A 1mm ^{2} 1.3 mW GSM/EDGE Digital Baseband Receiver ASIC in 0.13 μm CMOS

C. Benkeser*‡, A. Bubenhofer*, Q. Huang*‡

*Integrated Systems Laboratory
ETH Zurich, 8092 Zurich, Switzerland

‡Advanced Circuit Pursuit AG
8702 Zollikon, Switzerland

***Abstract*—This paper addresses complexity issues at algorithmic and architectural level of digital baseband receiver ASIC design for GSM/GPRS/EDGE, in order to reduce power and die area as desired for cellular applications. A 2.5G multi-mode architecture is implemented in 0.13 μm CMOS technology occupying 1.0 mm^2 and dissipating only 1.3 mW in fastest EDGE data transmission mode.**

***Index Terms*—Mobile Communication, 2G, 2.5G, GSM, EDGE, EGPRS, Equalizer, Baseband, Receiver, Low Power.**

I. Introduction

Cellular wireless communications has changed our lives in the past 20 years. Today, with 3.5 billion subscribers [1] the far most important cellular communication system GSM is used by all sorts of people all around the world. Simple 2G data communication is possible with the standard extension GPRS, but good user experience during wireless web-browsing and other non-voice applications can only be achieved since 2.5G EDGE data-services have been introduced.

EDGE increases 2G peak data rates to 240 kbps, mainly due to a higher modulation order (8PSK) and stronger channel coding. The symbol rate of 270 kbps is identical with basic GSM, but 8PSK modulation improves the spectral efficiency, which comes at the cost of receiver complexity. Especially channel equalization and symbol demodulation is a challenging task in EDGE, and requires sophisticated low-complexity solutions to preserve the power-advantage and cost-advantage of 2G receivers. Although basic GSM can be implemented on advanced signal processors with tolerable power-levels today, EDGE requires efficient ASIC accelerators to reduce the power consumption to what is required in modern cellular phones.

In GSM with GMSK modulation typically the (optimum) Maximum-Likelihood Sequence-Estimator (MLSE) is used for channel equalization and demodulation. The complexity of an MLSE algorithm for 8PSK signals, however, would even exceed what can be implemented on ASICs. Therefore, in recent years a lot of work on channel equalization for EDGE has been performed. It has been shown that sub-optimum variants of the MLSE algorithm with acceptable computational complexity are suitable for channel equalization of 8PSK modulated EDGE signals (e.g., [2], [3]). Turbo equalization has also been discussed [4] and proposed as a solution for EDGE [5]. However, all these explorations are simulation-based system analyses where implementation aspects and crucial design metrics like power consumption are not considered. Further, a practical multi-mode solution supporting GMSK as well as 8PSK modulation have not been treated so far.

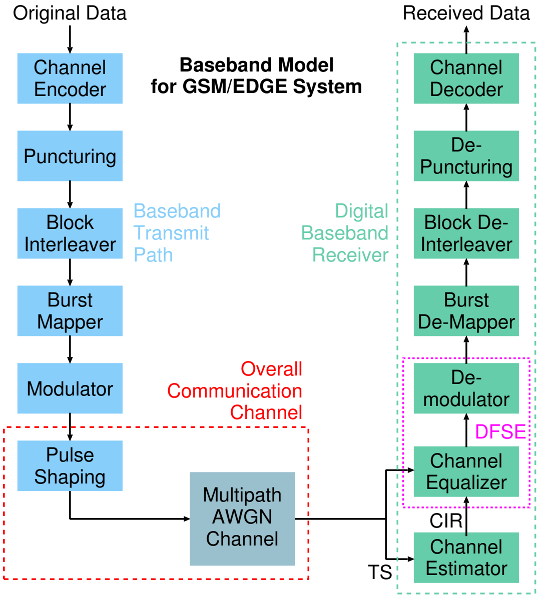

Fig. 1. System model of the GSM/EDGE baseband transceiver model.

