DESIGN Story: A Hiperlan2/IEEE802.11x reconfigurable SoC for indoor WLANs and outdoor wireless links. A pilot project for the future generation configurable wireless communications products


Abstract

A design story will be presented for the development of a reconfigurable System on Chip (SoC) that performs the complete baseband processing of an OFDM wireless transceiver. This SoC will form a part of a Hiperlan2 and/or an IEEE 802.11a WLAN Network Interface Card or of an Access Point. The same component could be used for outdoor wireless links. An emulation of this baseband processor will be implemented first on a reconfigurable prototyping board and it will be tested exhaustively until to verify completely the required functionalities first of Hiperlan2 and then modifications will follow for realizing the IEEE 802.11a standard. Next priority is to modify the development, to function as access point (mobile terminal will be the first implementation). This will be achieved by instantiating the required peripherals in each implementation (Access Point –AP- or Mobile Terminals –MT-) on the FPGAs of the reconfigurable prototyping board and test the systems. Last but not least the indoor implementation of the AP and MT, for the H2 standard will be modified algorithmically to perform in outdoor environment (Point-to-Point links of high distances -5 up to 20 Km-).

It is a gradual approach that will lead step by step to the development of a reconfigurable SoC that at its final stage will cover a broad area of systems that represent very important units of wireless communication products. By applying the methodologies for hardware-software partition and then for the identification of the reconfigurable parts of the hardware logic, a first implementation for the first system (H2 Network Interface Card –NIC-) will be developed on the FPGA prototyping platform and it will be verified. It is unfeasible to decide at the beginning of the development for architecture and a partitioning that would be the optimum for all targeted systems. Proceeding to the next steps by modifying the design and the partitioning decisions (HW-SW and reconfigurable part), it is expected that it will deliver at the end an updated methodology to be followed for the development of similar reconfigurable SoCs for wireless communications.

The FPGA prototyping board will allow the test of each implementation (H2, IEEE 802.11a, MT and AP, and finally of the outdoor link). Then after the complete testing and verification of each of the systems, the final parts of the SoC that need to be reconfigurable for instantiating the SoC to each of the different systems, will be decided.

The next step will be to turn the FPGA implementation to the final SoC putting the parts of the circuitry that would need to be reconfigured on the reconfigurable cells of the SoC. The transition from FPGA implementation to SoC will require particular treatment for important issues as clock handling, synthesis, testability, debugging etc.

The most important issue that will be achieved is that it will open a new era for system manufactures of low volume wireless products. These systems (e.g., outdoor links) traditionally are built by large and expensive FPGAs and a high-price Bill of Materials (BoM). ASIPs that usually address only mass-market applications cannot provide the necessary features needed by the low volume products. The existence of reconfigurable SoCs that address mainly mass-market products but also can be configured to provide the functionalities of other systems that are of low volume will bring to the companies involved in that markets (usually medium size companies), the competitive advantage of lower prices and/or higher profit margins. Additionally any system that will use it, it will be possible to upgrade to future versions whenever algorithmic modifications increase the performance (maximum distance of the link), and improve its competitiveness.

The paper structure will be as follows:
Section 1 is addressing the market trends that made the last years Wireless LANs to be one of the most attractive developments for a company that develops wireless communication products. Also the need in that category of products, of a SoC and furthermore reconfigurable SoC, is analyzed. Section 2 will present the similarities and differences of the Hiperlan2 and IEEE 802.11a standards. Section 3 will describe the differences between an Access Point and a NIC. In the same section the outdoor bridge product additional features will be explained. Section 4 is presenting the alternative architectures that will be developed and the criteria to choose the optimum one. Section 5 is describing the FPGA prototyping platform adopted for the development. Section 6 is presenting the verification methodology that will be followed and for all the systems, the reconfigurable SoC will address. Finally the transition planned from the prototyping platform to the final ASIP will be at section 7, when the conclusion will finalize the paper.

1. Introduction

In wireless data communications there have been several standardization efforts (including GPRS, EDGE, and UMTS), which had as a target to meet the expected increased requirements of users and applications. The 2 Mbps offered by UMTS is considered as not sufficient rate for the services that will require concurrent access of a plurality of wireless users in data that will contain images and/or video.

