Differential PSK Detector ASIC Design for Direct Sequence Spread Spectrum Radio

Henrik Olson, Daniel Kerek, Bengt Oelmann and Hannu Tenhunen

Electronic System Design Laboratory
Royal Institute of Technology, Stockholm, Sweden
Email: henrik@ele.kth.se

Abstract—**In this paper we present an optimized architecture for differential PSK detection which effectively minimizes the required logic complexity for ASIC implementation. This complexity reduction is obtained by doing intelligent truncation of the detector input words and by utilizing a bit-serial/word-parallel structure. We also show that this can be done without degrading the performance in terms of bit error rate. The structure has been implemented in a direct sequence spread spectrum transceiver circuit, and preliminary measurement results are presented.**

I. Introduction

A. Background

The design of flexible and efficient future high data rate mobile communication systems such as e.g. the Walkstation concept [1], is a major challenge due to the inherent complexity of underlying algorithms and to the harsh radio environment in which these systems will operate. Even though VLSI technology continuously allows increasingly complex systems to be integrated into single chip solutions, simplifications of the existing signal processing algorithms are needed to allow implementation.

In this work, we present our CMOS standard cell design of a digital transceiver for a direct sequence spread spectrum (DS SS) radio targeted for up to 2 Mbit/s cellular data communication, see figure 1. We will concentrate on the development of the detection unit, which is based on differential phase shift keying. A top-down design flow has been employed which allowed us to early identify large system-related complexity reductions and at a later stage do the implementation technique related savings.

Most of the design which now has been implemented in an ASIC is based on our earlier work, which is reported in references 2 to 5. In this paper we report our new results on the complexity optimized detection scheme and its implementation architecture.

B. Phase Shift Keying

Phase shift keying (PSK) is a since long well known modulation method for digital radio applications. Confining ourselves to binary and quadrature PSK, this linear modulation

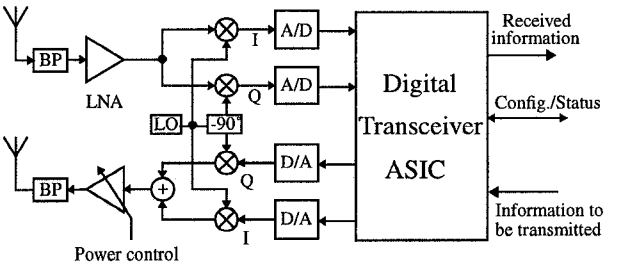

Figure 1. The radio including the analog frontend.

scheme allows optimal detection in the sense that with a fixed signalling bandwidth, it gives the best performance in terms of probability of error as a function of the signal-to-noise ratio E_b/N_0 compared to all memory-less, uncoded digital modulation methods. This requires completely coherent detection, i.e. very accurate phase- and frequency recovery, in the receiver; the detector, on the other hand, then becomes quite simple. Figure 2, where complex notation is used for simplicity, shows a PSK system (the dashed box disregarded), which is coherent if $\theta(t) \equiv 0$. The binary information $b^{(k')}$ is mapped to a binary or quadrature phase, φ_k, according to table 1 or 2. The MF block is a receiver filter which is matched to the transmitted pulse.

Another well-known but sub-optimal detector for PSK is the differential-PSK detector, which does not require coherent demodulation of the carrier-modulated message signal. It is based on that the PSK symbols $u^{(k)}$ are differentially encoded

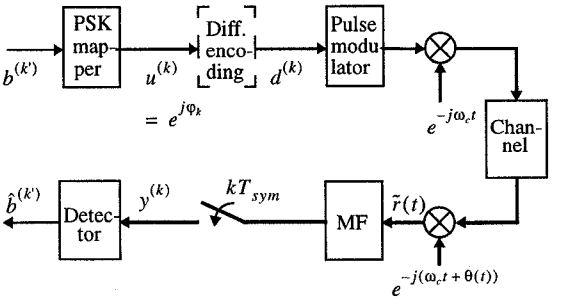

Figure 2. A phase shift keying communication link, with or without differential coding. Thick lines represent complex signals.