This work describes the design of a low-power, low-cost digital baseband receiver ASIC for GSM/GPRS/EDGE with a focus on the most challenging block, the channel equalization. To this end, solutions for the key components have been realized that provide the flexibility to deal with the required modulation types and coding schemes. Costly hardware resources are shared to keep the silicon area (cost) low, and algorithmic/architectural optimizations have been applied to achieve very low power consumption.

A. Outline

The paper is organized as follows. In Section II the GSM baseband transmitter and channel model are introduced. In Section III the high-level architecture of the receiver ASIC realized in this work is presented. Section IV describes the implemented channel equalizer solution in detail. Finally, measurement results are presented in Section V, and Section VI concludes the paper.

II. System Overview

Fig. 1 shows the GSM/EDGE baseband transceiver system model used in this work. On the transmitter side the input data is first encoded with a convolutional encoder (rate 1/2 for GSM/GPRS or 1/3 for EDGE). Then, selected bits are removed in the puncturing unit to adjust the code rate as required for the specific transmission mode. After that, the order of bits is scrambled in the block interleaver to distribute data blocks of up to 1184 bits across 4 or 8 bursts [6]. The resulting bits are mapped in interleaved order on bursts with a length of 156.25 symbols, which includes fixed tail bits for channel equalization, a fixed training sequence for channel estimation and a guard period[1] as shown in Fig. 2. Finally, the bursts are GMSK or 8PSK modulated with a symbol period of $T_s = 3.69\,\mu s$ and pulse shaped [7]. Typically the bursts are transmitted every 8 time slots as indicated in Fig. 2 according to the time division multiple access scheme defined for GSM, and the remaining 7 slots are reserved for other users. However, several slots can be reserved for data traffic of a single user in high-speed EDGE data transmission modes.

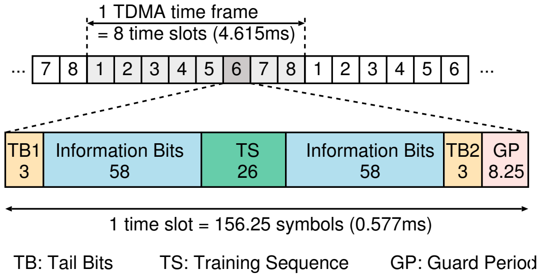

Fig. 2. Time division multiple access and the *normal burst* in GSM.

The bursts are submitted through a multipath communication channel with Additive White Gaussian Noise (AWGN). The GSM test channel profiles [8] have a delay spread of up to $5\,T_s$, causing considerable inter symbol interference (ISI). Even more ISI is introduced by the system itself, because the GSM channel bandwidth of 200 kHz is significantly lower than the symbol rate. Therefore, an overall communication channel with a delay spread of up to $8\,T_s$ has to be considered when removing ISI in the receiver with the channel equalizer.

III. Digital Baseband Receiver

The complete receive chain shown in Fig. 1 has been implemented in our ASIC. As indicated in the corresponding simplified block-diagram (Fig. 3) two 2.5 kb RAMs allow for buffering of up to 2 GSM bursts, each comprising the in-phase and quadrature-phase component of 156 symbols. Each time a new GSM burst has been received the channel impulse response (CIR) is estimated with the training sequence (TS) by using the Least Squares technique [9]. The ISI of unknown data symbols adjacent to the training sequence causes large estimation errors when considering all 26 symbols of the TS. Therefore, only the central 16 symbols of the TS are used to estimate the CIR in our design. Since linear 8PSK modulation can be realized with simple FIR filtering in the transmitter, the CIR of the overall communication channel including the pulse shaping filter can be estimated at the receiver (c.f., Fig. 1). In (non-linear) GMSK instead the signal is modulated indirectly by filtering and accumulating the phase. Hence, the transmitter pulse shaping can not simply be included in the estimated CIR.

Channel equalization and demodulation are performed in our receiver implementation with the combination of a minimum-phase FIR pre-filter and a Decision-Feedback Sequence Estimator (DFSE) which are explained in detail in Section IV. The pre-filter coefficients computed with the estimated CIR are buffered to be used to filter the CIR and the stream of received symbols. These are fed into the DFSE for channel equalization and demodulation.

