireless link. However, on the way, if not properly implemented, network reconfigurability has the potential to introduce instability into communication networks. While the instability of a user's SDR handset can be irritating for the individual, instability of the network could have much more serious commercial repercussions.

In a highly reconfigurable network the potential exists for confusion over the current configuration status of nodes to impinge upon the complete reconfigurable communication system. This could result in message and/or data collisions and could lead to system failure due to nonconformant states within a network node or between different network nodes, e.g. if the configured standard differs between terminal and base station. Reconfigurability of complete communication platforms, ranging from hardware to the upper layers of the protocol stacks, is thus, implicitly, a threat to the communication platform stability. For these reasons a robust reconfiguration management scheme is required if the full benefits of SDR are to be realized.

Reconfiguration management has to accommodate the internal configuration of network nodes/terminals and the external relations of such nodes; external reconfiguration management is required to coordinate the states of the reconfigurable nodes along a particular communications path, while its internal counterpart is needed to control and manage the reconfiguration within the reconfigurable device (or network node) itself.

Reconfiguration may manifest itself in different variations/categories, each of which may have different classes/levels of effects on the communication link; these different categories need to be identified and the classes described. In the subsequent parts of this section we introduce two concepts of reconfiguration control (internal and external), we outline the peculiarities of terminal reconfigurability, and we identify these reconfiguration categories with a particular focus on protocol reconfigurability. Subsequently, we identify the network requirements for providing support for terminal reconfiguration.

12.3.1 The Scope of Reconfiguration Management

The major tasks of external reconfiguration management are to monitor current traffic requirements and settings between the communication end points and to ensure that the transport means between the terminal and the network gateway (or, if applicable, all the way through the network to the other communication end point) are synchronized (i.e. using the appropriate transmission standards). External reconfiguration management must also include the provision of additional, reconfiguration related, services. Within the approach developed by Mobile VCE, this latter role is undertaken by an AcA server (authentication/authorization/encryption – virtual configuration – accounting/billing), with associated databases used to store downloadable reconfiguration software. The architectural framework is shown in Figure 12.8 (see also [16]).

The scope of internal and external reconfiguration management is depicted in Figure 12.9. Despite the formal separation, internal and external reconfiguration cannot in practice be completely separated; their decision making processes and reconfiguration software acquisition and installation procedures are interrelated. For example, software download and (internal) reconfiguration validation within a handset are tasks that require support from the network (i.e. from the aforementioned AcA server and its associated software database). External reconfiguration management enables and supports the interactions between network nodes, including external (network originated) triggering of terminal (or network node) reconfiguration.