In addition, and complementary to the mobile telephony data transmission standards there where developed in Europe, Japan, and the US, standards for Wireless Local Area Networks in the 2.45 GHz and 5 GHz bands. In the unlicensed band of 2.45 GHz the IEEE 802.11b standard, has provided to the users up to 11Mbps transmission rates. The IEEE 802.11a and the Hiperlan2, (which is being specified by the ETSI BRAN project), were specified to provide data rates of up to 54 Mbit/s for short-range (up to 150 m) communications in indoor and outdoor environments.

The European and US standard have a very similar physical layer and their combination on a single piece of silicon is considered a feasible development. Europe and USA cover about the 75-80% of the world market in wireless communication products including mobile telephony products [1]. A baseband processor component for a product compliant to those 2 standards which have very similar physical layers, would have as alternatives both US and Europe’s market. The estimated worldwide enterprise market for WLAN products for the next 3 years (2003-2005) is expected to be 7.2 billion USD. The number of shipments for the same period for Hiperlan2 and IEEE 802.11a Network Interface Cards (NICs), is expected to be around 37 million pieces and their price is expected to be less than 70 USD by 2005 [1]. The corresponding number of Access Points (APs) is estimated to about 5.4 million pieces with prices less than 250 USD by 2005 [1].

It is evident from the quantity of the worldwide shipments and the estimated NIC prices that a System on a Chip solution is the only cost-efficient solution for the realization of the baseband processing for the OFDM reception. The cost of a SoC of about 1MGates with integrated A/Ds and DAC is estimated to be about 15 USD. An Access Point built with FPGA and discrete microprocessors, would have a BoM higher about 100USD.

An Access Point implementation using the same SoC as used for a NIC would allow a significant reduction in Access Point prices. Then such an ASIC would address a much larger market (50% larger than the NICs only), because of, economy of scale, the component’s price would decrease and consequently the associated products’ prices. Access Points may use as a core for their system architecture a NIC interfaced through its “CardBus interface”, to a low cost general purpose processor (ARM9 or 486) for realizing the additional features of an Access point (Web based configuration and monitoring etc.).

On the other hand the modification and verification of the NIC/AP SoC architecture, is a major task and would require a significant delay to introduce in the market a NIC/AP SoC for WLANs. Nevertheless a reconfigurable SoC, verified initially for NIC operation may be introduced on-time in the market, and then a new configuration for the reconfigurable part of the SoC would turn it to an AP version verified and fully functional. An outdoor
configuration of the same reconfigurable SoC may follow for wireless Point-to-Point links. The latter will have an additional advantage by modifying the reconfigurable circuitry. It would be possible to improve at a late stage the performance of the modem (the algorithmic parts that are put in the reconfigurable circuitry). The most important improvements would concern the increase of the maximum distance of the link. Downloading the new firmware to the reconfigurable Soc a new version of the complete outdoor system would be available to the customers.

Next sections analyse the systems under development. For the modem part algorithmic developments for the physical layer of all 3 implementations have resulted to satisfactory simulations [2-5].

2. H2 and IEEE 802.11a

Both standards support wireless data rates of between 6 to 54 Mbit/s using OFDM (Orthogonal Frequency Division Multiplexing) [6]. This multiplexing technique involves aggregating several low data rate channels to form a high rate wireless link. The low rate channels are less prone to ISI caused by multipath propagation compared to a single high-speed link.

The two standards may be handled by the same MODEM hardware for their physical layer [7]. The block diagram of the Rx and Tx (MODEM part only), is shown in Figure 1. The blocks for the two standards that have differences and have to be “dual-mode” are, in the Tx part, the FEC (different puncturing) and the stored preambles, when for the Rx part, the depuncturing unit only.

MAC is different and more demanding in the IEEE standard but in both standards the receiver reads the header of the frame to be informed about the destination, the physical mode and the quantity of data in the following frame.

Concerning the higher levels of the protocol stack, Hiperlan2 is confined to the functions, which are located in the two lowest layers of the open systems interconnection (OSI) model. The Hiperlan2 basic protocol stack [8] consists of the physical (PHY) layer on the bottom, the data link control (DLC) layer in the middle and one or more convergence layers (X-Specific CL) on top. The basic DLC functions are the Radio Link Control (RLC), sub-layer functions (Radio Resource Control, Association Control, DLC Connection Control) and the Data Transport sub-layer functions (Error Control and Medium Access Control).