High-speed EDGE transmission modes require continuous burst reception, because more than one burst can be allocated to a specific user in each time frame. Therefore, the demodulation of each burst has to be finished in one burst period of $T_b = 577\,\mu s$. When 4 or 8 bursts have been demodulated and de-mapped, the de-interleaver can begin to reorder the bits of the received data block. Both, de-mapping and de-interleaving are performed by generating the specific RAM addresses for writing to and reading from the intermediate buffer. Then, the de-puncturing unit introduces zeros (no a-priori information on the demodulated symbol) into the data stream to provide data blocks to the channel decoder with the correct code rate. A flexible Viterbi decoder for trellises with up to 64 states has been implemented to support the specified coding schemes. The channel decoder processes all branches in parallel, i.e., one trellis step per clock cycle, to keep the receiver latency low.

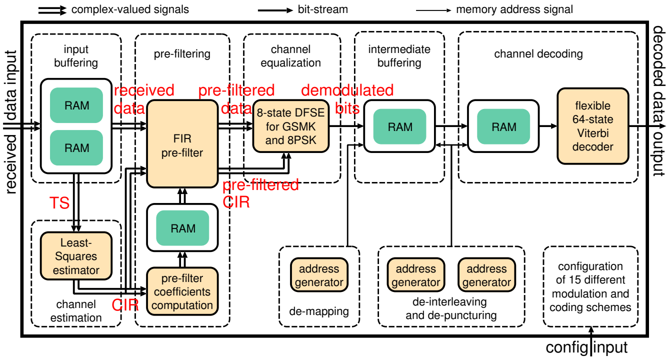

Fig. 3. GSM/EDGE receiver ASIC block diagram.

IV. Combined Pre-filter and DFSE Solution

In an MLSE, as typically used for GSM, all possible symbol combinations of length L that could have been transmitted are generated as reference symbol sequences. These sequences are modulated and convolved with the CIR to emulate the effect of ISI. The resulting reference signals are compared with the received samples to generate branch metrics which serve as the likelihood that the specific symbol sequence has

[1] The guard period of 8.25 symbols ensures that subsequent bursts do not overlap and that different distances between transmitter and receiver can be compensated.

been transmitted, given the received signal. The branch metrics are used by the Viterbi algorithm in a trellis to find the most probable transmitted symbols. Contrary to implementations of Viterbi decoders, the branch metric generation is the most challenging part in an MLSE. The computational complexity can be characterized with the number of trellis branches $B = M^L$ to be processed, where M denotes the modulation order (alphabet size) with $M = 1$ for GMSK and $M = 3$ for 8PSK. The length of the reference symbol sequences L defines the maximum tap delay in the channel profile that is considered in the equalization. The storage requirements are defined by the number of states in each trellis stage according to $S = M^{L-1}$.

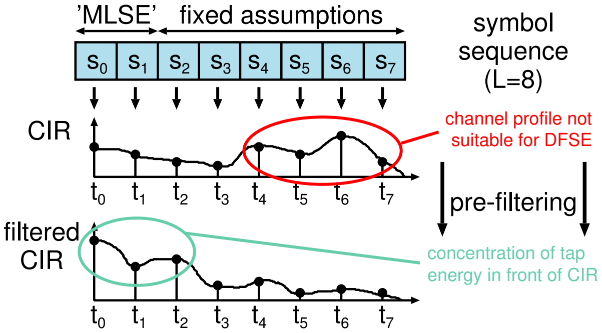
'MLSE'
fixed assumptions
symbol sequence (L=8)
s_0 s_1 s_2 s_3 s_4 s_5 s_6 s_7
CIR
channel profile not suitable for DFSE
t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7
pre-filtering
filtered CIR
concentration of tap energy in front of CIR
t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7

Fig. 4. The concept of pre-filtering required for the DFSE.