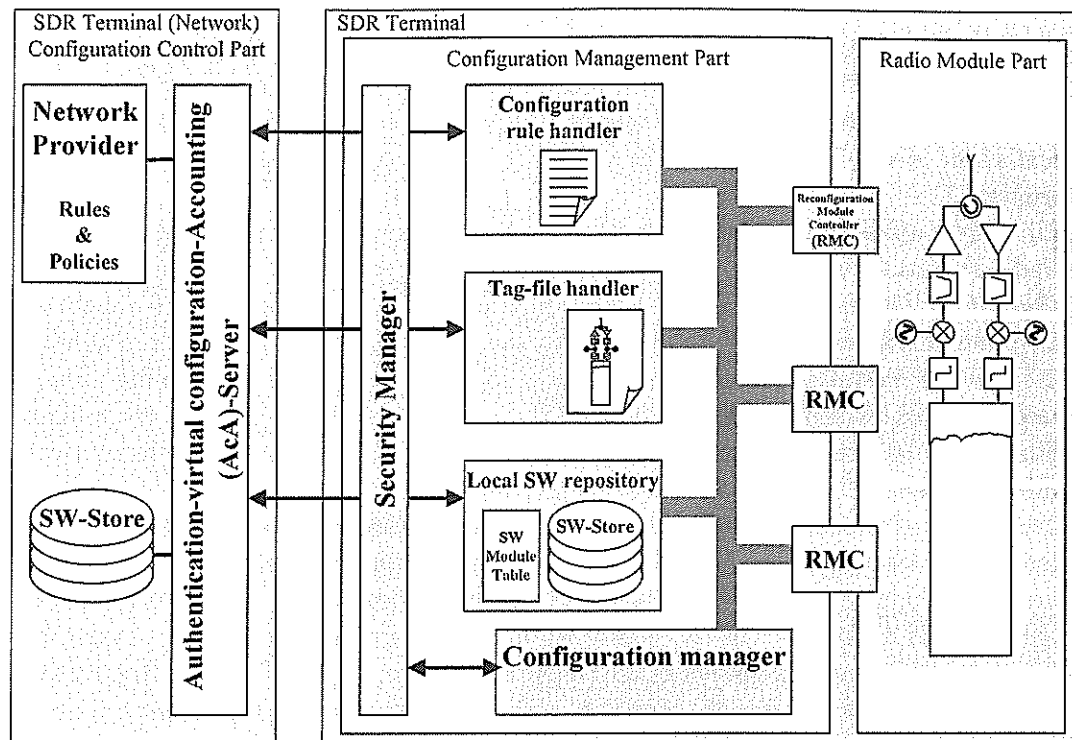


Figure 12.8 Reconfiguration management architecture

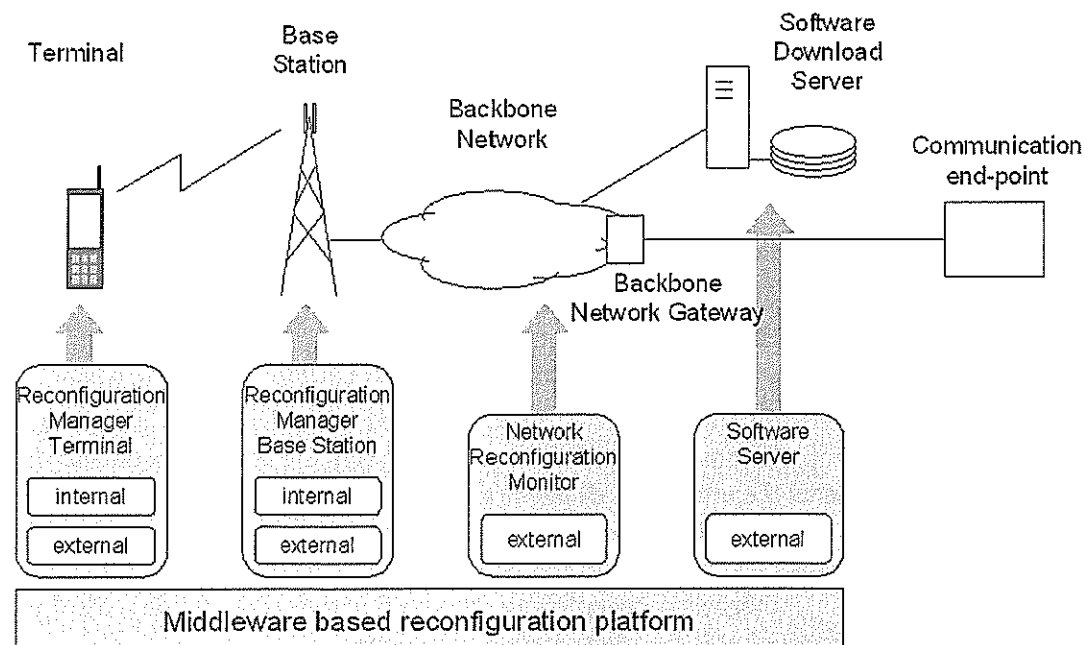


Figure 12.9 Communication network reconfiguration

12.3.2 Requirements of a Management Architecture

Open terminal platforms, through their programmability, allow an SDR terminal to be reconfigured to meet the air interface, network, and application specific requirements of any mobile communication system. However, reconfigurability of such terminals is not confined to the physical layer alone; it includes all levels of the protocol stack and application support environment.

An ideal reconfigurable terminal may be considered to comprise a collection of low to high level radio and protocol (software) modules, separated by open interfaces and controlled via a reconfiguration interface. Each of these reconfigurable radio modules offers its functionality (via the open interfaces) to the system platform and, where applicable, to other, neighboring, (reconfigurable) modules. Furthermore, each of the modules has to be reconfigurable (i.e. software modules exchangeable) via this reconfiguration interface. Such a reconfigurable platform could in principle accommodate any standardized, or indeed nonstandardized, transmission scheme.

Independent of the level within the terminal protocol stack, reconfigurability may be pursued in different ways:

- using parameterized radio (and protocol) modules
- exchange of (a) single component(s) within a module
- exchange of complete radio modules or protocol layers

While external reconfiguration management ensures the overall configuration of the network, terminal reconfigureability itself falls within the scope of internal reconfiguration management. Internal reconfiguration management must control and manage the configuration and functionality of a particular network node before, during, and after reconfiguration; it also has to facilitate the compliance of the node/terminal to given transmission standards and regulations.

The process of an internal reconfiguration sequence encompasses manipulation of a number of different reconfigurable modules such as the hardware interface (i.e. antenna, filters, converters, etc.), protocol stack reconfiguration (internally and/or using active wireless networks), and also adaptation of the application execution environment.

Many different SDR terminal architectural concepts have been published and some implemented.¹ One of the first was the United States Department of Defense-initiated SPEAKeasy platform [17]. This was followed by the SDR Forum open terminal architecture [18], the open terminal concept [19], and the joint tactical radio system (JTRS) model [20]. Common among all these models is that they split the terminal in (at least) two parts, one implementing the communication path (including antenna, RF, ADC/DAC, baseband, security, signaling, and basic communication applications) and the other part facilitating the configuration and reconfiguration processes within the terminal. All the models distinguish between lower and higher level functional blocks. Since each serves to implement a specific task, the single blocks may have different requirements for reconfiguration: some may be reconfigurable by parameters only (as proposed in [21]); others may need ready compiled native code; while a third group of modules may be reconfigurable by interpretable byte code.

¹ For background on SPEAKeasy and JTRS see Bonser, W., in *Software Defined Radio: Origins, Drivers and International Perspectives*, Tuttlebee, W. (Ed.), John Wiley & Sons, Chichester, 2002, Chapter 2.

