Implementation of a wireless ATM transceiver using reconfigurable logic

Gerard J.M. Smit, Paul J.M. Havinga, Marcel van Opzeeland, Remco Poortinga
University of Twente, department of Computer Science, the Netherlands
{smit, havinga }@cs.utwente.nl

ABSTRACT

In this paper we present the design, implementation and realization of a receiver for a wireless nano-cellular ATM network using a small Field Programmable Gate Array. The network is designed for an office environment. The method used for transmitting data, at the rate of 1Mbps, is Differential Phase Shift Keying (DPSK) on a 3 MHz carrier using near-field RF. It uses differential coherent detection to demodulate the received signal. A matched filter combined with a notch filter reduces the bit error rate. A MAC protocol for this network that is suitable for multimedia traffic is implemented in a micro-controller. The system has been demonstrated in a setup where uncompressed video frames (320x200x6) were transmitted at a rate of 4 frames per second.

1 INTRODUCTION

In the Moby Dick project [5] we develop and define the architecture of a new generation of hand-held computers, the so-called *Mobile Digital Companions* (MDC). With a Digital Companion we shall be able to make telephone calls, make payments and we can use it to access all the information we now carry in our briefcases.

Several factors have hampered the advance of the use of electronic information in everyday life. One of the most important has been that the information cannot always be made available where it is needed; electronic information was never very mobile. This is now about to change. Digital wireless telephony can now be used to connect small portable computers to the internet so that, wherever one is, one's data can be accessed and interaction with others is possible. Mobile computing, assisted by wireless networking, is an essential step in making electronic document exchange, electronic communication, and electronic commerce replace at least some portion of their non-electronic counterparts. This development will have to be accompanied by much better infrastructures for securing the privacy of one's personal data and the integrity and authenticity of financial and other transactions.

Wireless networking is of eminent importance and greatly enhances the usability of these portable-computing devices. The combination of networking and mobility will engender new applications and services. Not only does it provide the means for users to stay in touch while on the move and to receive notifications of important events, it also gives people a whole new way to interact with the infrastructure of large public institutions, such as airports, supermarkets, or even whole cities. This interaction can be used for information about services, access to them and transactions with them. Standing in line for ticket or teller windows may become a thing of the past. Instead offices and public places will be equipped with access points, through which hand-held computer users will be able to communicate with the existing infrastructure.

RECONFIGURABLE COMPUTING

We strongly believe a radical new approach has to be taken in order to fulfill the requirements, in terms of processing power and energy consumption, of the Pocket Companion. A *reconfigurable systems-architecture* in combination with a QoS driven operating system is needed that can deal with the inherent dynamics of a mobile system. The system architecture should be flexible and/or reconfigurable in many ways. The concept of a software radio is considered as a possible solution to the problem of ever-increasing number of different wireless infrastructures mobile users encounter roaming the world.

In this paper we describe the design and implementation of a prototype of an ATM cellular network, based on near-field radio, using field programmable components [4]. We will show that reconfigurable devices are favorable in the design of a wireless interface. The prototype board consists of a simple analog receiver with an AD converter, that passes 8 bits samples at 12 Msamples per second to a Xilinx FPGA. Connected to the FPGA is a micro-controller. The micro-controller handles a real-time MAC protocol and performs the interface to the host. The programmable components allow us to experiment in real-life with several implementations of the transceiver and MAC protocol.

NANO-CELLULAR SYSTEM

Our initial target is nano-cellular systems in which cells have the size of one office room or the size of a meeting table or desk. In this system each office contains one or more base stations connected via an existing wired ATM infrastructure (see Figure 1). The physical layer transfers ATM cells only.

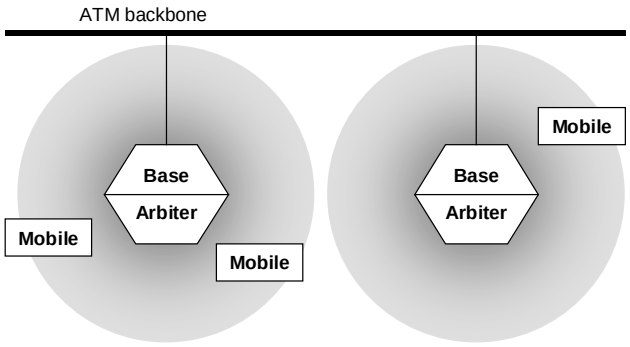

FIGURE 1: GEOGRAPHICAL STRUCTURE OF THE WIRELESS NETWORK

A nano-cellular system has a number of advantages:

1. *Localization* -- Knowing where people are is the key principle for building location-aware applications. Sample applications are: tracking people or equipment within a building, location oriented paging, switching telephones, admission control, security, personal diary, etc. This will give new opportunities for issues like privacy and security. This is only possible when the location boundaries of a cell are well defined. This can be obtained when the transmission does not pass through walls (very high-frequency RF of IR) or has a rapid spatial decay of field strength (near-field RF coupling).
2. *High capacity* -- Small cells imply that users share the full capacity of distinct cells only with few other users, thereby attaining a high bandwidth density (Mbps/m2). The total aggregate bandwidth of an entire office building is the number of cells times the bandwidth per cell.
3. *Low transmission power* -- Because the distance between base stations and mobiles is small in a nano-cellular network, low transmission power is required.