Contrary to the MLSE algorithm, in the DFSE not all possible symbol combinations of the reference sequence are used for the comparison with the received samples. Decisions of the Viterbi algorithm on previous symbols, so-called per-survivor estimates, are taken as fixed assumptions to reduce the number of branch metrics to be computed. Only the permutations of the first D symbols are considered as separate state transitions in the trellis, which reduces the number of branches to be processed to $B = M^D$ and the required number of trellis states to $S = M^{D-1}$. Therefore, the taps of the CIR which correspond to these first D symbols have the greatest impact on the equalizer performance. In channels with strong delayed taps and comparably weak main taps, correct symbol demodulation can be difficult or even impossible for the DFSE algorithm. Therefore, minimum-phase pre-filtering is required to transform such channels into channels better suited for the DFSE. Such a pre-filter concentrates the energy of the channel taps in the front, in order to maximize the impact of the first D symbols, thus improving the DFSE performance (c.f., Fig. 4).

Simulations have shown that, depending on the channel profile, varying D from 2 to 4 brings a maximum performance gain of 1.0 dB-1.6 dB in terms of SNR required to achieve an uncoded DFSE bit error-rate (BER) of 10^{-4} with both GMSK and 8PSK modulated signals. Almost optimum (MLSE) performance can be achieved with $D = 4$, and further increasing D introduces additional complexity without a noticeable performance gain.

The impact of the pre-filter can be more significant. The uncoded BER of demodulated 8PSK signals, impaired by the equalizer test channel [8], are shown in Fig. 5. Different filter orders p for the pre-filter in front of a DFSE with $D = 2$ have been simulated. As can be seen, without pre-filtering the symbol sequence can not be demodulated with a DFSE at all. A pre-filter order p of less than 20 causes the BERs to saturate, and the achievable DFSE performance degrades more than 3 dB in terms of SNR, whereas the BER does not improve any further with $p \geq 32$.

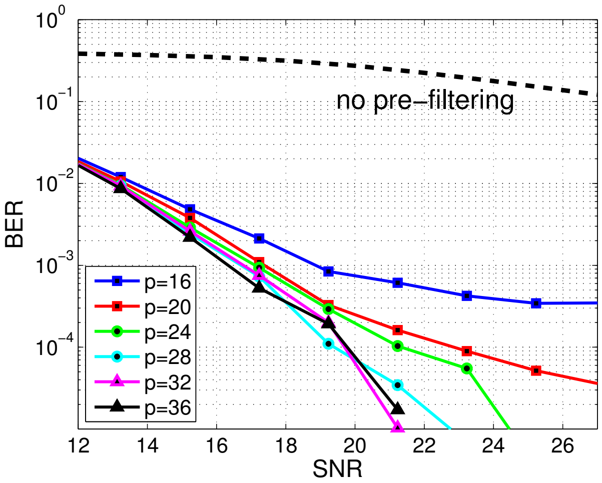

Fig. 5. Uncoded BER of 8PSK EDGE signals after pre-filtering and channel equalization with DFSE ($D = 2$) with variable pre-filter order p.

The most critical parts of a GSM/EDGE digital baseband receiver hardware implementation, i.e., the matrix inversion for the pre-filter coefficients computation and the DFSE realization, are described in detail in the following.

A. Matrix Inversion in Pre-filter Implementation

A suitable pre-filter transforms the CIR to its minimum-phase equivalent (e.g., [10], [11]), where the energy of the first filter taps is maximum compared to the impulse response of all other causal and stable filters with the same magnitude response [12]. Such pre-filtering can be realized with a hardware-friendly FIR filter, and the corresponding pre-filter coefficients can be computed by using the Linear Prediction (LP) method which requires less computations than comparable approaches [11]. In a pre-filter implementation with LP the far most complex part is the inversion of a complex-valued pxp Toeplitz matrix required to generate the pre-filter coefficients.