Another approach includes the development of a radio definition language (such as radio knowledge representation language (RKRL) (see also [22]) that can be compiled after download, or may even be able to follow the Java platform model (by designing a virtual machine for radio software) with code interpretable during run time.

The described object-based implementations of the protocol software layers (i.e. by using the OPTIMA framework) imply embedding of various protocol functionalities that are captured within individual software elements (i.e. objects or components). Such functional elements will interact with other member elements within the layer, or even across different layers, via open programming interfaces. These elements will therefore contain implementations of protocol functionalities (or parts thereof) and their boundaries will be described by these interfaces. The reconfiguration management unit must ensure the proper exchange or replacement of the protocol elements (i.e. functions, layers, or the complete protocol stack).

To summarize, the three categories to be considered within an internal reconfiguration procedure are:

- *Partial reconfiguration of any layer (or part thereof) within an existing standard implementation:* partial reconfiguration refers to exchanging/updating/replacing one or more elements within a given protocol layer. The need for such a reconfiguration arises, for example, when a new/optimized functionality is to be introduced to the terminal (e.g. a new/optimized call control protocol element). As previously mentioned, the reconfiguration mechanism depends on compliance with open interfaces in order to guarantee proper functioning of the new/optimized element.
- *Complete reconfiguration of any given layer:* complete reconfiguration refers to exchanging/updating/replacing one or more complete protocol layers within a given protocol stack. Compliance with published/open interfaces between strata ensures proper operation of the layer/stack subsequent to this exchange/replacement.
- *Full reconfiguration of a complete stack:* full reconfiguration refers to the exchange/replacement of the full protocol stack by another existing standard or even a custom/proprietary nonstandardized protocol stack. Once again, compliance with ['to be'] standardized interfaces is presumed.

Finally, we note that the source of a request for reconfiguration introduces additional complexity to the reconfiguration procedure; requests for reconfiguration may be made either from within the terminal (e.g. user, application, radio requirements, etc.) or from within the network (e.g. network, service, or application provider). Both possible originators of a request must be assigned different priorities and rights for requesting terminal/radio link reconfiguration. For example, only the network provider may *authorize* changes to the air interface, while such a change may be *requested* by the user or any other entity.

Each of the three aforementioned reconfiguration categories may result in different effects upon the communications network. Reconfiguration classes describe and classify the possible effects that an intended reconfiguration process may have on the air interface. These effects may range from no change at all, to slight changes resulting in no interference to neighboring channels, to severe interference with neighboring channels. An initial classification identifies three different reconfiguration classes, summarized in Table 12.1, along with their operational implications.

Table 12.1 Reconfiguration classes

Class	Reconfiguration	Effect
0	– Application – Application execution environment – Parameter update	No effect for the radio link, no violation of the currently used radio standard, no effect to other users <i>requires</i> user and <i>may require</i> network authorization
1	– Parameter update – Protocol component exchange – Protocol exchange	Small effect for the radio link, no interference with neighboring channels, requires network and user authorization
2	– Parameter update – Protocol component exchange – Protocol exchange – Protocol stack reconfiguration	Moderate to serious effect for the radio link, and whole transmission, requires network and user permission + standard authorization

12.3.2.1 Soft Terminals and the Issue of Standards Compliance

Reconfigurable terminals and open platforms will enable third parties to provide and deliver software for the various reconfigurable modules (e.g. antenna, protocol stack, etc.) within a software definable terminal; the types of software may range from parameters for filter implementations or other signal processing aspects, to complete signaling applications/protocols for end user applications.

Reconfiguration of terminals is complex and offers a vast number of possibilities; the originator of a request, the category, and class, and also the type of reconfiguration must be considered and will influence the processing of a reconfiguration sequence.

The complexity of reconfiguration, and therefore the range of possible configurations or reconfiguration procedures, depends heavily on the variety of parameters and software modules that are used to describe a terminal or network node. The aforementioned reconfiguration categories reduce this theoretically infinite number of reconfiguration parameters and modules.

As a result of the complexity potentially involved in a reconfiguration procedure and the many (reconfigurable) features of terminals and network nodes, a major problem arising (with completely programmable radios/nodes) is how to ensure that any possible combination of software modules and terminal hardware conforms to a given air interface standard and to the physical benchmarks set by the relevant spectrum regulatory² authorities. Previous methods for permitting type approval or self-certification by the terminal manufacturers that demonstrated product compliance will not be possible in the future once terminals are completely reconfigurable and reconfiguration software is available from a wide range of sources, potentially outside the manufacturers' control. Furthermore, different software implementation for different reconfigurable modules may be available from different providers or vendors, in

² For further discussion of the regulation of SDR in Europe and North America, please see Bender, P. and O'Fee, S. 'European regulation of software radio' and Grable, M., 'Regulation of software defined radio – United States' in Tuttlebee, W. (Ed.), *Software Defined Radio: Origins, Drivers and International Perspectives*, John Wiley & Sons, Chichester, 2002, Chapters 10 and 11.

which case responsibility for ensuring standards compliance cannot be assigned to a single individual party.