NEAR-FIELD RADIO

For system simplicity, it is desirable to have a minimum set of communication frequencies. If the range of each cell is limited, frequency re-use is possible. The ideal system would have one common frequency, but would require a high degree of isolation between cells or an adequate access mechanism. For several reasons it is difficult to obtain a well-defined cell RF boundary in buildings. In part this is due to gradual spatial decay of field strength (1/r). This slow decay limits the packing density of cells on a common frequency. Furthermore, metal objects, such as office furniture and wall framing, cause reflections of the propagating radio waves, leading to distortions in the cell shapes. Finally, the reflections result in standing wave patterns with quite large peak-to-null ratios over distances of only a few centimeters. In practice this means that for base stations operating on the same frequency, very slight movements of a mobile receiver can cause the communication channel to behave in an unpredictable manner.

In order to combine the advantage of small, well-defined cells and to overcome the problems mentioned above, we used a cellular infrastructure based on near-field RF coupling [2]. In a conventional RF system, the receiving antenna is in the far-field region of the transmitting antenna. This is generally desirable when the range is to be maximized. For UHF and microwave transmitters, the far-field region is well established beyond several centimeters from the transmitting antenna. In this region, EM field strength decreases with distance as $1/r$, where r is the distance from the transmitting antenna. However, all antennas exhibit a near-field region as well. Within this region, the main energy is in the magnetic field and its field strength follows a much more precipitous rate of decay. Exact definition of the near field region will vary with antenna geometry, but for most purposes it will approximately defined by ($\lambda < r < \lambda/2\pi$), where λ is the free space wavelength of the signal. Near-field coupling exhibits a more rapid spatial decay of field strength, thereby permitting close packing of cells, all of which operate on a common carrier frequency. The carrier frequency is identical for all cells. The system was designed for a network cell radius of approx. 5 meters.

The Network cell is build up by the magnetic field induced by the transmitter into the loop antenna of the receiver. The special transmission technique used by NEDAP N.V produces only a near magnetic field and a negligible electric field. The advantage of the near field effect is that the magnetic flux density decreases by the third power as a function of the distance. Whereas the electromagnetic field density only decreases by the second power as a function of the distance. This makes the near field suitable to be used in an office environment. The effect is demonstrated in [2]

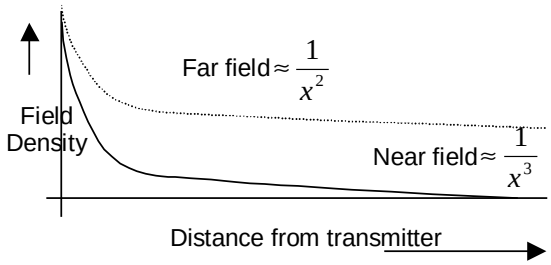

FIGURE 2: THE NEAR FIELD COMPARED TO THE FAR FIELD

Energy transfer in the near-field region is accomplished by magnetic coupling between source and sink, rather than by propagation from transmitter to a receiver. For example, a loop antenna connected to a receiver in the near-field region will extract energy from the magnetic field components, and hence couples to the transmitter in a

similar way as a transformer's secondary is coupled to its primary.

In order to use the near-field region, one must choose a signal wavelength appropriate for the desired cell size. For a carrier frequency (between 1 and 10 MHz) the cell size is approximately that of a single office (cell radius of 3 - 5 meter).

2 DESIGN

The system is defined by a number of parameters. They were defined during the design and realization of the analog part of the receiver. Nedap N.V. has built the analog receiver and the transmitter. This company uses near-field radio in identification products. There are several constraints concerning the design of the receiver. This is due to the transmitter, the analog receiver, and the prototype board. The modulation method used is Differential Phase Shift Keying. The bits in an ATM Cell are modulated on a 3 MHz carrier using phase shift keying. In the realized transmitter the carrier is not synchronized with the modulation bitstream. The consequences of some of these system variables are discussed in the following sections. Table 1 summarizes the constraints.

Modulation method	DPSK
Sample frequency ADC	12 Msamples/second
Cell radius	6 meters
FPGA clock	12 Mcycles/second
Data format	ATM Cells (53 bytes)
Carrier frequency	3 MHz.
Bit frequency	1 Mbit/second

TABLE 1: SUMMARY OF PHYSICAL PARAMETERS.

The modulation method used is Differential Phase Shift Keying. The phase of the carrier is shifted by $+\pi$ radians if the modulating bit does not equal the preceding bit. The phase of the carrier is not shifted if the modulating bit equals the preceding bit. The principle is demonstrated in Figure 3.

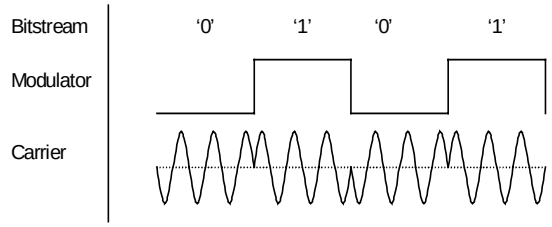

FIGURE 3: DATA FROM AN MODULATING CARRIER

In the realized transmitter the carrier is not synchronized with the modulation bit stream, which makes it unsuitable for coherent detection. The modulated carrier is received by the analog part of the receiver. After filtering the carrier is sampled by an ADC (Analog to Digital Converter) at 12 Msamples/second. This results in 12 samples per bit.

This paper describes the digital part of the near-field radio receiver. Figure 4 shows the main modules of the digital receiver. The modulated carrier is received by the analog part of the receiver. After filtering the carrier is sampled by an AD converter at 12 Ms/sec.

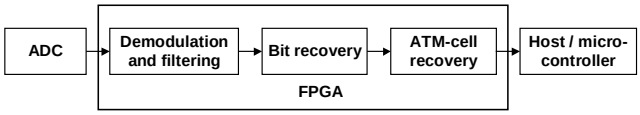

FIGURE 4: SCHEMATIC PATH FROM ADC TO HOST

First the signal of the ADC is demodulated and filtered and after that the bits and bytes are recovered. The design was simulated using Matlab and then implemented in a Xilinx XC4005 FPGA using VHDL. The objective of the simulation was to make a solid choice between the different digital filters and to validate the demodulation method.