Figure 1: MODEM Block Diagram
The IEEE 802.11 MAC protocol [9] provides two types of service: asynchronous provided by the Distributed Coordination Function (DCF) which implements a Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol as the basic access method, and contention free provided by the Point Coordination Function (PCF) which implements a polling access method. The PCF relies on the DCF.

The implementation of the higher levels of the protocol stack seem to be software implemented and thus is not an issue for the reconfigurable SoC development.

Below are described the differences of the PHY and MAC layers of the two standards in detail.

MAC

Since, both standardization bodies have worked together in order to harmonize the physical layer of these two standards, the main differences between these two standards are in the Medium Access Control (MAC).

The IEEE 802.11a OFDM frame format includes protocol specific fields. It consists of the PLCP (Physical Layer Convergence Procedure) preamble, the PLCP header, the PSDU and the tail/pad bits, as shown in Figure 2. Within the PLCP header is defined the rate and the length of the following PSDU where the details of the sender and destination of the frame are described.

Hiperlan2 OFDM frame has a broadcast part informing the mobile terminals about the identity of the Access Point and the content of the H2 MAC frame along with acknowledgment of requests for transmission. The length and the rate of the Hiperlan2 frame are determined on MAC layer level by reading the content of the broadcast phase (FCCH part). Then the downlink part follows where mobile terminals allocated for the current frame receive their data. Downlink phase is followed by the uplink phase where the Access Point receives data from the mobile terminals. Last part of the H2 MAC frame is the RCH (Random Access Phase), used for radio resource request, handover and initial access to the network. The structure of a H2 MAC frame is shown in Figure 3.

Comparing the structure of the two different frames is obvious that IEEE is much more demanding. For identifying the physical mode of the following part of the frame (after the header), there is available only 1 OFDM symbol (the header) plus the latency of the receiver (until data reaches the demodulator -constellation decoder-). Hiperlan2 has available to set it’s

---

**Figure 2** IEEE 802.11a frame structure

**Figure 3** Frame structure of Hiperlan2
receiver at the appropriate physical mode, at least 12 OFDM symbols (plus the receiver’s latency as well as in the IEEE case).

Control circuitry of the MODEM for the Tx and Rx part is different for the two standards. H2 MAC frame is of constant length (2 msec) when IEEE is of variable length defined within the header of the frame. In IEEE standard there are control frames (RTS and CTS) that have to be formed or decoded very fast at HW level (else the communication will be inefficient). Address decoding has to be done by HW to identify the validity of a control frame. CRC has to be performed at the end of the variable length packets and the acknowledgment messages after the CRC check have to be sent immediately (HW implementation). Various timers (measuring Tx and ACK ‘distance’), are required.

The CSMA approach used by IEEE necessitates the existence of a Finite State Machine (or dedicated microcontroller), for implementing the back-off algorithm.

**PHY**

The operating frequencies are distributed in a low and in an upper band. The low band is for both 5.150-5.350 MHz at 200mW when for the upper band Hiperlan2 uses the 5.470-5.725 MHz band with a power limit of 1 W. IEEE’s upper band is 5.725-5.825 MHz with power limit the 50 mW. The frequency channels have a bandwidth of 20 MHz. In both standards (ETSI and IEEE), 48 active subcarriers plus 4 subcarriers for pilot symbols are used. A 64-point FFT and IFFT is adopted by both standards for moving between the frequency and time domains.

Both IEEE 802.11a and Hiperlan2 standards scramble the input data with a length of 127 pseudo random sequence, but the initialisation process is different (minor implementation cost). IEEE 802.11a initialises with 7 random bits which are inserted as the first 7 bits of each packet, while in Hiperlan2 the scrambler is initialised by three 1’s plus the first 4 bits of the broadcast channel at the beginning of the MAC frame. The subcarriers may be modulated by four different modulation schemes (BPSK, QPSK, 16 QAM and 64 QAM). A convolutional coding with coding rates ½ and ¾ is selected by both standards to implement a Forward Error Correction (FEC) technique. Additional coding rates are supported by each standard (H2 9/16 and IEEE 2/3). Hiperlan2 uses an extra puncturing (P1) to accommodate the tail bits to keep an integer number of OFDM symbols in 54 byte (LCH) packets. This extra puncturing operation punctures 12 bits out of the first 156 bits of a packet.