0-7803-3302-0/96 $5.00 © 1996 IEEE

Table 1: BPSK

$b^{(k')}$	φ_k
'0'	0
'1'	π

Table 2: QPSK

$b^{(k')}b^{(k'+1)}$	φ_k
'00'	0
'01'	$\pi/2$
'11'	π
'10'	$-\pi/2$

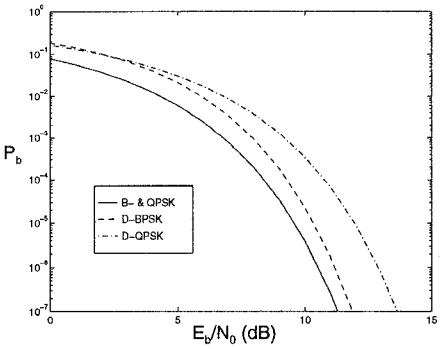

Figure 3. The bit error probability versus signal-to-noise ratio for different PSK detection schemes.

before modulation in the transmitter:

$$d^{(k)} = e^{j\psi_k} = u^{(k)} \cdot d^{(k-1)} = e^{j(\varphi_k + \psi_{k-1})} \quad (1)$$

which is represented by the dashed block in figure 2.

If the transmission medium is modeled as an additive white Gaussian noise (AWGN) channel, the k'th sampled baseband symbol $y^{(k)}$ will have the following appearance:

$$y^{(k)} = y_i^{(k)} + jy_q^{(k)} = \alpha\sqrt{E_b} \cdot d^{(k)} e^{j\theta_k} + n^{(k)} \quad (2)$$

where α is the channel gain factor, E_b is the transmitted energy per bit, θ_k is an unknown phase, and $n^{(k)}$ is a complex Gaussian noise term.

Due to the differential coding, the original phase information, φ_k, can then be retrieved by comparing the phases of two consecutive symbol samples or by removing the unknown channel phase θ_k by e.g. a complex multiplication:

$$\begin{aligned} v^{(k)} &= y^{(k)} \times \overline{y^{(k-1)}} = (\alpha\sqrt{E_b} \cdot d^{(k)} e^{j\theta_k} + n^{(k)}) \times \\ &\times (\alpha\sqrt{E_b} \cdot \overline{d^{(k-1)}} e^{-j\theta_{k-1}} + \overline{n^{(k-1)}}) \\ &= \alpha^2 E_b \cdot e^{j(\varphi_k + \psi_{k-1})} \cdot e^{-j\psi_{k-1}} \cdot e^{j\Delta\theta_k} + \text{noise} \\ &= \alpha^2 E_b \cdot e^{j(\varphi_k + \Delta\theta_k)} + \text{noise} \end{aligned} \quad (3)$$

where an overbar denotes complex conjugation. We see that with a small channel phase difference $\Delta\theta_k = \theta_k - \theta_{k-1}$ and moderate noise levels, the transmitted phase can be distinguished. The performance of D-PSK is plotted in figure 3 and compared to the performance of coherent PSK. We see that the difference is quite small for BPSK and slightly larger for QPSK [6].

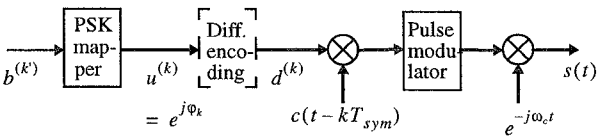

Figure 4. A (differential) direct sequence spread spectrum transmitter.

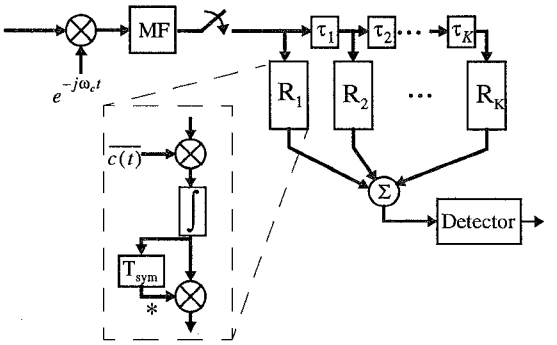

Figure 5. A RAKE receiver for differential DS SS modulation. An overbar or asterisk denotes complex conjugation.