To summarize: each radio and protocol stack module within an SDR terminal may be configured (and also reconfigured) independently, and reconfiguration, whether partial (i.e. affecting only one module (e.g. the antenna)) or complete (i.e. affecting all modules), may cause the terminal to adapt to a different access scheme. Therefore, two different requirements may arise:

- a mechanism that allows type approval testing/compliance to standards and yet still supports the use of open platforms (and thus the software provision from any third party), or, for a more progressive approach;
- negotiation mechanisms between the network nodes (terminals) which freely allow reconfiguration of the network node (terminal) within the scope of standardized interfaces, but without the need for any formal test approval mechanisms (in the traditional sense).

12.3.3 Management Architecture Implications

To accommodate the issues previously described, reconfigurable terminals will have to be viewed from more functional perspectives than the classical management, control, and user planes (cf. the GSM functional models). Reconfigurable terminals will have to execute additional tasks that stretch beyond traffic transport, call control, and connection management. They will need to manage a variety of tiers, each of which processes requests and executes certain tasks for new, value added, functionality. However, these 'value adding' functionalities will still rely upon the services offered via the classical traffic/user – control and network management planes – i.e. a new plane to support reconfiguration will be required alongside the old user/signaling/management planes.

Downloading reconfiguration software, via the air interface, from servers within the communication network, or from servers at sites of third party software providers, requires some additional network infrastructure and suitable storage devices within the networks. Part of this additional infrastructure is the provision of a management control unit, implemented within a server, whose tasks will include the coordination of possible reconfiguration requests, the management of reconfiguration software, and the verification of intended terminal configurations (reconfiguration software exchanges).

It has been noted earlier that a reconfiguration process may affect all functional entities within a terminal, ranging from the application layer to the physical interface. This 'global' type of reconfigurability requires strong reconfiguration management within the terminal but also the ability to interact with the responsible management units within the network environment. The distributed reconfiguration management architecture is a possible platform to implement this 'reconfiguration management plane' within reconfigurable wireless communication networks. This additional control level between distributed reconfigurable platforms, across the air interface, will require additional channels to transport the signaling messages needed, as well as network infrastructure extensions. Eventually, this demand, together with the need for software download and bootstrap (or pilot) channels, may necessitate the introduction of a new universal/global control channel. Such an additional functional plane (i.e. the reconfiguration management plane) will ensure secure and reliable reconfiguration of terminals and network nodes.

12.4 Network Support for Software Radios

Reconfiguration control and management, download of protocol and other reconfiguration software, and initial access to 'unknown' networks are prime functionalities that are required for over the air (OTA) reconfigurable software (defined) radios. There is a need for additional network infrastructure, capable of supporting reconfiguration and seamless cross system roaming and mobility management. There are multiple reasons for networks to support software (defined) radios: the number of co-existing radio access standards will be huge [23] and it can be assumed that one single radio standard will hardly be able to deliver multimedia services constantly at 'anytime, anywhere, and in any QoS'. Additionally, to enable cross system mobility, the access networks require additions to form a unified signaling infrastructure that will be able to provide a platform for signaling for mobility and call management, as well as for billing/accounting information gathering. Three approaches to such issues have been proposed – a network access and connectivity channel, a bootstrap channel, and a global/universal control channel – which we describe below. We then conclude with a brief discussion of the wider issues relating to the plethora of wireless access networks and their interconnection to provide the criteria for seamless interconnection and roaming that are claimed as the ultimate promise of reconfigurable networks.

12.4.1 The Network Access and Connectivity Channel

Full flexibility of mobile communication systems can only be achieved if connectivity to all possible wireless access networks is ensured. Reference [24] proposes, to enable this connectivity, a network access and connectivity channel (NACCH) in combination with a minimum air interface standard enabling access to this channel (see Figure 12.10). This symbiosis of minimum air interface and signaling channel is intended to provide not only the basic connectivity between network and mobile terminal but to be independent from possible co-located access network standards. Another task of the NACCH is to enable basic extra access standard network mobility management features, such as tracking and paging of a terminal as it roams across the boundaries of different access networks. This cross standard mobility management has to be active at least until the terminal has been configured – from the always inherent 'minimum' functionality to a complete configuration.

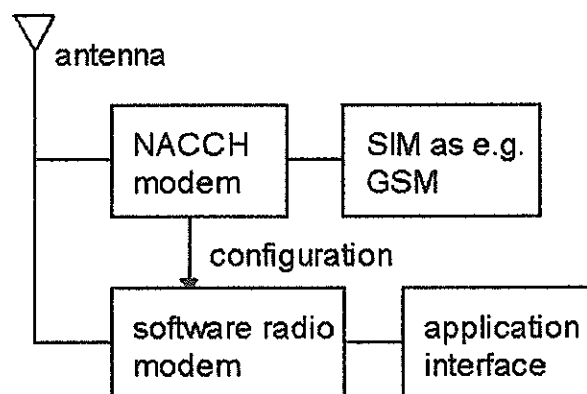


Figure 12.10 SDR terminal with NACCH, after [23]

The NACCH proposal assumes support of minimal connectivity to enable terminal configuration to the required wireless access configuration, independent of the current geographical position. To enable this, NACCH has to include a basic type of mobility management, authentication and registration facilities, call paging, and call establishment signaling, and also an interface to define traffic channels.