There is also need for different types of convolutional decoding. On top of that specifically for the convolutional decoding there are short packets to decode (for IEEE 802.11a) and long packets (for both IEEE and ETSI). In Hiperlan2 only in the broadcast phase FEC coding appends tail bits in three different points in the broadcast PDU train, forming three sequential long packets to be decoded separately. This requires reset of the convolutional decoding procedure and immediate output of the remaining data for start the new decoding. Normally in IEEE 802.11a long packets decoding have a distance of 32μs (DIFS) and in Hiperlan2 12 μs. This particular need of immediate output of convolutional decoding results for the header of IEEE 802.11a and the broadcast phase of Hiperlan2 may be addressed either by putting the convolutional decoder to the reconfigurable part and instantiating dynamically the appropriate decoder, or, implement a decoder with the capability of flashing out the decoded results at the end in only few cycles.

Depending on the flexible frame composition (download phase from AP/CC to MTs, one or several uploads in the reverse direction, contention phase), Hiperlan2 can use several types of preambles with either a long sequence (from five up to ten), of short symbols followed by a short sequence of two long symbols (broadcast, uplink –short and long–), or just a short sequence of two long symbols (downlink preamble). On the other hand IEEE 802.11a uses only the PLCP preamble, which contains ten short and two long symbols for the same purpose as in Hiperlan2. In both standards long sequences of ‘short symbols’ are used for synchronization, antenna selection in antenna diversity implementations, coarse frequency acquisition and Automatic Gain Control (AGC) of the RF/Analog front-end. Short sequences
of long symbols are there mainly for channel estimation and fine frequency correction in combination with the pilots. The different preambles do not mandate different implementation of synchronisation but only additional stored preambles (in the Tx), for facilitating both standards.

Implementation of the demanding parts of the MAC layers of both standards (control frames and back-off algorithm for the CSMA procedure of IEEE, and the frame content-decoder of H2), on reconfigurable hardware would be feasible and it is planned to be the approach to follow. Physical layer differences (puncturing, depuncturing and different preambles), will be implemented in the fixed logic of the Reconfigurable SoC as additional supported operation modes for those blocks.

3. Mobile Terminal, Access Point and Outdoor links, similarities and differences

Access Point and mobile terminal differences are investigated at three different levels. PHY (MODEM), external interfaces and DLC/networking levels for both standards H2 and IEEE 802.11a.

At MODEM level Access point and mobile terminal in the Hiperlan2 standard have only few differences. Synchronisation for the mobile terminal has been designed for operation with all preambles of H2 standard (downlink, uplink, direct link and broadcast) and of IEEE 802.11a as well. Access Point receiver has to support receiving of consecutive bursts with different frequency offset, channel impairments and physical modes. The receiver of the Mobile Terminal is mandatory to handle consecutive PDU trains with different rates that need different channel estimation, and synchronisation (for the case of Direct Links). Thus the entire receiver chain is designed to handle the worst case of the studied units and no reconfigurable hardware is needed.

Access points and mobile terminals similarities and differences at the level of interfaces and networking are the same for both Hiperlan2 and IEEE 802.11a. The external interfaces of access point and mobile terminal units differ. Access point is connected to the Local area network (Ethernet 10/100 BaseT) using the standardised interface (IEEE 802.3 MII interface). Mobile terminals are Network Interface Cards (NICs), which have either a CardBus interface (PC cards) or a PCI. In the reconfigurable SoC, the PCI or Ethernet MAC Controller will be implemented in the reconfigurable part of the SoC, depending the target system. Anyhow additional pins have to be reserved in the chip for allowing the CardBus configuration.

Internal system interfaces of NIC or Access Point (communication between the Reconfigurable SoC and the system memories -SDRAM etc.-), are the same for access points and mobile terminals (the selected architecture and the 32 bits address bus satisfies the increased memory requirements (data queues and program code) of the access point.

DLC requirements of mobile terminals and Access Points in Hiperlan2 are different. Access Points have to schedule for each frame data and/or control information traffic. Also the control data received at every frame from all the transmitting mobile terminals have to be processed and served. Mobile terminals on the other hand have only to process the control messages they receive from the Access Point and then to send their answers and requests. The processing of the control messages remains the same for both systems. Extra DLC needs of access points have been measured and it was shown that they may be satisfied.