Because of non-linear effects (e.g. clipping) in the analog part of the receiver, it was not realistic to use theoretical generated data. Therefor the data stimulus to Matlab was gathered from the actual data on the AD converter. A digitizer was implemented in the FPGA that recorded data of the AD and stored them in a buffer. This data was read by a PC and used as a stimulus for Matlab. The signal applied to the transmitter was a 100 kHz square wave, resulting in a '5 bits high, 5 bits low' signal.

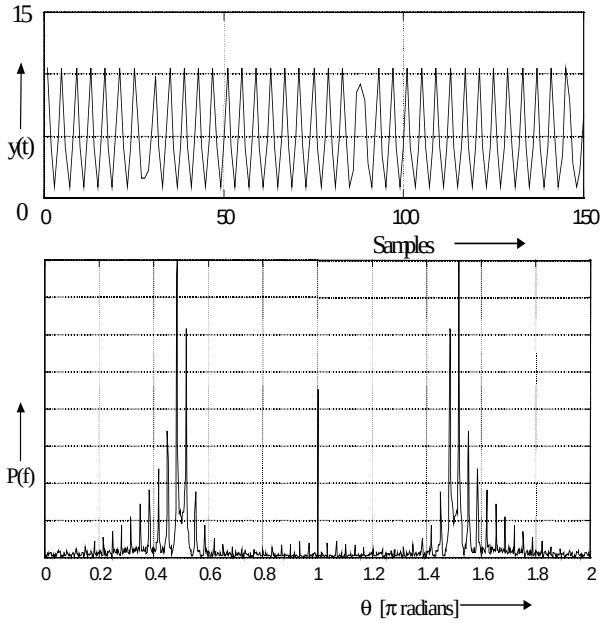

FIGURE 5: SIGNAL FROM THE ADC IN THE TIME DOMAIN AND ITS POWER SPECTRUM

The signal from the ADC in the time and frequency domain is shown in Figure 5.

Demodulation

The DPSK demodulation is done by differential coherent detection [1]. In this approach the received signal is delayed for one bit-time and then multiplied by the original signal. The principle is demonstrated in Figure 6.

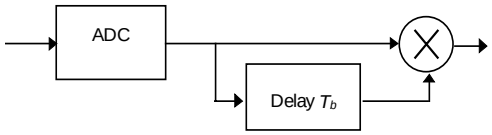

FIGURE 6 SAMPLING COMBINED WITH DELAY&MULTIPLY FILTER

The multiplication of two signals that only differ in phase will result in a new signal with a doubled frequency and an offset. The amplitude of this offset component is determined by the phase difference (in our case π radians if the bit changes).

$$y(t) = \frac{A^2}{2}\left[\cos(2\omega t + \varphi) + \cos(\varphi)\right]$$

The result of the delay and multiply operation is given in Figure 7. As a result of this operation, a phase shift is transformed to a negative DC offset.

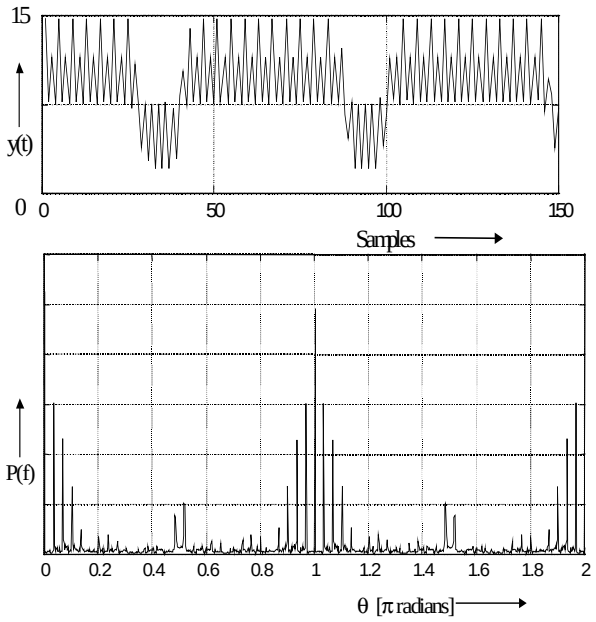

FIGURE 7: SIGNAL AFTER THE MULTIPLIER IN THE TIME DOMAIN AND ITS POWER SPECTRUM

There are however a few boundary conditions for the multiplier. First, two nibbles (4 bits) are used as input and secondly it produces one nibble as output. The reason is that only 4 bits are significant in the ADC signal and nibbles are more suitable for an implementation in a FPGA. Finally we work with a virtual zero at half the input range. The output is in the 2-complement form. The multiplier function is defined by:

$$y_n(x_n, x_{n-k}) = round\left\{\frac{(x_n - 7.5)\cdot(x_{n-k} - 7.5)}{8}\right\}$$

The zero point was set to $x = 7.5$. The division by 8 is needed to obtain an integer $y \in [-7,7]$.

The multiplier is a combinatorial function that can be realized with a lookup table. The input data defines the address for the ROM. The output contains the result of the four highest bits of the multiplication. The data in the lookup table can also deal with the conversion to 2-complement data. The non-delayed signal data nibble forms the lower nibble of the address in the lookup table and the delayed nibble forms the higher nibble of the address in the lookup table.