In our digital baseband receiver ASIC a pre-filter with order $p = 32$ has been implemented to guarantee maximum DFSE performance, even when the received signals have been impaired by channels with strong delayed taps. The required inversion of a 32x32 matrix with an internal precision of 2x20 bit has been implemented in our design by excessively applying time-multiplexing and resource sharing to curtail silicon area. To this end, the recursive Levinson-Durbin (LD) algorithm shown in Alg. 1 is used. Its recursive nature is highly suitable for sequential implementations and the computational

```
• Initialization of the (real) scaling factor $E$
      $E = \varphi_0$                                                  (1)
• Recursive generation of the (complex) coefficients $a_i$
  by looping over $1 \leq i \leq q_p$
    – Computation of the unscaled $a'$
          $a' = -\left[\varphi_i + \sum_{j=i}^{i-1} a_j \cdot \varphi_{i-j}\right]$          (2)
    – Scaling with $E$ to generate the new $a_i$
          $a_i = \frac{a'}{E}$                                          (3)
    – Updating $a_j$ by looping over $1 \leq j \leq i-1$
          $a_j = a_j + a_i \cdot a^*_{i-j}$                             (4)
    – Generation of the new scaling factor
          $E = (1 - |a_i|^2)E$                                          (5)
• The LP filter coeffs are given by $a_i$ with $1 \leq i \leq q_p$
```

Alg. 1. LD recursion for complex coefficients.

complexity is close to optimum matrix inversion algorithms for Toeplitz matrices [13].

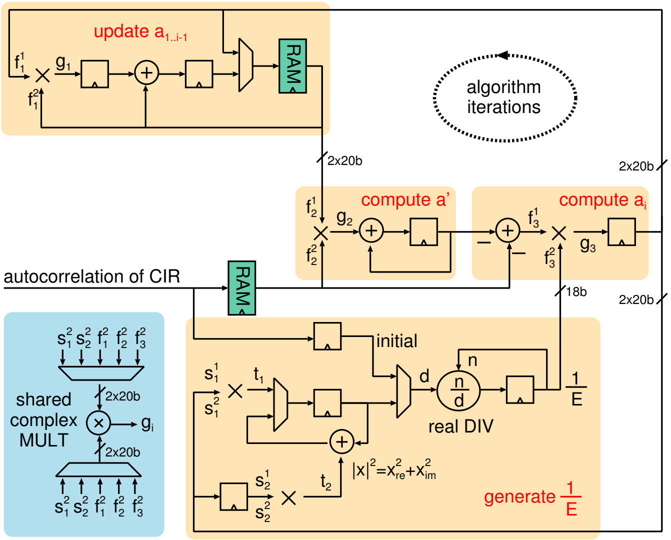

Fig. 6. Block diagram of the LD recursion implementation.

The block diagram of the LD algorithm implementation in our receiver ASIC is shown in Fig. 6. By working with the inverse of the scaling factor $1/E$, the algorithm can be implemented more efficiently: the division of the complex-valued a' in (3) is replaced by a multiplication, and the the multiplication in (5) is replaced with a real-valued division. Thus, one costly real-valued division is saved in each iteration of the LD recursion. Further, the division now has to be performed in the generation of the new (inverse) scaling factor in (5), and therefore can be performed mostly in parallel to updating the a_j. Thus, the division can be realized with a low-complexity sequential divider, which requires 9 clock cycles, without slowing down the LD recursion. This sequential divider implementation significantly reduces the logic depth of the costly divide operation, such that timing closure of our receiver ASIC is not affected, and requires only 1.7 k gate equivalents (kGE). In our design the matrix inversion of the complex-valued 32x32 matrix takes 2630 clock cycles.

Costly hardware resources like multipliers, register arrays and RAMs, used in the LD recursion implementation, are also shared with other units of the pre-filter block. Thus, the total pre-filter coefficients computation together with the actual FIR pre-filtering requires only 32 kGE and 2.5 kb memory. The total pre-filter coefficients computation takes less than 3000 clock cycles, and the FIR pre-filter of the received burst provides one filtered symbol per 16 clock cycles to the DFSE.

B. *DFSE Design for GMSK and 8PSK*

To support processing both GMSK and 8PSK modulated signals with little overhead, the DFSE algorithm always operates on the same number of trellis states. In GMSK mode 8 states allow $D = 4$ symbols to be considered in building the trellis, such that near MLSE performance can be achieved. In 8PSK mode with $D = 2$ the high-order pre-filter allows for good performance without having to introduce additional trellis states as previously shown. The binary alphabet in GMSK requires only 2 branches per trellis state to be processed, whereas 8PSK has 8 branches in the trellis going to each trellis state, which leads to a 4x higher computational complexity.