12.4.2 The Bootstrap Channel

Software download is regarded as a fundamental requirement for reconfigurable terminals [25]. Three major types of reconfiguration software may be downloaded to a mobile terminal. Reference [26] classifies these as 'end user applications', 'protocol stack and management modules', and 'signal processing algorithms and air interface parameters', whereby the size of downloadable software modules may even reach the Mbyte range. The access scheme proposal of [26] incorporates a 'bootstrap channel' (consisting of both a signaling and download subchannel), and also a so-called 'pilot channel' of small bandwidth and with the task of pointing to a download channel within the currently visited access network, i.e. this pilot channel constitutes a 'small bandwidth information channel' which may be implemented as a global broadcast channel transmitting information about local bootstrap channels (see Figure 12.11).

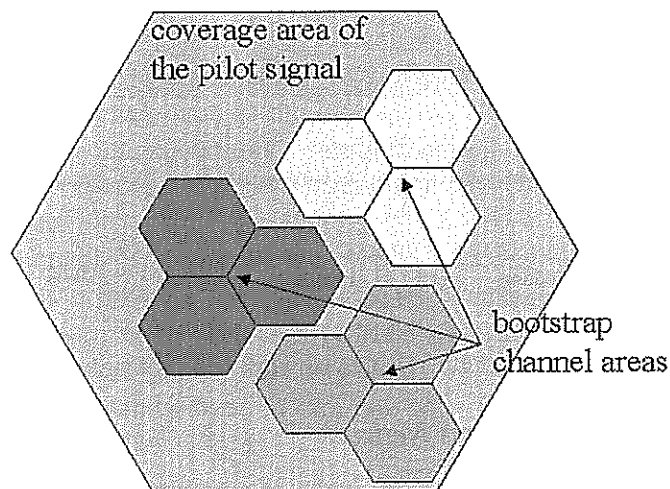


Figure 12.11 Bootstrap and pilot channels, after [26]

12.4.3 A Global or Universal Control Channel

Other proposals put forward in recent years include a global or universal control channel (UCCH). This concept not only tackles the air interface aspect, like the bootstrap and NACCH proposals, but looks beyond these into the signaling backbone network and the required protocol stacks.

The UCCH concept is intended to establish a signaling backbone, or bridge, to support software communication functionality between the different traffic network types. UCCH includes the definition of a support network topology, the necessary network entities, and the appropriate protocol stacks for these network entities. The UCCH network topology includes signaling applications to perform a number of signaling tasks, including:

- connection management (CM), i.e. establishment, maintenance, and release of signaling connections
- mobility management (MM), i.e. location management and support of ISHO signaling
- software download signaling

To cope with these demands, the UCCH control infrastructure consists of a OSI-like protocol stack containing an application layer, a transport and network layer, a link layer, and a physical layer. The different protocol entities within this signaling/support infrastructure address different parts of the underlying UCCH network; some messages need to be processed in each of the network nodes while other messages require direct end to end connectivity (e.g. software requests involve only the terminal and the software server while all nodes in-between are transparent to this message exchange). The protocol stack and network structure of the UCCH are depicted in Figure 12.12; UCCH signaling protocols for software download and mobility management are documented in [27].

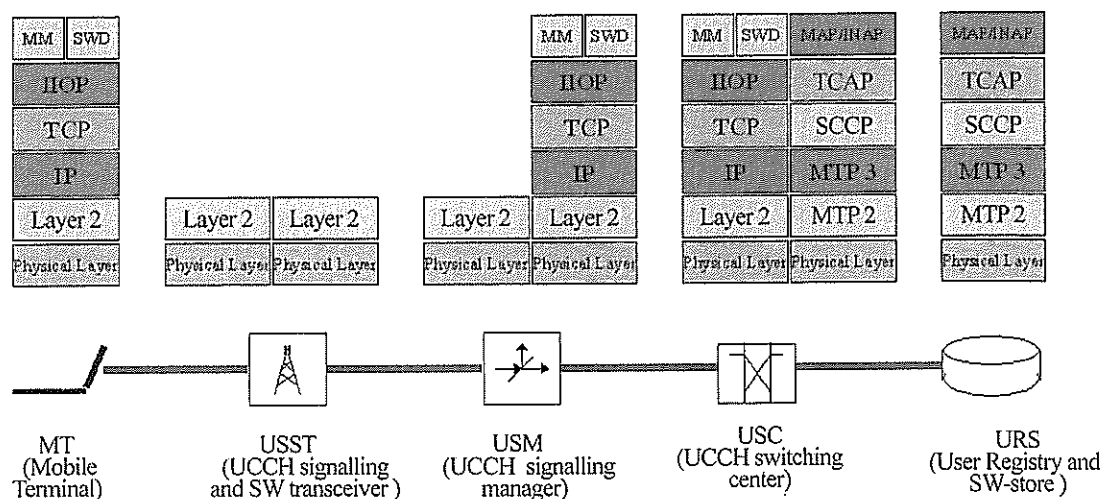


Figure 12.12 UCCH network and protocols

12.4.4 The Interconnected Seamless Network

Reconfigurability of terminals and network nodes is a precondition for the support of future seamless service provision in mixed network environments. So far, we have described the requirements and possibilities with regard to protocol stacks, their reconfigurability, and the necessary network support to enable this reconfigurability. The future wireless world is often depicted as a multitude of different wireless access techniques and networks connected via an IPv6 based core network. Within this scenario, SDR terminals are considered by some as a key technology that will enable this seamless multimedia, multiservice provision through their reconfigurability. However, a closer look at the infrastructure needed to implement this future vision raises numerous questions about the technological and commercial adaptability of wireless access networks to one 'cover and connect everything' network.