C. D-PSK in Direct Sequence Spread Spectrum

PSK is usually used as the symbol modulation form in direct sequence spread spectrum (DS SS) systems[1] and often in combination with differential coding; i.e. the bits to be transmitted are formed into binary or quadrature phase symbols and encoded according to (1) before modulating the usually binary valued pseudo-noise sequence *c(t)*, as illustrated in figure 4.

With the assumption that the transmitted signal is subjected to a multi-path channel, the differential DS SS receiver which is shown in figure 5 is appropriate [6]. Each R_i unit consists of a correlator and a differential demodulator. With a well chosen code *c(t)*, the output of the correlator's integrator is still accurately described by (2). The output of the R_i units can then be added together to form one symbol sample on which a decision can be made.

With perfect synchronization and proper design so that the delay-line time slot of length $\sum\tau_i$ is larger than the delay spread T_m of the channel, essentially all multi-path signal energy can be used in the detection [7]. This structure is commonly referred to as a RAKE and can also be used in a coherent manner which however requires phase recovery in each of the fingers of the RAKE (replaces the differential demodulator). If the different multipath components of the channel are assumed to fade independently, the RAKE can be viewed upon as a diversity combiner, since the different multipath components are resolved in the RAKE fingers.

It is clear from figure 5 that the correlator and differential demodulator in a RAKE finger will require careful design and optimization.

1. As a good reference on spread spectrum, we recommend the tutorial by Pickholtz et. al. [7].

II. Implementational Details

A. Preliminary Considerations

The three most critical parameters in a digital implementation of a RAKE finger are the processing gain $L_c = T_{sym}/T_c$ (i.e. the number of code-bits (chips) of duration T_c per symbol of duration T_{sym}), the oversampling ratio $L_{os} = T_c/T_s$ (i.e. the number of samples of duration T_s per chip), and the word width n_{adc} of the incoming sample. L_c is a system requirement parameter and L_{os} is of more importance with respect to code synchronization (which is not treated in this paper). The proper choice of n_{adc}, however, needs commenting before proceeding.

There is a trade-off between the dynamic range of the digital receiver on one hand and the complexity (read speed and power consumption) of the adders and other subsequent logic on the other hand. One strategy, which is employed in e.g. Berkeley's Infopad terminal [8], is to keep n_{adc} as small as possible in order to minimize power consumption in both the ADC's and the digital logic; this puts high demands on the preceding AGC (automatic gain control). Another is to relax the requirements on, or even eliminate the need of, an AGC by increasing n_{adc}, which is our strategy ($n_{adc} = 8$).

In a digital implementation of the correlator, the accumulator (i.e. integrator) constitutes the greatest challenge. In order to avoid overflow, the adder needs to be sufficiently large; essentially, the word width difference between the output and input of the correlator increases linearly with the exponent of the spreading factor and the oversampling ratio, i.e.

$$n_{corr} = n_{adc} + \log_2(L_c) + \log_2(L_{os}) \,. \tag{4}$$

Hence with an initial word width of 8 and a maximum L_cL_{os} of ~250, the correlator output will be 8 + $\log_2$(250) = 16 bits wide. Clearly the complex multiplier in the subsequent differential detector will become excessively large if the operands are 2x16 bits wide. The next section presents a way to reduce the word widths of the complex operands with nearly no sacrifice of performance.