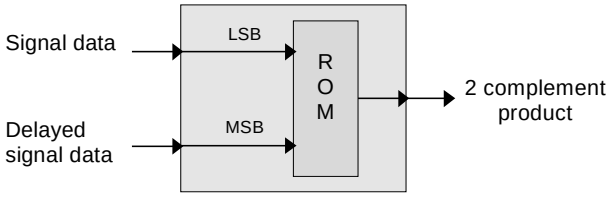

FIGURE 8: IMPLEMENTATION MULTIPLIER

The data in the ROM lookup table has been calculated using the following program:

```
Zero=7.5
 for i=0:15
  for j=0:15
   Table(j+i*16)=round((((i-Zero)*
            (j-Zero))/8));
 end;
end;
```

Filtering

Multiplication of two signals in time causes sum and difference frequencies in the spectrum of the resulting signal. The power spectrum of the signal shows a lobe at π radians, which is an undesired frequency product. These frequencies are filtered with a *notch filter* and subsequently with a *matched filter.*

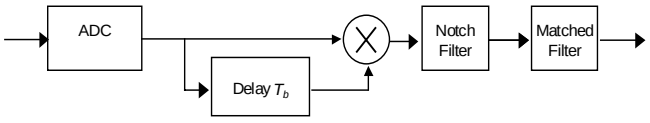

FIGURE 9: SAMPLING COMBINED WITH THE DELAY&MULTIPLY

The mathematical expression for the notch filter in the time domain becomes:

$$y(k) = floor\left(\frac{x(k) + x(k-1)}{2}\right)$$

Because the notch frequency f_c is exactly half the sample frequency f_s the notch filter reduces to a frequency reject filter. This is a simple first order filter so the equation reduces to:

$$H\left(e^{j\cdot\theta}\right) = h_0 + h_1 \cdot e^{j\cdot\theta}$$

The floor operation rounds the result of the calculation to the closest lower integer. The transfer characteristic of the equation is plotted in Figure 10.

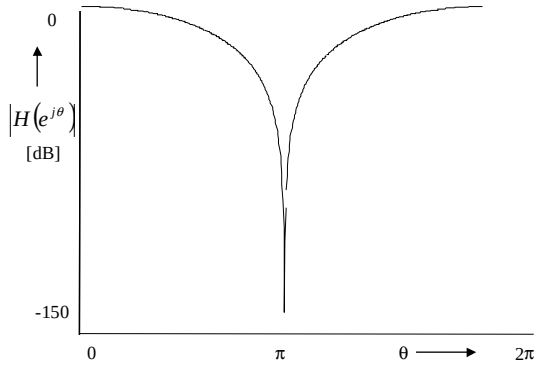

FIGURE 10 :TRANSFER CHARACTERISTIC OF THE NOTCH FILTER

The effect of the *matched filter* is that noise is filtered in the parts of the spectrum where no data is present, i.e. it shapes the amplitude response to that of the data, while the notch filter attenuates the higher frequencies. An important reason for using these filters is that they are easy to implement. Other discrete filters can achieve a better frequency response, but their filter coefficients are not integers. The implementation of a floating-point multiplication in a small FPGA is not feasible. The transfer characteristic of the combined filter is shown in Figure 11.

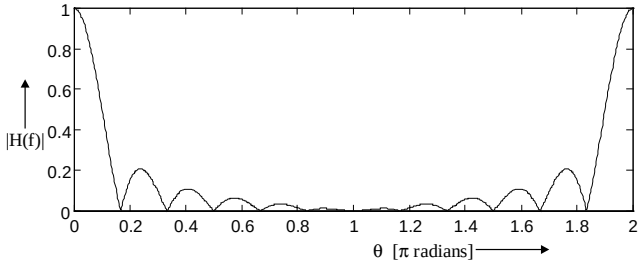

FIGURE 11: TRANSFER CHARACTERISTIC OF THE COMBINED NOTCH AND MATCHED FILTER

The result of the notch filter in combination with the matched filter on the actual signal is shown in Figure 12.

In the time domain every time there is a phase change in the modulated signal we get a negative pulse. These pulses can be detected by a zero-crossing detector.

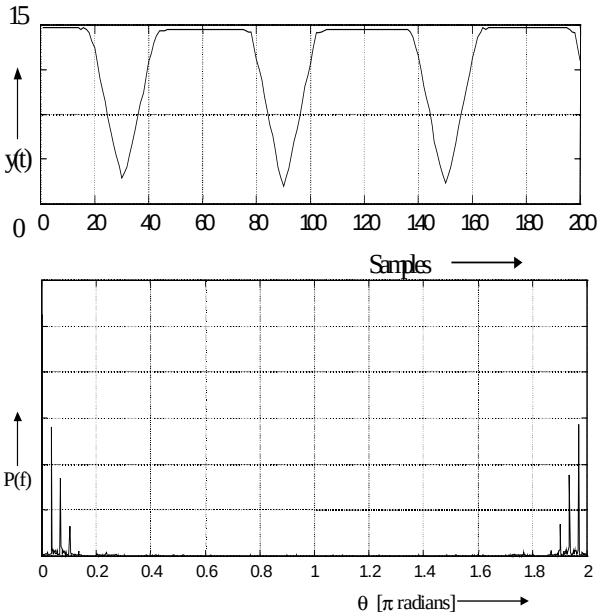

FIGURE 12: SIGNAL AFTER THE COMBINED FILTER IN THE TIME DOMAIN AND ITS POWER SPECTRUM

ATM-cell recovery

The output of the filter has to be converted to ATM cells.

An important issue is the synchronization of the receiver with an arriving ATM Cell. Transmitting synchronization bits followed by a preamble performs this task. The synchronization bits are used to synchronize the bit synchronizer with the data stream. After the synchronization bits, the preamble is sent. A *Serial to Parallel Converter* with *preamble detector* is used to detect the preamble so the receiver interprets the next bit that arrives after detection as the first bit from an ATM Cell. In the time between the ATM Cells, noise is being received and also converted to bits. The probability that noise bits form a preamble is:

$$P_{\text{preamble detected}}(\text{noise}) = 2^{\text{bits in preamble}}$$

The design uses a preamble of 16 bits. Statistically a false preamble could be detected 15 times per second. Measurements in real time show that the number of false detected preambles is much smaller. This is mainly caused by the fact that the input has a small offset, resulting in a stream of mostly logical '1's if only noise is received.