12.4.4.1 Access Networks

The range of available wireless access networks has, on various occasions [22,28], been

grouped into several categories ranging from satellite/HAPS and broadcast technologies, via cellular and quasi-cellular access networks, to the wireless local loop, and, finally, to local area networks, including personal and body area networks. Some of these technologies, such as GSM, DECT, etc., are already commercially deployed, while others are still in the planning and implementation process (e.g. UMTS, S-UMTS, etc.). A third group has not even reached that state, but the technologies are under discussion and in research (e.g. body area networks).

The features of these different access technologies are completely diverse: they are varied in their carrier frequencies and bandwidths, coding and modulation schemes, QoS provision and guarantees, and in their signaling structures and service types. Furthermore, these access networks assume different business models and use a variety of billing and accounting structures. With such a plethora of access schemes how might we, in reality, provide seamless access for users/terminals roaming across these networks?

Multimode terminals for a number of access schemes would be an initial solution, and provide a significant advance – indeed dual- and tri-band GSM phones are now common, and Bluetooth-enabled phones have also been introduced. However, the sheer number of different, possible access schemes makes the extension of today's hardware based approach to all access types something of a joke. Apart from the scalability problem, a true interworking, or only seamless roaming, between the different access technologies will not be possible without additional signaling infrastructure.

The availability of truly software defined radios could potentially deliver a commercially sensible and practical solution that could adapt to all possible access networks. Significant problems still remain, however, such as the interoperability between the different networks and issues like cross access technology roaming, billing for the different services, and network prioritization, in the case of a user served by several different networks and requiring a certain QoS guarantee.

The recent development of the universal mobile telecommunications service (UMTS) standard points towards a possible approach. UMTS licenses granted so far require that a communication connection via the UTRAN (UMTS terrestrial radio access network) must not be terminated because of [an initially expected] fragmentary network coverage. This requirement has led to the standardization of a one-way 'in-call' intersystem roaming from

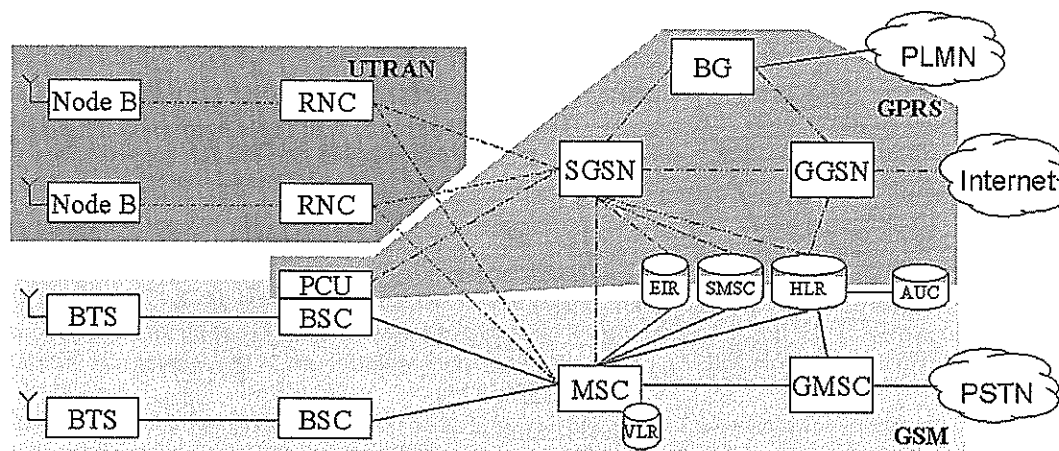


Figure 12.13 UMTS-GSM as example for network integration

UTRAN to GSM networks. Unfortunately, with current networking realities, this approach is a result of the evolution from GSM to UMTS, whereby UTRAN uses the GSM and GPRS networks as a backbone (see Figure 12.13) beyond its radio network controllers (RNCs). Apart from that, the approach is only valid for the particular case of UMTS to GSM roaming; it does not support in-call roaming from GSM to UMTS.

A network access and connectivity channel (NACCH), a 'bootstrap' or a global/universal control channel (UCCH), as introduced in a previous section, bear the capability to close this gap between the access networks and to introduce the desired seamlessness between these wireless access networks.

12.4.4.2 The Backbone

Connectivity between the different access networks is the key functionality that a backbone network infrastructure has to provide; this applies particularly to the anticipated migration of circuit and packet switched access networks. A backbone network ought to support the provision of globally accessible services independent of the attached access networks; it should also provide accessibility to any type of communication or data service a user may want to access. It is expected that any possible core network will be based on the Internet protocol (IP) suite, and at least the border gateways of access networks will be connected to the core network using IP. Apart from these general demands, core networks will require sufficient intelligence to cover issues like unified accounting and billing of users who employ the services of different access network providers.