B. Bit-stripping

Our solution, which we call bit-stripping, was found to be a conversion from the 16-bit integer representation of $y^{(k)}$ to a complex-exponential-representation. To see this we recognize that $y^{(k)}$ from (2) can be written as:

$$y^{(k)} = y_i^{(k)} + jy_q^{(k)} = (\tilde{y}_i^{(k)} + j\tilde{y}_q^{(k)}) \cdot 2^{-x^{(k)}} \tag{5}$$

where $x^{(k)}$ is the largest possible integer exponent such that the 16-bit integers $\tilde{y}_i^{(k)}$ and $\tilde{y}_q^{(k)}$ do not overflow (i.e. exceed $2^{15} - 1$ or go below -2^{15}). It is now possible to truncate $y^{(k)}$ without risking to lose too much phase information. A (16-l)-bit truncation yields

$$\begin{aligned} y^{(k)} &= (\tilde{y}_i^{(k)} \cdot 2^{-(16-l)} + j\tilde{y}_q^{(k)} \cdot 2^{-(16-l)}) 2^{-x^{(k)}+(16-l)} \\ &\approx (\hat{y}_i^{(k)} + j\hat{y}_q^{(k)}) \cdot 2^{-x^{(k)}+(16-l)} = \hat{y}^{(k)} \end{aligned} \tag{6}$$

where $\hat{y}_i^{(k)}$ and $\hat{y}_q^{(k)}$ are scaled, l-bit versions of the MF outputs. The complex multiplication as defined in the first line of (3) now becomes

$$\begin{aligned} v^{(k)} &= y^{(k)} \times \overline{y^{(k-1)}} \approx \hat{y}^{(k)} \times \overline{\hat{y}^{(k-1)}} = \\ &= (\hat{y}_i^{(k)} + j\hat{y}_q^{(k)}) \times (\hat{y}_i^{(k-1)} - j\hat{y}_q^{(k-1)}) \cdot \\ &\quad \cdot 2^{((16-l)-x^{(k)})+((16-l)-x^{(k-1)})} \end{aligned} \tag{7}$$

Since we for decoding only are interested in the phase of the complex product $v^{(k)}$, the exponent of (7) is of no interest in the decision process.

The performance of this simplified architecture on an additive white Gaussian noise channel has been evaluated through simulations. The main interest was to find the smallest truncated word-width l allowed without decreasing the performance compared to ideal differential detection as shown in figure 3. In figure 6, the bit error rates (BER) for three different values of l are shown and compared to the ideal bit error probability of differential BPSK. In the same figure, three different cases for D-QPSK are also shown. We conclude that l = 6 will give virtually no performance degradation for D-BPSK and very little for D-QPSK.

III. Logic Design

A. Overview

The detection scheme described in the previous section has been implemented in our DS SS transceiver. The transceiver, which is shown in figure 1, consists of a baseband transmitter and receiver and a configuration unit which handles status reading and configuration data writing from/to the circuit.

Preceding the detector in the receiver is a correlator unit which apart from despreading the incoming data does code synchronization by means of a modified delay-locked loop [9].

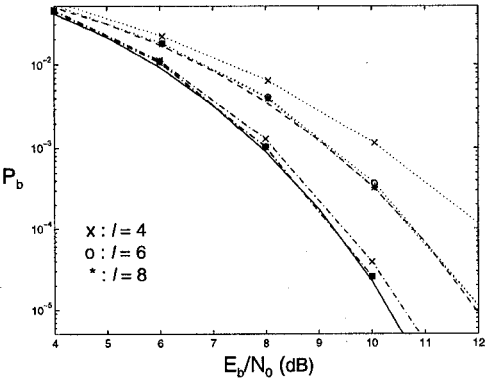

Figure 6. Simulated bit error rates for D-BPSK (dashed-dotted) and D-QPSK (dotted) for different values of bit-width l. Ideal D-BPSK (solid) and ideal D-QPSK (dashed) are shown as a reference (solid).

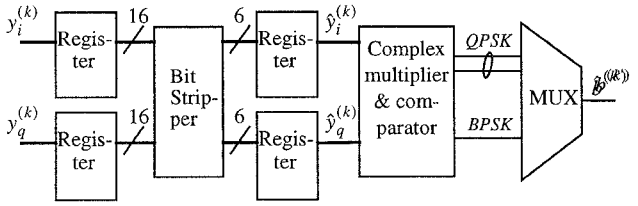

Figure 7. A block diagram of the detector architecture. The registers are updated every symbol period.