IEEE 802.11a differences between access points and mobile terminals at the level of DLC are the differences in the implementation between the Distributed Coordination function (DCF) and Point Coordination Function (PCF) which implements a polling access mechanism. The additional task that a Point Coordinator (PC) has to implement, is the polling method (building and continuous update of the polling list which is a similar procedure done by the H2 scheduler) and the association and disassociation requests of the terminals. The same
choice made for Hiperlan2 access point (ARM9 processor) will be sufficient and for the IEEE 802.11a operation, without any reconfigurability requirements.

Networking needs for mobile terminals and Access Points are similar. Access to a Management Information Base is required using either SNMP or http protocols. There is need to support a protocol to communicate with the unit and either configure it, or monitor its performance. PCs will have drivers that will communicate with the NICs through the PCI interface using an appropriate protocol supported by the NIC’s processor. Access Points are connected to the 10/100 BaseT connector and Ethernet packets facilitate access to the MIB. The additional support of these higher level protocols to manage and monitor the access point unit, require additional program code and processing capabilities. The monitoring and configuration needs, are not real time and as such it is possible to be served by the ARM9 protocol processor. If better performance is required, for a concurrent monitoring and operation, an additional processor would be added to the unit.

Access Points and Mobile Terminals in both standards do not have such different hardware needs to be beneficial to put on reconfigurable hardware for the DLC/networking and MODEM requirements. Additional processor in Access Points and mobile terminals could give the opportunity to the unit for monitoring during normal operation. Nevertheless the interface of the system should be put on the reconfigurable part to support either CardBus interface (for a mobile terminal), or Ethernet MAC MII interface (for an access point).

Bridges, are Access Points that may also act as mobile terminals. As such the same comments made above are valid for the interface/networking level and DLC. For the physical layer of the MODEM, it is a significant competitive advantage for a manufacturer to have a bridge that outperforms at outdoor environment (achieve longer distance links). Point-to-Point links may be different than standards and may be proprietary. Some of the parameters of the MODEM that could improve the distance of the link are, higher number of subcarriers, additional decoding unit and higher bit precision at the equalisation block, or an additional equaliser. Putting those parts of the MODEM on reconfigurable hardware (FFT, IFFT, additional decoder, equaliser), could give the opportunity to a system manufacturer using the reconfigurable SoC as a core of a bridge system, to update a product and achieve better performance whenever a new version of the MODEM is verified on a prototype version.

4. Alternative Reconfigurable SoC Architectures

The baseband processor consists of processor cores, the modem core, internal memory and a DMA engine for fast data-transfers plus a number of peripheral components for I/O and auxiliary tasks. All these components are organized as master and slave peripherals of a central AHB bus.

![Figure 4: Abstract view of a typical AHB system.](image_url)
entire clock, thereby avoiding complications with synthesis for high-speed operation (synthesizers prefer clock-to-clock paths for high-quality results). The data and control signals bus is multiplexed and controlled by a central bus arbiter, while the read/write data and slave response signal multiplexers are controlled by a central address-decoder. A representative AHB bus is shown in Figure 4.

Two alternative architectures/implementations have been studied to realise the reconfigurable SoC. Both architectures have an ARM7 TDMI RISC core to implement the protocol stack (or an ARM9 in case that the complexity of the Access Point protocol stack will be too high). The two alternatives concern the way that the lower MAC controller will be implemented (handling of the MACPHY requirements). The first choice will be to have it on hardware and the second is to design a dedicated reconfigurable microcontroller that is expected to be friendlier to modifications. Both architectures will be tested/verified.

In both alternatives the reconfigurable parts implement the tasks that were identified in the previous two sections as appropriate for reconfigurable hardware (RF interface, the PCI or Ethernet interface, the H2 or IEEE 802.11a lower MAC layer tasks).

Figure 5 shows reconfigurable System on chip architecture (reconfigurable blocks are dotted). One of the two ARM processors serves as protocol processor, while the other serves as networking processor for configuring and monitoring the system and if required to run other tasks (eg. routing). The protocol processor has locally its memory where are stored, the protocol stack program and the queues of the received control messages (DLC H2 or IEEE 802.11a). It communicates with the modem core through the AMBA bus (trunk bus of the architecture). The two processors communicate via the trunc AHB bus connecting the two domains. For quick block-memory transfers, a DMA-Engine is attached to the AHB, configured as a third bus-master of the highest priority.

