Prototyping Environment for Software Defined Radio

1. Need for Software Defined Radio

Software Defined Radio (SDR) affords the most flexibility in investigating and prototyping a variety of wireless solutions. The goal of this project is to implement two modules: low-level physical communication layer (the PHY), and the high-level medium access control layer (the MAC) as shown in Figure 1.

![Figure 1. Block diagram of a typical wireless LAN system](image)

The complexity and evolving nature of wireless LAN make a degree of programmability essential for the MAC. Most MACs achieve this by embedding a processor coupled to hardware blocks, such as packet engine or cryptography module. However, it is difficult a-priori to make good decisions as to what areas require hardware acceleration. The ability to run firmware on real hardware in a real network allows design decisions to be based on real data.

Conversely, the PHY standard is stable and well understood, however the PHY is usually a dedicated ASIC with almost no reconfigurability. There is little, if any, scope to update different modules within the PHY. This means that studying different protocols and decoding schemes is impossible.

2. Field Programmable Array SDR

The goal of our Field Programmable Array (FPGA) SDR prototype is to address both the MAC and PHY implementation issues discussed above. The capacity and speed of modern FPGAs offers a realistic prototyping environment. Using PHY Verilog source and MAC functionality implemented with C targeted to a PPC, we have a design/test/debug cycle that is achievable on a time-scale of hours.

3. Project Goals

1. Implement an 802.11 SDR system for both 2.4 GHz and 5.9 GHz with MIMO functionality.
2. Design most (90%) of the MAC on a PPC based system using the C programming language.
3. Prototype a PHY layer in Verilog.
4. Develop an interface to the MAXIM 2829 2.4 GHz/5.9 GHz R/F front end.
4. Hardware

Currently we are working with a DN8000k10PCI FPGA board. A block diagram of this system is shown in figure 2.

1. The DN8000k10PCI has three FPGA chips. This is ideal from our design standpoint since one chip (FPGA A in figure 2) will be the client interface (PCI), another chip will hold the PHY (FPGA B in figure 2) and the third chip will have the MAC (FPGA C in figure 2).
2. FPGA C in figure 2 is a Virtex-4 FX chip, this has an embedded PPC processor that will hold 90% of our MAC.
3. The client interface is PCI (64-bits at 66 MHz),

Nevertheless, the lack of an analog front end interface to this board makes it a prototyping platform for testing base-band only.

5. Future Platform

Once we implement based-band functionality we are planning to migrate the design to the LitePoint based TrueChannel system. This system is a 4 radio platform that supports both 2.4 GHz and 5.9 GHz frequency bands along with support for 20 MHz and 40 MHz bandwidths. It also includes support for all bit rates and modulations included in 802.11 legacy and MIMO product requirements. The LitePoint platform will enable further FPGA development as well as investigation into general processor based SDRs. The evolution of processing power has enabled more and more computationally intensive activities to be performed by multi-core general processors in real time. Our Native SDR (nSDR) is attempting to explore a better partition of dedicated cores for wireless PHY/MAC implementation to achieve the best efficiency in full software solution. This approach leverages the ongoing technological roadmap for steady and predictable increase of computing power. This approach requires rethinking how radios and communications are handled because one is no longer constrained by a single fixed radio standard. The flexible SDR allows multiple standards to be accessed depending on location, mobility, throughput, and other requirements.

Designing the PHY blocks in a HDL like Verilog and the MAC in a hardware independent like C makes the migration between different FPGA platforms possible.

Figure 2. DN8000k10PCI block diagram