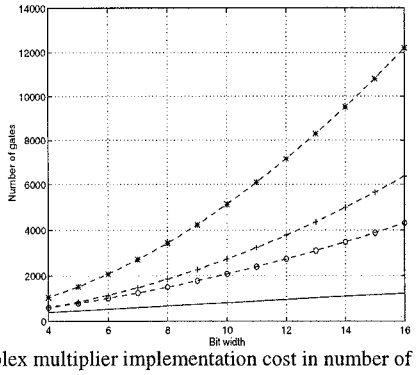

Figure 8. Complex multiplier implementation cost in number of gates for 1) four multipliers (*), 2) two multipliers (+), 3) one multiplier (o), and 4) four bit-serial/parallel multipliers (solid line).

In the current transceiver version there is only one correlator (i.e. no RAKE). The output of the correlator is one complex 2x16 bits wide symbol sample every T_{sym} seconds. In section II the value of the product $L_c L_{os}$, which is equivalent to the number of clock cycles per symbol period, was maximized to 250. Because of the configurability of our transceiver, it is not fixed; i.e. the parameters $f_{sym} = 1/T_{sym}$, L_c and L_{os} can all be varied. However, the product $L_c L_{os}$ is lower limited to 20. This is the most important constraint in the design of the detector. With the detector partitioned as indicated in figure 7, this means that we have at least 20 clock cycles for the bit-stripping operation and then another 20 cycles for multiplication and decoding.

We can now examine different solutions to achieve an area-efficient circuit implementation of the detector. Multipliers are complex circuits and by examining different types of resource sharing strategies we found an efficient solution. We will here show what was taken into consideration when the final architecture for the detection unit was selected. We will use the complex multiplier as an example, but a similar analysis was done for the bit-stripping unit. One complex multiplication requires four real-number multiplications and two additions. We have considered four different types of resource sharing solutions: 1) A fully parallel solution with four multipliers, 2) re-usage of two multipliers twice, 3) re-usage of one multiplier four times, and 4) four bit-serial/word-parallel multipliers [10].

Even for the moderate bit width of six, the bit-serial/word-parallel solution is less than 50% of the size compared to any of the other, see figure 8. A full bit-serial solution, however, will not reduce the complexity any further due to the increase of registers needed for temporary storage. From figure 8 we can also make a coarse estimation of the initial savings we do

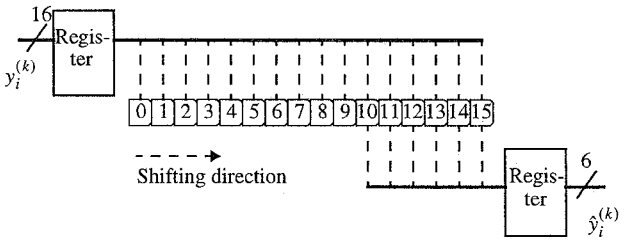

Figure 9. One of the bit-stripping shift registers.

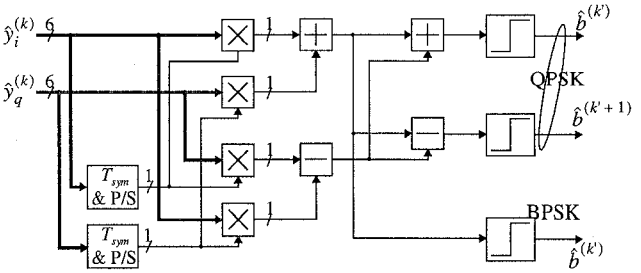

Figure 10. A schematic of the complex multiplier and symbol decoder.

by stripping from 16 to 6 bits. If we assume that a 16x16 complex multiplication requires a fully parallel implementation, we save nearly 85% by using a fully parallel 6x6 complex multiplication. Eventually, we have selected a bit-serial/parallel architecture for the complex multiplication and detection and a bit-serial solution for the bit-stripping, and in the following two sections we will in detail describe the functionality of these units.