Figure 5 Reconfigurable SoC architecture

The DMA Engine features a dual address-generator/bus-controller, as it is attached to the main (trunk) bus, and concurrently to the MMU local bus. DMA facilitates local buffering of transferred data for read-then-write bus-access to both ports. It is possible to read from any of
the two ports data and afterwards write it to any of two. It features a configurable number of request-channels, which are individually programmable to perform block, multi-block and scatter-gather memory transfers. Each channel is statically assigned a priority figure plus one of two privilege attributes. This scheme allows the implementation of a TDMA algorithm for sharing bus-bandwidth among requesting clients. The transfers that are handled by the DMA are the following (mentioned by their priority): MODEM buffer and SDRAM data queues, MODEM buffer and control messages queue, Ethernet buffer memory and SDRAM, Ethernet buffer memory and PC memory (in case of NICs), program memory of networking processor and SDRAM, program memory of protocol processor and Flash memory, SDRAM and Flash memory. The clients of the DMA are, the MAC controller of the MODEM, and the two processors.

Having a single trunk bus and local busses behind the AHB interfaces eases chip resource requirements (which would be large for the case of two complete AHB systems, one per processor) as well as design considerations for cross-bus communication HW/SW.

A specialized MAC/PHY controller can be used to replace the logic circuitry of the MAC controller reconfigurable block in the architecture in Figure 5. The controller, features limited programming capabilities compared to a general-purpose core such as the ARM, but on the other hand it can be tuned for high-performance on the tasks specific to modem-control and lower-MAC processing and is more flexible than the logic circuitry.

5. Description of the reconfigurable prototyping SoC platform

The first implementation of the SoC under development will be on a prototyping platform (ARM Integrator). The platform hosts the bus of the architecture, with all the required support peripherals for its operation (arbiter, interrupt controller etc.), on an ATX type motherboard that is also possible to be plugged in a Compact PCI rack. The logic circuitry of the SoC is on FPGA “logic modules” (modular up to 4) and the ARM microprocessor instantiations on the “core modules” (multiple up to 4). The modules seat on the motherboard as daughter-boards. There are SSRAM and FLASH memories on the motherboard of the platform that allow the storage of data for communicating the various blocks of the architecture, but also for booting the microprocessors. There are PCI slots on the motherboard to plug peripheral units (eg. Ethernet card), controlled by a PCI-AMBA bridge. The total allowed number of bus masters of the prototyping architecture is five (5) and is a limitation put by the arbiter implementation of the bus. The bus implemented in the motherboard is the AMBA bus, which connects the master and slave blocks of the architecture through the bus multiplexers that are controlled by logic mastered by the arbiter.

The “core modules” are plugged (stacked) on the motherboard and each one hosts a separate local bus, an SSRAM and an SDRAM for running associated software. The clock frequency of the local bus may reach up to 50 MHz and the AMBA bus up to 30 MHz. Communication of each ARM with the other processors and blocks of the architecture through the AMBA bus is realized by using a FIFO (when is write operation) without any wait cycles, when for reading data is done directly and the processor blocks the bus and waits the necessary cycles to access the target block.

Each “logic module” hosts a large FPGA (XILINX Virtex E 2000, 0.18µm, 6 metal layers), with 500K usable gates and 832 Kb of additional RAM (blockRAM) and built-in clock management circuitry (8 DLLs). There is also available on each logic module an SSRAM to be accessed through a local bus, if required. Additional clock management circuitry exists on the board (external to the FPGA), to generate two separate clocks under the control of logic implemented (eventually) in the FPGA.

The Analog to Digital and Digital to analog conversion, for communicating with the analog front ends of the Rx and Tx respectively, is implemented on a separate board which seats on a dedicated connector for external communications on the “top” of the stack of logic modules. Also the communication with the PCI or Ethernet interface is done through that port.

Figure 6 shows the partitioning of the architecture described in section 4 on the logic modules of the prototyping platform.
6. Verification of the complete system

To verify our system as NIC or AP the following categories of tests will be performed:
1. Functional tests of the OFDM MODEM and then of the NIC and AP blocks
2. ETSI conformance tests of NIC and AP Prototypes

In category 1 the various levels of verification of the target system start from the level of the datapath modem (Tx and Rx), up to the SoC architecture and system level of NIC and Access Point. Here follows the list of the tests:
- Modem Tests (Receiver Block Tests, Transmitter Block Tests, Modem I/F tests, RF-IF I/F tests)
- DMA tests and DLC HW block tests
- Integrated Modem-DMA tests
- PCI controller tests
- Ethernet controller tests

Figure 7 shows the environment for the MODEM verification adopted in the development.