The symbols of the TS do not have to be demodulated since they are known at the receiver. Therefore, computations can be saved by simply skipping the received symbols that correspond to the TS. Our DFSE implementation builds two separate sub-trellises per burst: one trellis for the first 58 information bits, using TB1 and the first symbols of TS as tail bits, and one trellis for the second 58 information bits, using the last symbols of TS and TB2 as tail bits (c.f., Fig. 2). Consequently, tail bits for both sub-trellises are known in the DFSE, which enables reliable demodulation of the information bits with 20% less symbols to be processed.

Fig. 7 shows the block-diagram of the implemented DFSE algorithm. To allow the equalization of all specified channel profiles, channel taps with a maximum delay of $L = 8$ symbol periods are considered in our DFSE implementation. Therefore, the reference symbol sequences have a length of 8 symbols, and the (pre-filtered) CIR fed into the DFSE block consists of 8 symbol-spaced taps.

To modulate the symbols each reference sequence (one for each trellis branch) is fed into a look-up table with all possible constellation points. In GMSK mode all 512 possible constellation points which arise from ISI due to the pulse shaping filter have to be stored. In 8PSK mode, where the estimated CIR also comprises transmitter pulse shaping, only 2x8 different constellation points have to be stored[2].

[2] 16 instead of 8 constellation points have to be considered, because a phase rotation of $3\pi/8$ between subsequent transmitted symbols is specified [7].

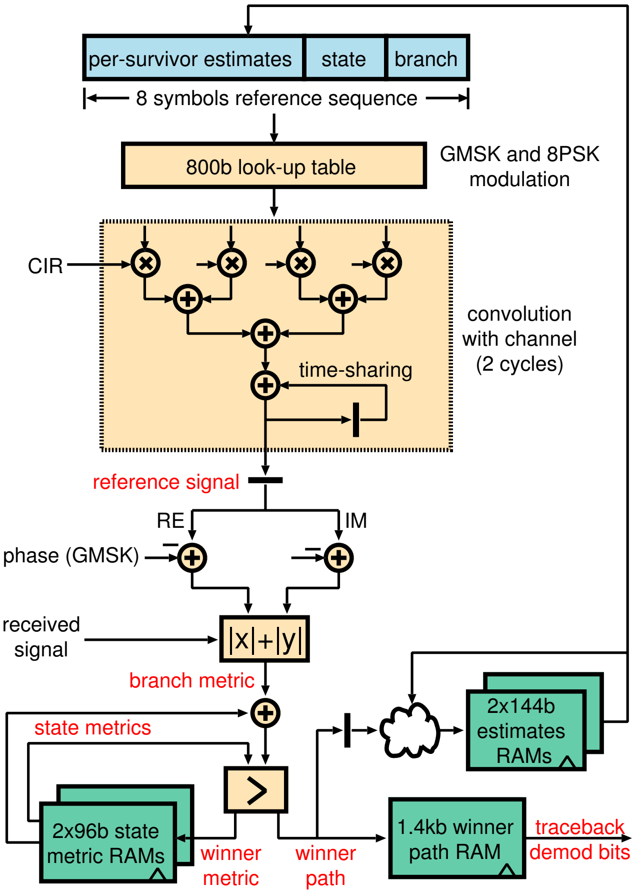

Fig. 7. Block diagram of the DFSE implementation.

With a precision of 6 bit and 8 bit for the GMSK and 8PSK constellation points respectively, the LUT can be be reduced to only 0.8 kb by exploiting symmetries.

The vector product of the modulated reference symbol sequence with the pre-filtered CIR, both complex-valued and with a length of 8, is by far the most computation-intensive part in a DFSE implementation: 8 multiplications and 7 additions are required for each trellis branch, leading to high implementation complexity. To keep the silicon area low we have time-multiplexed the vector product implementation for 4 symbols with 4 channel taps, thus spending 2 clock cycles for the generation of each reference signal.