Another issue with an IP-based backbone network is asset ownership. Although from a technological point of view the existing Internet model may be applied, it is arguable whether this model will satisfy the commercial interests of network providers; a distribution of

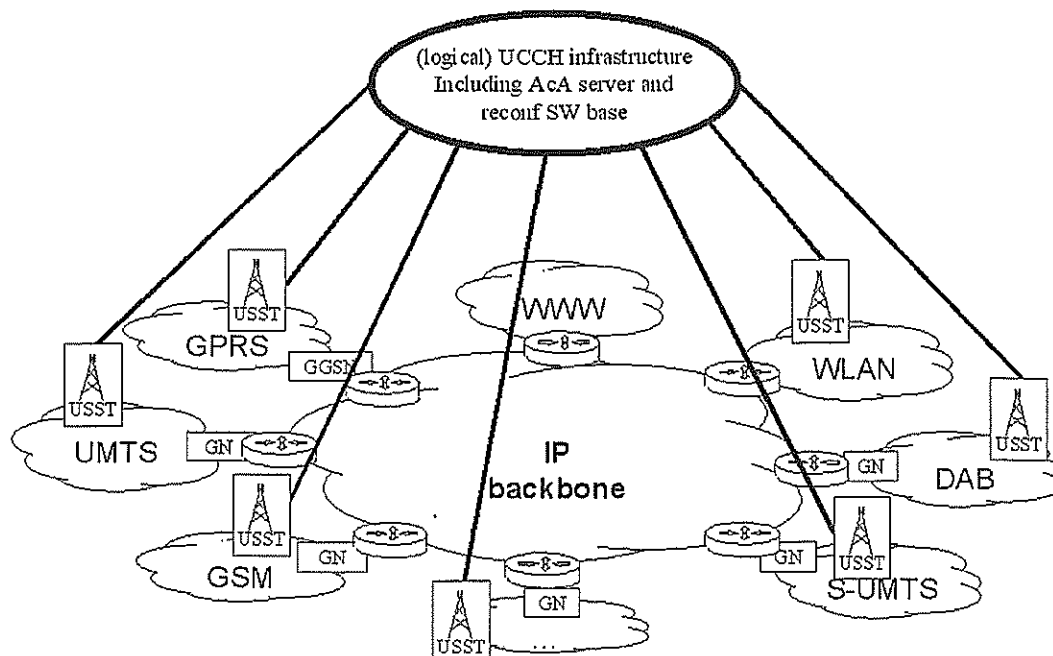


Figure 12.14 Interconnected access networks with UCCH

responsibilities and services may be a solution. Other technical challenges like local access to data, automatic recovery and synchronization of wireless transactions, and the implementation of [home] network provider policy tools to support the roaming of users into another provider's access network, will also have to be supported by the backbone infrastructure. The distributed nature of the Internet already provides the basic means for supporting these features, but connecting all possible active networks will require the introduction of additional management mechanisms that ensure synchronization of distributed subscriber and service databases.

Apart from these technical challenges, there are also commercial issues that have to be overcome: interconnection agreements between the providers of different parts of the networks and the conditions for these agreements have to be defined. So far, within the telecommunications industry there have been interconnection agreements between horizontally organized networks (e.g. in the North American market between GSM and DAMPS providers); however with the incorporation of vertically organized networks, essentially involving different industries with their own cultures and implicit assumptions, such mutual agreements will prove more difficult to achieve (e.g. agreements between local WLAN network providers and satellite or cellular access network providers, etc.) (Figure 12.14).

12.5 Conclusions

At first glance, software defined radio is often interpreted by the newcomer as simply meaning user terminals which can be reconfigured by software. In this chapter we have sought to explore the implications of such software reconfigurability, demonstrating a logical extension from the handset to understand the implications for network infrastructure.

Terminal reconfigurability can involve simple changes of parameters, an upgrade to a software module, or the replacement of a complete protocol stack. We have described the limitations of today's protocol stack approaches, using service access points, and have described various technical approaches which could facilitate the introduction of dynamically reconfigurable protocols. Having done so, we have then considered the risks of deploying such devices to illustrate the need for robust reconfiguration management mechanisms, which span both network and terminal and which give rise to the need for a new 'reconfiguration management plane'. Finally, we have briefly considered a range of issues that still remain to be addressed if the full potential of SDR for allowing full, seamless cross network roaming is to be eventually realized. While SDR undoubtedly has much to offer – to users, manufacturers, operators, regulators, and others – one still wonders whether such a goal may yet prove as elusive as a common global air interface standard.

References

- [1] Raivio, Y., 'Enabling technologies for the system beyond 3', *Proceedings of the Wireless World Research Forum (WWRF) Kick-Off Meeting*, Munich, Germany, March, 2001.
- [2] Mohr, W., presentation at the Wireless Strategic Initiative. An EU IST project, see <http://www.ist-wsi.org>, 2000.
- [3] Evans, B.G. and Baughan, K., 'Visions of 4G', *IEEE Electronics & Communication Engineering Journal*, Vol. 12, No. 6, December, 2000, pp. 293–303.
- [4] Tanenbaum, A.S., *Computer Networks*, 3rd edition, Prentice Hall, Englewood Cliffs, NJ, 1996.