The MODEM tests will be organised as follows:

a. RF Power sweep for all modulation types (BPSK, QPSK, 16-QAM 64-QAM) and coding types without and with frequency offset (up to 40 ppm offset) and without antenna.
b. Same tests as a. but with various antenna setups (various distances, LOS, NLOS).
c. Same tests as a but with the indoor channel models A-E that are given by ETSI (BRAN#8), to be used for Hiperlan2 simulations. These channels cannot be
modelled exactly by Rhode-Schwartz generator as it provides only up to 6 taps models. However they can be approximated by six taps and have tests with these channels models.

The Hiperlan2 or IEEE 802.11a OFDM signal is generated by software in a PC (WinIQOFDM of Rhode-Schwarz). It is then downloaded to Rhode Schwarz signal source. The RF 5 GHz signal that is generated by the instrument is received by the front-end implementation and then is fed to the FPGA prototype platform. The results of the Rx processing are compared to the transmitted data and the performance of the Rx is measured by the SNR and BER figures. Using this measurement set-up the algorithms are tested for their robustness under various impairments (frame synchronization, frequency offset etc.), for all different PHY modes and coding types.

In category 2 the ETSI conformance tests are related to the physical layer and to the protocol stack. The ETSI PHY layer conformance testing concern, carrier frequencies and channelization, RF output power, transmitted unwanted emissions, transmit spectral power masks, modulation accuracy, switching types, receiver sensitivity, receiver spurious emissions, dynamic frequency selection mechanism and transmitter power control.

The SW verification of Hiperlan2 and IEEE 802.11a protocol stack. ETSI conformance tests in MT/AP Prototype concern:

- Critical parts of DLC ETSI conformance tests in MT and AP prototypes
  - Radio link control sublayer (RLC) Access Point related tests
  - Radio link control sublayer Mobile Terminal related tests
  - Error control (EC) Access Point related tests
  - Error control Mobile Terminal related tests
  - Packet based Convergence Layer (PCL) Access Point related tests
  - Packet based Convergence Layer Mobile Terminal related tests

- IEEE 802.3 conformance tests and CardBus conformance tests

The prototyping of the target system is in progress. For the tasks assigned to instruction set processors C code is developed and mapped on ARM7. Execution times for basic tasks of Hiperlan2 DLC layer on the ARM Integrator are presented in Table 1. These results assume an operation frequency of 50 MHz (cycle 20 ns). The code and the data for the tasks are stored in SDRAM memory.

<table>
<thead>
<tr>
<th>Task</th>
<th>Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP-Scheduler</td>
<td>0.5 ms</td>
</tr>
<tr>
<td>AP/MT-TxCL</td>
<td>2.6 ms</td>
</tr>
<tr>
<td>AP/MT-TxBuilder (full frame)</td>
<td>1.8 ms</td>
</tr>
<tr>
<td>AP/MT-TxBuilderCopy (580 bytes – word transfer)</td>
<td>880 µs</td>
</tr>
<tr>
<td>AP/MT-RxDecoder</td>
<td>1.5 ms</td>
</tr>
<tr>
<td>AP/MT-RxCL</td>
<td>2.8 ms</td>
</tr>
</tbody>
</table>

*Table 1: Execution times for basic tasks of Hiperlan2 DLC layer (where AP: Access Point, MT: Mobile Terminal, CL: Convergence Layer, Tx: Transmitter, Rx: Receiver)*

Execution times for the major subtasks of the process of Hiperlan2 MAC sublayer decoding the BCH and FCH frames and controlling the modem (MT-BCH/FCH Decoder Modem Ctrl) are presented in Table 2. These results assume also an operation frequency of 50 MHz (cycle 20 ns). The code and the data for the tasks are stored in a core module SRAM memory of the ARM integrator.
MT-BCH/FCH Decoder Modem Ctrl