ATM Cells must be demodulated from the carrier data. So knowledge of some relevant ATM Cell parameters is needed. First of all, the length of an ATM Cell is 53 bytes. Five bytes are used for the header and 48 bytes are used for the payload. The 5th byte in the header is used for error control of the header data, so the digital receiver could use this byte to validate a received ATM Cell header.

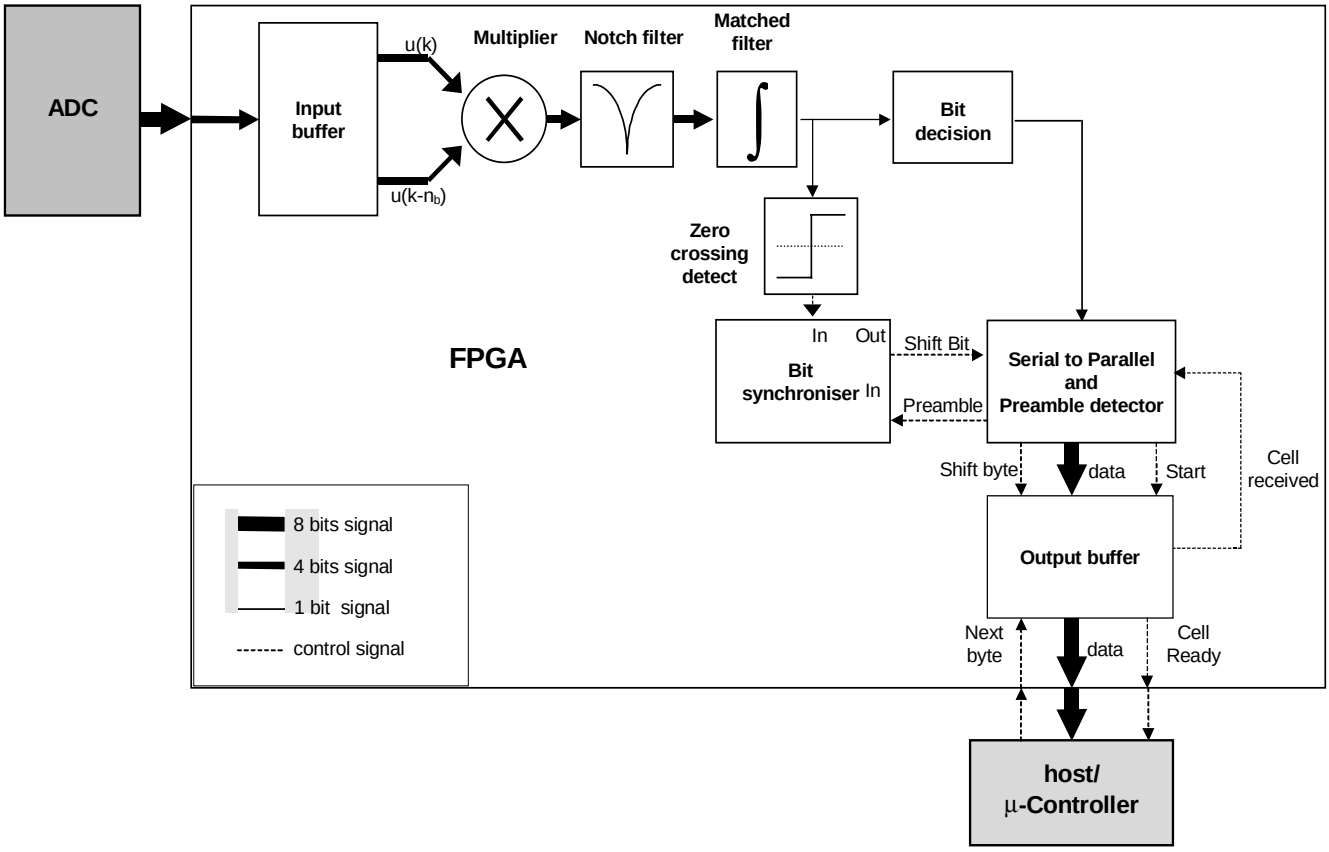

FIGURE 13: GENERAL DESIGN OF THE ATM RECEIVER

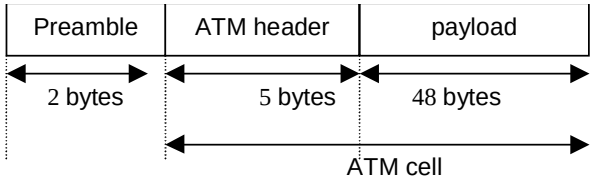

Figure 14: An ATM cell preceded by a preamble

In detail the ATM cell recovery works as follows: A zero detect circuit detects every zero crossing of the input signal. The bit synchronizer uses this to synchronize the receiver to the incoming bit stream. It operates in two modes. In the first mode the bit synchronizer synchronizes every zero crossing to allow the detection of the start of a preamble. The sampled bits are converted to bytes and in parallel scanned for a preamble, indicating the start of an ATM cell. The second mode is entered when a preamble is detected. Since the bit synchronizer is now assumed to be synchronized with the incoming bits, the bits can be sampled at the top of the signal from the filter. Synchronization while receiving an ATM cell is not necessary because the drift between the sending crystal and the receiving crystal is small.