- [5] Gumley, L., *Practical IDL Programming*, Morgan Kaufmann, 2001.
- [6] Siebert, M., 'Design of a generic protocol stack for an adaptive terminal', *Karlsruhe Workshop on Software Radios*, Karlsruhe, Germany, March, 2000, pp. 29–30.
- [7] Plagemann, T., 'DaCaPo – dynamic configuration of protocols', University of Oslo, Center for Technology at Kjeller (UNIK), <http://www.unik.no/~multe/multe-publications.html>, 1996.
- [8] Plagemann, T., 'Protocol configuration – a flexible and efficient approach for QoS', University of Oslo, Center for Technology at Kjeller (UNIK), <http://www.unik.no/~multe/multe-publications.html>, 1996.
- [9] 'Digital cellular telecommunications system (phase 2+): mobile station application execution environment (MEExE)', ETSI TS 101 438 V7.0.0, 1999.
- [10] <http://www.sics.se/~onthemove/>, 2000.
- [11] Lin, P., Suzuki, M., Redlich, J.-P., Cuervo, F., Biswas, J., Miki, K., Vicente, J., Denazis, S. and Gutierrez, J., *Programming Interfaces for IP Routers and Switches, an Architectural Framework Document*, IP Subworking Group IEEE P1520, 1999.
- [12] Beyer, D., et al., 'API framework for Internet radios', <http://www.rooftop.com>, further links within this url: Technology – APIs and Whitepapers, 1998.
- [13] Moessner, K., Vahid, S. and Tafazolli, R., 'Terminal reconfigurability – the OPTIMA framework', *IEE International Conference on 3G Technologies, 3G2001*, London, March, 2001.
- [14] Kulkarni, A. and Minden, G., 'Composing protocol frameworks for active wireless networks', *IEEE Communications Magazine*, Vol. 38, No. 3, March, 2000, pp. 130–137.
- [15] Biswas, J., Lazar, A.A., Huard, J.-F., Lim, K., Mahjoub, S., Pau, L.-F., Suzuki, M., Torstensson S., Wang, W. and Weinstein S., 'Application programming interfaces for networks', Working Group on IEEE P 1520 on APIs for Networks, *IEEE Communication Magazine*, Vol. 36, No. 10, October, 1998, pp. 64–70.
- [16] Moessner, K. and Tafazolli, R., 'Integration and configuration management', *26th SDR Forum General Meeting*, San Francisco, CA, November, 2001.
- [17] Lackey, R.J. and Upmal, D.W., 'SPEAKEasy: the military software radio', *IEEE Communications Magazine*, Vol. 33, No. 5, May, 1995, pp. 56–61.
- [18] <http://www.sdrforum.org/>, 2001.
- [19] Vahid, S., 'The Open Terminal Concept', internal report, Mobile Communications Research Group, CCSR, University of Surrey, 2000.
- [20] 'Software communications architecture specification', MSRC-5000SCA, V1.0, prepared for the Joint Tactical Radio System (JTRS) Joint Program Office, by the Modular Software-programmable Radio Consortium under Contract No. DAAB15-00-3-0001, May, 2000.
- [21] Wiesler, A., Muller, O., Machauer, R. and Jondral, F., 'Parameter representations for baseband processing of 2G and 3G mobile communication systems', *12th Tyrrhenian International Workshop on Digital Communications - Software Radio Technologies, and Services*, Portoferraio, Island of Elba, Italy, September, 2000. See also Chapter 8 by Friedrich Jondral in this volume which describes this approach.
- [22] Mitola, J. and Maguire, G., 'Cognitive radio: making software radios more personal', *IEEE Personal Communications Magazine*, Special Issue on Software Radio, Vol. 6, No. 8, August, 1999, pp. 13–18.
- [23] *The Book of Visions 2000 – Visions of the Wireless World*, version 1.0, <http://ist-wsi.org/BookofVisions2000.pdf>, November 2000.
- [24] Fettweis, G., Charas, P. and Steele, R., 'Mobile software telecommunications', *Proceedings of 2nd EMPCC'97/3. ITG-Fachtagung 'Mobile Kommunikation' (ITG-Fachbericht 145, VDE-Verlag Berlin, Offenbach, 1997)*, Bonn, Germany, 1997, pp. 321–325.
- [25] Cummings, M., Heath, S., Drew, N., Nuotio, M. and Williams, L., 'MMITS standard focusses on APIs and download features', *Wireless Systems Design*, April, 1998, pp. 45–49.
- [26] Noblet, C. and Aghvami, H., 'Assessing the over-the-air software download for reconfigurable Terminals', *IEE Colloquium on Personal Communications in the 21st Century (II)*, Digest, February, 1998, pp. 6/1–6/6.
- [27] Moessner, K., Vahid, S. and Tafazolli, R., 'A minimum air interface implementation for software radio based on distributed object technology', *IEEE International Conference on Personal Wireless Communications – ICPWC'99*, 1999, pp. 17–19.
- [28] www.wireless-world-research.org, 2001.