B. Bit-stripping

From the 2x16 bits arriving at the input, the 2x6 "significant" bits should be picked out according to the principle described in section II. This can be done by parallel loading the two 16 bits words into two shift registers, see figure 9, and then shifting these in the direction indicated in the figure until either of them has non-equal bits in position 14 and 15. When that happens, the bits in position 15 down to 10 in each shift register are loaded into the output registers. This procedure works because as long as bit 14 equals bit 15 in a 16 bit two's complement integer, it is just a prolongation of the sign bit. One stripping operation is designed to take less than 17 clock cycles and is controlled by a finite state machine.

C. Complex Multiplication and Decoding

From the bit-stripping unit, the $\hat{y}_i^{(k)}$ and $\hat{y}_q^{(k)}$ are fed into the complex multiplier shown in figure 10. Both inputs are delayed one symbol period and serialized in a parallel-to-serial converter (P/S). The complex multiplication is completed in the subsequent adder where the real part is formed, and in the subtractor where the imaginary part is formed. The result at the output of the complex multiplier is sufficient to do BPSK decoding, since the value of the transmitted symbol then can be determined by observing at the sign bit of the real part. QPSK decoding requires comparisons between the real and imaginary parts, which is carried out in the adder and subtractor after the complex multiplication. It should be noted that

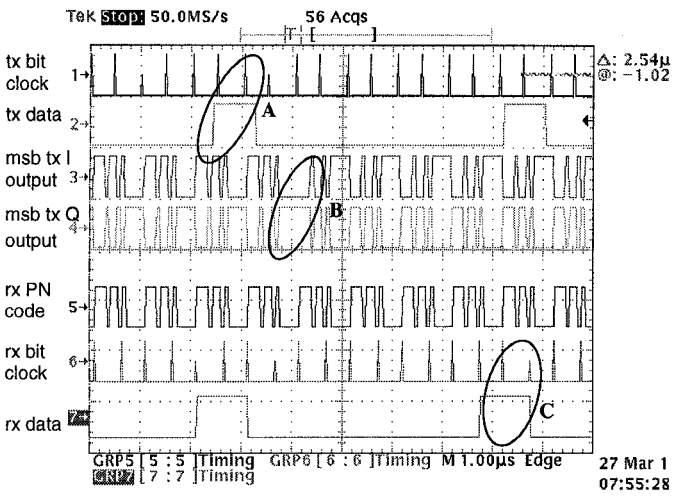

Figure 11. Printout from measurements on the fabricated DS SS transceiver circuit. In event A, two 1's are clocked in to the transmitter resulting in a 180° phase shift (event B). Delayed a few symbol periods, the two 1's come out from the receiver (event C).

all operations after the multiplication, is carried out on bit-serial data, and the logic needed for detection becomes very simple.

IV. Fabrication and Measurements

The DS SS transceiver was designed with AMS 0.8 µm CMOS standard cell library and fabricated during 1Q96, see figure 12. Preliminary measurements indicate i) that the circuit functions as specified (see figure 11) and ii) that the correlator and synchronization units which are the most time critical components handle sampling frequencies $f_s = 1/T_s$ up to around 75 MHz. The power consumption at the targeted f_s = 50 MHz and a supply voltage of 3.3 V is in the range of 120 - 140 mW.

V. Conclusions

In this paper we have presented the circuit architecture of our DS SS modem. A structure for D-PSK detection which dramatically reduces the detector complexity while preserving the dynamic range, was introduced. It was shown that the non-idealities introduced by our approach have very small degrading effect on the performance in terms of the bit error rate for a Gaussian channel. By doing this so called bit-stripping at the input of the detector, area savings of nearly 85% are possible, when comparing fully parallel structures. This results in that power will be saved to the same extent. In addition, trade-offs between serial and parallel implementations can result in additional savings in area and power. E.g. by careful selection of implementation technique for the detector, we have shown that another 50% size reduction is obtainable by using bit-serial/ word-parallel and bit-serial structures.