Received bytes are shifted to an output buffer for two ATM cells. This buffer stores the received ATM cell and sends the previously received ATM cell asynchronously to the micro-controller or host.

3 REALIZATION

The digital receiver has been implemented using an FPGA. Figure 14 gives an overview of the realization.

The multiplier is implemented as a ROM that also converts the data to 2-complement. The notch filter has been implemented by using two 4 bits input latches and an adder. The matched filter has a depth of 4 bits. A straightforward implementation of a matched filter would require the addition of twelve separate nibbles every clock cycle, which would take a lot of the FPGA resources. A cheaper implementation is to determine the difference between the incoming and outgoing samples. This can be done by using a delay of twelve clock cycles, subtracting the delayed samples from the non delayed samples and integrating them continuously. A zero crossing is analogous to a change of the sign bit in two's complement, therefore a zero crossing can be detected by comparing the sign bits of two successive numbers. The bit synchronizer is essentially a state machine with twelve states. The serial to parallel converter is a basically an 8 bits shift register and a 3-bits counter. A comparator is used to detect a preamble. The output buffer is implemented as a ping-pong buffer of two RAM buffers, each capable of storing 53 bytes. One buffer is used for storing an incoming ATM cell while the other is sending a previously received ATM

cell to the host/micro-controller. A control unit determines dynamically which RAM buffer to use.

The designs were synthesized using a VHDL synthesis tool. The final design fits in a small FPGA (Xilinx XC4005-6). The design was tested with a simple video application. The transmitter was fed with ATM cells coming from a frame grabber. The frame grabber was made of a micro-controller and a *Connectix QuickCam* camera. The ATM cells were received by a PC acting as host, converted to bitmaps (320x200x6), and finally converted to JPEG images. These JPEG images formed a movie (with a frame rate approximately 4 frames per second) on the Internet. One of the images, showing the prototype receiver, is displayed in Figure 15.

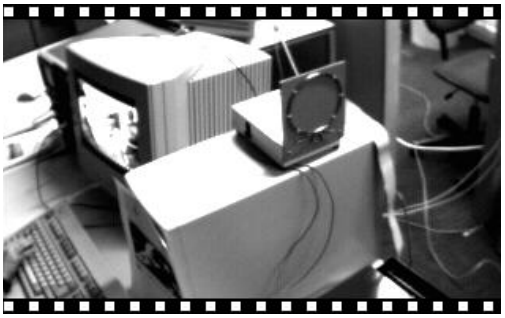

FIGURE 15: DEMONSTRATION WIRELESS VIDEO CONNECTION

4 ARCHITECTURE OF THE R-TDMA PROTOCOL

For this nano-cellular system we have designed a simple MAC protocol called Request-TDMA (R-TDMA). This protocol is optimized for the situation and can be implemented in a small micro-controller (PIC 16C74). R-TDMA is intended for a wireless ATM network in an office environment.

With Request TDMA (R-TDMA) time is divided into fixed size frames. Each frame in turn is divided into fixed size slots. The length of one slot is equal to the transmission time of one ATM cell, including powering up the transmitter and synchronization. A base station, called the arbiter, administers one geographical cell. In this cell mobiles can request connections from the arbiter. The arbiter controls the MAC protocol in its geographical network cell. This is visualized in Figure 16.

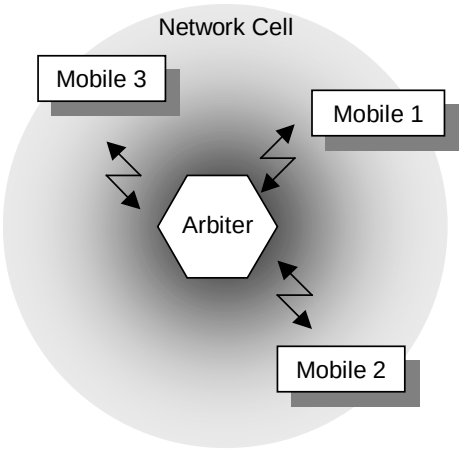

FIGURE 16: AN EXAMPLE OF A TYPICAL GEOGRAPHICAL CELL.

The arbiter divides the available slots over the mobiles present in the geographical cell and it listens for new connection setup requests. The sending of a CTS (Clear To Send) cell by the base-station marks the start of a frame. The CTS cell contains information about every open connection and enables the mobiles to synchronize and determine if and when they are allowed to send data. A mobile only sends data if it has received a permission to send data, the only exception being setting up a connection. A special slot (Contention Cell) is reserved for connection setup. In this Contention Cell (CC), mobiles that want to setup a new connection transmit a Setup Request. This is the only place where collisions can occur because the arbiter assigns all the other slots.

There are two possible kinds of connections that can be setup, one with a bandwidth guarantee (real-time) and one without such a guarantee (non-real-time). At connection setup real-time connections can specify the maximum number of cells they want to send per frame. If the connection request is granted, the specified number of cells per frame requested are guaranteed. Non-real-time connections are allocated the slots that are left after all the real-time connections are served.

Real-time connections are useful for multimedia applications like video or audio. However, the kind of traffic generated by multimedia applications is usually of a bursty nature. This could lead to applications requesting the peak rate they expect to generate, resulting in the loss of bandwidth if they generate less traffic. To overcome this problem, real-time connections have to specify the required number of cells for every frame. This is done in the so-called Request To Send (RTS) slots. By using this technique unneeded cells can be reused for other connections. Because the size of an RTS cell can be kept small, several of them can be fitted into a standard slot. These special small-sized slots for RTS Cells are called mini-slots. The structure of an R-TDMA frame is shown in Figure 17.