<table>
<thead>
<tr>
<th>Subtask</th>
<th>Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialisation Phase (Reset &amp; Config @ slot commands)</td>
<td>1.50 µs</td>
</tr>
<tr>
<td>Synchronisation Phase (BCH_SRCH, Rx_FCH with rpt = 1, Rx_ACH)</td>
<td>3.80 µs</td>
</tr>
<tr>
<td>BCH decoding and BCH CRC checking</td>
<td>12.40 µs</td>
</tr>
<tr>
<td>Decoding of a single IE (UL)</td>
<td>6.14 µs</td>
</tr>
<tr>
<td>Decoding of 3 IEs (2 ULs, 1 DL) including CRC checking &amp; Puncturing</td>
<td>41 µs</td>
</tr>
</tbody>
</table>

*Table 2: Execution times for basic subtasks of the MT-BCH/FCH Decoder Modem Ctrl Hiperlan2 MAC layer process*

The results presented in the tables above (in the ARM Integrator platform) are pessimistic, compared to a realistic SoC implementation. This is due to the overheads introduced by the ARM Integrator platform architecture (FIFOs of the bus interface, SDRAM controller etc.). The modem has been mapped on the core modules (Xilinx FPGAs) of the ARM integrator platform. Two Xilinx XCV2000E-6 are currently used for the realization of the modem. In the first one the frequency and data domain blocks of the receiver are mapped. The total utilization of the first FPGA is 79%. The utilization per resource type for the first logic module is presented in Table 3. The second FPGA includes the transmitter, the time domain blocks of the receiver, the interface to MAC and a slave interface to an AMBA bus. The total utilization of the second FPGA is 83%. The utilization per resource type for the first logic module is presented in Table 4.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOs</td>
<td>87</td>
<td>512</td>
<td>16.99%</td>
</tr>
<tr>
<td>Function Generators</td>
<td>14319</td>
<td>38400</td>
<td>37.29%</td>
</tr>
<tr>
<td>CLB Slices</td>
<td>7160</td>
<td>19200</td>
<td>37.29%</td>
</tr>
<tr>
<td>Dffs or Latches</td>
<td>3271</td>
<td>40812</td>
<td>8.01%</td>
</tr>
</tbody>
</table>

*Table 3: Utilization per resource type for the first FPGA*

<table>
<thead>
<tr>
<th>Resource</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOs</td>
<td>276</td>
<td>512</td>
<td>53.91%</td>
</tr>
<tr>
<td>Function Generators</td>
<td>12617</td>
<td>38400</td>
<td>32.86%</td>
</tr>
<tr>
<td>CLB Slices</td>
<td>6309</td>
<td>19200</td>
<td>32.86%</td>
</tr>
<tr>
<td>Dffs or Latches</td>
<td>5943</td>
<td>40812</td>
<td>14.56%</td>
</tr>
</tbody>
</table>

*Table 4: Utilization per resource type for the second FPGA*

7. Transition from the prototyping board to the final Reconfigurable SoC

There are several compromises made to build the SoC under development on the ARM integrator platform. The most important is the lack of a local bus for the communication between the MODEM and the protocol processor. The protocol ARM in the Integrator prototype platform communicates with the MODEM and the networking ARM through the AMBA bus, sharing the bus bandwidth with those units. That bottleneck is expected to put the real time performance of the system in danger. The ASIC implementation of the design will not suffer from such a restricted bus bandwidth because of the local busses shown in the architecture presented in Figure 5.
Other important features of the final SoC were not implemented in the first prototyping phase in the ARM integrator (clocking strategy, design for testability, debugging circuitry and strategy, etc.). Appropriate synthesis flow should be applied for the ASIC phase combined with the reconfigurable circuitry. Floorplanning for intercommunication between FPGA and fixed logic should be done to minimise the delays. Low power issues for the fixed logic part of the SoC should be applied to achieve a competitive performance.

8. Conclusions
A design story about the development of a reconfigurable System-on-Chip for wireless LANs has been described. The system will realize the Hiperlan2 and IEEE 802.11a wireless LAN systems and functionality both for Mobile Terminals and Access Points. Functionality for operation in outdoor environments for wireless point-to-point links is also targeted. The selected heterogeneous System-on-Chip implementation platform is expected to lead to an optimal trade-off between implementation efficiency and flexibility. Currently the system is under prototyping on the ARM integrator platform and then after its architecture evaluation, the final design will be ported to the final reconfigurable SoC ASIC implementation. The know-how that will be obtained from this reconfigurable implementation will build the guidelines for the future wireless communication products of the company, the products supporting in-the-field upgradability and configurability.

9. References