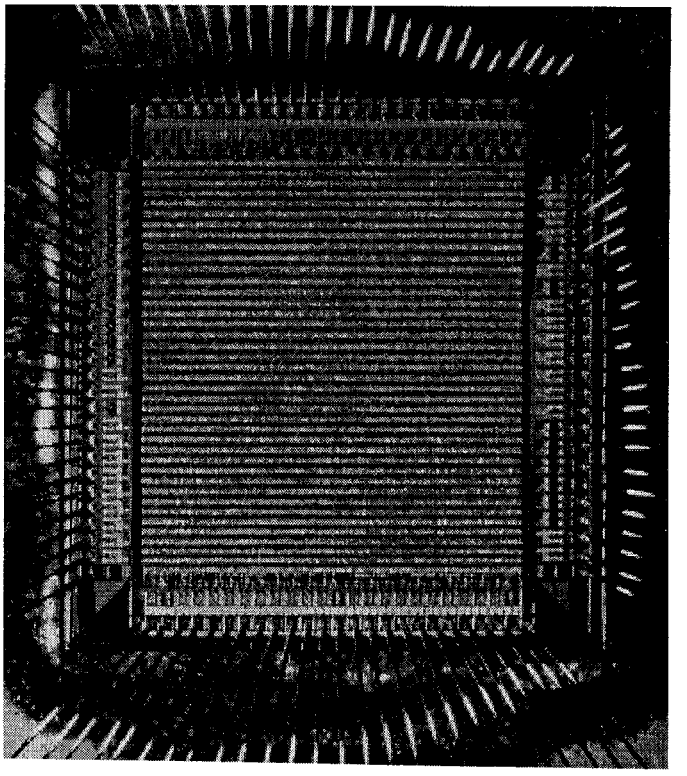

Figure 12. Photograph of the processed circuit. The die area is approximately 4x4 mm and there are slightly more than 100 pads.

VI. References

[1] R. Hager, A. Klemets, G. Maguire, M. Smith, F. Reichert, "MINT - A Mobile Internet Router," *Proc. of the 43rd IEEE Vehicular Technology Conference*, New Jersey, USA, May 93

[2] D. Kerek, H. Tenhunen, G. Maguire, F. Reichert, "The Walkstation Transceiver Design," *Proc. of the 44th IEEE Vehicular Technology Conference*, Stockholm, Sweden, June 1994.

[3] D. Kerek, H. Olson, H. Tenhunen, G. Maguire, F. Reichert, "Direct Sequence CDMA Technology and its Application to Future Portable Multimedia Communication Systems," *Proc. of the third IEEE Int. Symp. on Spread Spectrum Tech. and Applications*, Oulu, Finland 1994.

[4] D. Kerek, T. Saluvere, H. Tenhunen, "Direct Sequence Spread Spectrum Digital Radio DSP Prototyping Using Xilinx FPGAs," *Proc. of FPL '94*, Prague, Czechia 1994.

[5] H. Olson, D. Kerek, H. Tenhunen, "Direct Sequence Spread Spectrum Digital Radio Performance Analysis with Simulation," *Proc. of 36th SIMS Simulation Conference*, Stockholm, Sweden 1994.

[6] J. G. Proakis, *Digital Communications*, McGraw-Hill, USA 1989.

[7] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of Spread Spectrum Communications - A Tutorial," *IEEE Trans. Commun.*, pp. 855 - 884, May 1982.

[8] S. Sheng, R. Allmon, L. Lynn, I. O'Donnnell, K. Stone, R. Brodersen, "A Monolithic CMOS Radio System for Wideband CDMA Communications," *Proc. of Wireless '94 Conf.*, 1994, Canada.

[9] J.K.Holmes, *Coherent Spread Spectrum Systems*, Wiley-Interscience Publications, New York, USA 1982.

[10] N. R. Powell and J. M. Irwin, "Signal Processing with Bit-serial Word-parallel Architectures," *Real-time Signal Processing, Proc. SPIE* Vol. 154, 98-04, 1978.