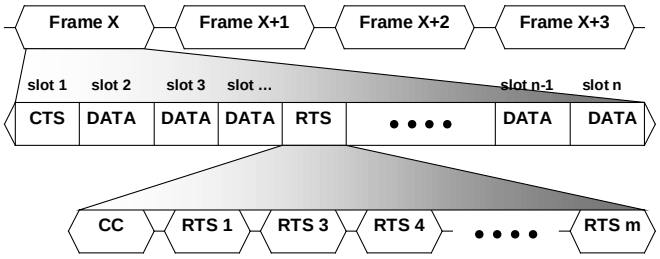

FIGURE 17: THE R-TDMA FRAME FORMAT

Mobiles and arbiters

As stated before geographical cells are administered by the arbiters. The base-station consists of a wireless interface in combination with an arbiter. The only function of the arbiter is to control the MAC protocol. The base-station is connected to an ATM backbone. Mobiles can access the ATM network via the base-station. This datapath is called the uplink. The uplink communication introduces new constraints, like cell interference, that were not taken into account in the design of the R-TDMA protocol. Data transfer from the ATM backbone to the mobiles is called downlink. The base-station forms the center of the network cell. The mobiles can detect if they are in a geographical cell by listening for CTS cells. Every arbiter transmits these, whether there are open connections or not.

As stated before the tasks for the arbiter and mobile are not the same. Because the arbiter administers the traffic in the geographical cell, it has to do most of the 'controlling' part, but it is not interested in the contents of the actual data in the ATM Cells transmitted by the mobiles. The 'controlling' part of the mobiles is less complex than that of the arbiter, they do not have to divide the available cells amongst the present connections. The different tasks of mobile and arbiter are summarized in Table 2.

Arbiter	Mobile
Listen for connection requests	Asking for a new connection
Listen for changes in data rate	Changing the requested data rate for a connection
Distribute cells amongst open connections	
Construct a CTS cell	
Transmit a CTS cell	Listen for the CTS cell
Listen for close connection requests	Transmit a close connection request
	Transmitting an ATM cell at the right time

TABLE 2: TASKS OF MOBILE AND ARBITER.

Signaling

Signaling is used to control the data stream of the virtual connections. These virtual connections need to be setup and controlled during the connection. A mobile can, for example, request a change in the allocated timeslots. Finally, a connection must be closed if it is not used anymore. These functions are served by the signaling part of the MAC protocol.

The MAC protocol needs a few internal procedures to provide the MAC service to the Data Link Layer. They are the connection setup, connection maintenance and connection close procedures. A successful connection needs to be setup before data can be send over the virtual connection. The signaling protocol uses two types of timeslots:

- Request to send (RTS timeslot)
- Clear to Send (CTS timeslot)

The formats of these timeslots are not the same. The CTS and Data Cells are ATM Cells, meaning that their length is 53 bytes. The CTS cell, however, uses another header than the header format of the ATM standard. The length of an RTS Cell is only four bytes. The mobile uses the RTS Cells to make a request. The arbiter uses CTS Cells to grant or deny a request. An example of the operation of signaling in the MAC protocol is visualized in *Figure 18.* It shows how the connection setup and the connection close take place.

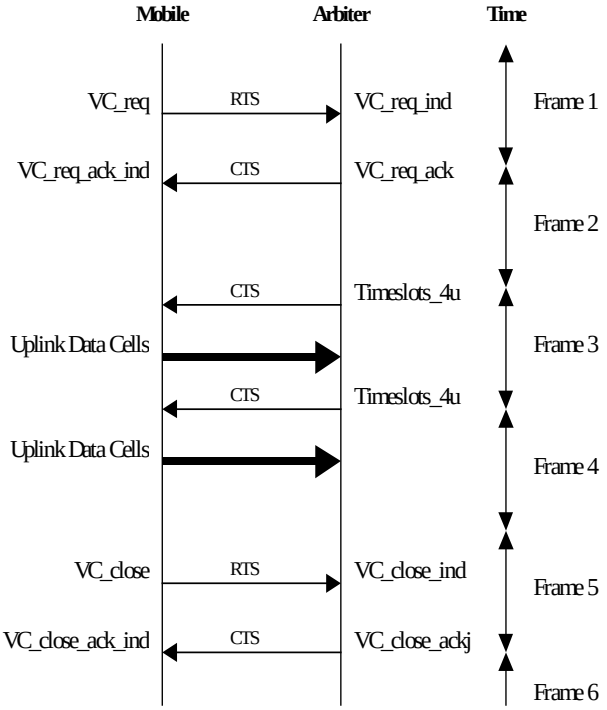

Figure 18: An example of signaling in the MAC protocol.

Setting up a connection

As stated before, time is divided into fixed-size frames, consisting of a fixed number of timeslots, some of which are reserved for signaling purposes. The typical procedure for setting up a connection is as follows (from the mobile's point of view):

1. A mobile enters a cell, it notices this because it receives a CTS cell.
2. The mobile wants to send some data.
3. The mobile sends a connection request in the contention cell (CC), in this request it specifies whether the requested connection has to be real-time or non-real-time. If it is a real-time connection it specifies the requested data rate.
4. Now the mobile waits for the next CTS cell and determines if the request is received and granted by looking at the last byte in the CTS cell.
5. If the request is not received (indicated by the absence of a response from the arbiter), there was probably another mobile trying to set up a connection at the same time, in this case the mobile should back off for some time and try again later.
6. If the request is rejected, the mobile can determine the cause. If the requested data rate is too high, it can try again with a lower rate. If there are too many open connections, the mobile can wait a while and try again.
7. If the request is granted the mobile can send its data at the time specified in the CTS slot. If it is a real-time connection, the mobile has to keep sending RTS cells to inform the arbiter of any changes in the data-rate needed.
8. After the mobile transmitted all the data it wanted to, it sends a close connection request. These requests are never refused.

From the arbiter's point of view the procedure is as follows:

1. The arbiter listens for connection setup requests in the CC timeslot.
2. If it receives such a request, the arbiter checks if there is still room for a new connection. If there are too many open connections at the time it rejects the request indicating the cause of the refusal.
3. If there is still room for a new connection, it determines the kind of connection asked (real-time or non-real-time).
4. If the wanted connection is real-time, the arbiter checks if the requested data rate is available. If not, it rejects the request, indicating that the available capacity is too small for the requested connection.
5. If the request is granted, the arbiter responds with a 'connection granted' (in the CTS cell) and indicates the number of the connection.
6. In the time slots reserved for RTS cells, the arbiter listens for RTS cells. These are used for changing the requested data rate in the case of real-time connections and for the closing of a connection. If the arbiter receives a close connection request it removes the connection for the list. Close connection requests are not confirmed.

5 CONCLUSIONS AND RECOMMENDATIONS

The design of the ATM receiver is implemented with programmable off-the-shelf components. The use of programmable components was productive and gave several advantages during the design phase, but also in real operation as it is able to adapt its behavior to its environment. First, in the design phase we could easily gather actual data from the AD converter from the analog radio using the same hardware and use this data as stimulus for simulations. Then we were able to incrementally implement the system and test various parts of the system. Third, with the same hardware we are able to implement and experiment with various modulation and filtering techniques.

The ATM receiver is designed using VHDL and realized with Xilinx FPGA technologies (XC4005).

The Realization of the ATM receiver was done in several steps. With every step a new part of the design was added and tested (and, if needed, redesigned). This iterative process continued till the design was complete.

The final design fits in an XC4005-6 FPGA from Xilinx as can be seen from the report file (Figure 19) generated by PPR (part of the Xilinx tools).

```
PPR RESULTS FOR DESIGN ATM_RECEIVER

Device Utilization
------------------

Partitioned Design Utilization Using Part 4005PC84-6

-------------------------------------------------------
                          No. Used   Max Available   %Used
-------------------------------------------------------

  Occupied CLBs                193         196         98%
  Bonded I/O Pins              16          61          26%
  F and G Function Generators (*)  296     392         75%
  H  Function Generators       99          196         50%
  CLB Flip Flops               168         392         42%
  IOB Input Flip Flops         4           112         3%
  IOB Output Flip Flops        0           112         0%
  3-State Buffers              0           448         0%
  3-State Half Longlines       0           56          0%
  Edge Decode Inputs           0           168         0%
  Edge Decode Half Longlines   0           32          0%
  CLB Fast Carry Logic         14          196         7%

  (*) If RAM/ROM elements are present in the design, this count
  includes the function generators used for them. A 16x1 memory
     uses 1 function generator; a 32x1 uses two.
```

FIGURE 19: DEVICE UTILISATION REPORT

Another aspect of the design is the timing. In the FPGA signals experience a certain delay. If this delay exceeds a certain limit, determined by the device speed, it will not work reliably. In the case of the XC4005 running at 12 MHz used for this design, the maximum delay is approximately 80 ns. The maximum delay in a design is also mentioned in the report file. It is below the limit imposed by the device and the clock speed (Figure 20).

We implemented a new MAC protocol called Request-TDMA (R-TDMA) [3] for multimedia traffic in a nano cellular network. The protocol supports a mixture of real-time and non-real-time communication with dynamic bandwidth allocation and communication priorities. The startup-time of the transmitter causes a decrease in the transmission capacity of the MAC layer. Reducing the startup-time would increase the transmission capacity of the MAC layer significantly. The MAC protocol easily fits in a small micro-controller.

```
Xact Performance Summary
    ------------------------

    Parttype Used      : 4005PC84

    Speed Grade        : -6


                        End-
      Limit   Actual   Points
       (ns)   * (ns)  Missed   Specification
      ------  ------  -------  -------------
      <auto>    51.3    0/299  DEFAULT_FROM_FFS_
                               TO_FFS=FROM:ffs:TO:ffs
      <auto>    25.0    0/136  DEFAULT_FROM_PADS_
                               TO_FFS=FROM:pads:TO:ffs
      <auto>    51.8    0/9    DEFAULT_FROM_FFS_
                               TO_PADS=FROM:ffs:TO:pads
```

FIGURE 20: TIMING REPORT OF THE ATM RECEIVER

The complete system has been demonstrated with a simple video application. Measurements showed that the BER of the system is less than 1E-6, and that the cells have sharp boundaries.

6 ACKNOWLEDGEMENTS

We would like to thank all persons who contributed to this project, in particular Harry Kip of Nedap N.V., Wim van Etten and Jan Geerlings of the University of Twente.

7 REFERENCES

[1] Carlson A.B.: "*Communication systems*", 3rd edition, ISBN 0-07-009960-X, Mc Graw-Hill, New York, 1986.

[2] Demers A., Elrod S., Kantarjiev C., Richley E.: "A nano-cellular local area network using near-file RF coupling", *Proceedings of Virginia Tech's Fourth Symposium on Wireless Personal Communication*, pp. 10.1-10.16, 1994

[3] Linnenbank, G.R.J. et al.: "A request-TDMA multiple-access scheme for wireless multimedia networks", *Proceedings MoMuC-3*, 1996.

[4] M. van Opzeeland, R. Poortinga: "Design and realization of a wireless ATM-network", *Ms. thesis University of Twente*, 1997.

[5] Smit G.J.M., et al: "Overview of the Moby Dick project", *Euromicro Summer School on Mobile Computing '98*", pp. 159-168, Oulu, Finland, August 1998.