In GMSK mode a phase rotation of $^{\pi}/_{2}$ introduced in the transmitter modulator [7] is compensated after the convolution with the channel. Finally, the trellis branch metrics (typically the Euclidean distance between the reference signal and the received sample) are generated by using the ℓ^1-distance as an approximation to avoid costly complex-valued square operations in this time critical path. In terms of SNR the performance degradation with this sub-optimal solution is less than 0.5 dB in GMSK and within 1 dB in 8PSK mode.

Each branch metric is added to the state metric of the state that the specific branch is originating. The smallest metric incident on each state is selected and stored in the state metric memory which consists of 2 RAMs to avoid that the old state metrics are overwritten. The decision on the winner branch is stored in the *winner path* and the *estimates* RAMs. To avoid access problems two *estimates* RAMs are required for the storage of the last $8-D$ per-survivor estimates for each of the 8 trellis states. The *winner path* RAM stores all decisions in the trellis for the final back-tracing, which finds the most likely symbol sequence.

The total DFSE implementation requires only 18 kGE and less than 2 kb memory. The demodulation of one GMSK burst takes 6228 clock cycles, and processing an 8PSK modulated burst requires 17360 cycles. When taking into account the time required to generate the pre-filter coefficients plus 128 cycles for the channel estimation, burst-wise operation of the fastest EDGE transmission modes is possible with a system clock frequency of 40 MHz.

C. *Complexity of Combined Pre-filter and DFSE*

The pre-filter block is almost twice as complex in terms of silicon area as the DFSE, and occupies one third of the logic cells of the total digital receiver. As previously shown, proper pre-filtering is essential to achieve high DFSE performance, especially when reducing the number of trellis states with $D < 4$. Although, the significant complexity of the pre-filter block gives rise to the question, if a pre-filter with a lower order and a DFSE with more trellis states would not achieve a similar performance at a lower total equalizer implementation complexity.

A lower pre-filter order causes more energy in the delayed taps of the filtered CIR, which requires more trellis states in the DFSE (increased D) to achieve comparable equalizer performance. Increasing the number of channel taps considered in building the trellis from $D = 2$ to $D = 3$ increases the number of branches to be processed by a factor of 8 in 8PSK modulation. Hence, the processing delay would increase by more than 120 k cycles with our architecture. On the other hand, with the highly sequential and resource-saving architecture of the proposed low-complexity pre-filter implementation, going down from a pre-filter order of 32 to for example 16 would save less than 2 k cycles. Furthermore, the silicon area of the pre-filter cannot be reduced significantly when lowering the filter order, because hardware resources have already been extensively re-used in the proposed architecture. Concluding, the implementation of a pre-filter with a high filter order pays off since it allows to reduce the complexity of the DFSE significantly.

V. Implementation Results

Functional tests and power measurements of fabricated chips (see Fig. 8) have been performed on a digital tester. The fixed-point simulation model of the implemented design has been verified and used to generate the block error-rates (BLERs) of the receiver as shown in Fig. 9 for the various modulation and coding schemes (MCS) of EDGE with the specified equalizer test channel. Other key characteristics are summarized in Table I. The ASIC occupies 1.0 mm^2 in 0.13 μm CMOS, comprising 97 kGE and 20 kb of memory. A measured maximum clock frequency of 172 MHz allows the

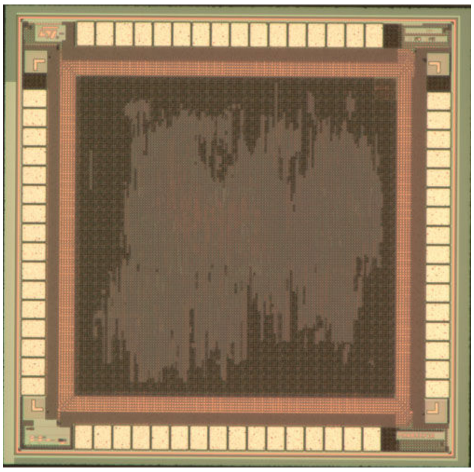

Fig. 8. Chip photo.

supply voltage to be lowered to 0.6 V when operating at the target frequency of 40 MHz. During continuous burst reception the average power consumption is as low as 5.2 mW in the fastest EDGE mode MCS9, and only 1.3 mW when lowering the supply voltage.

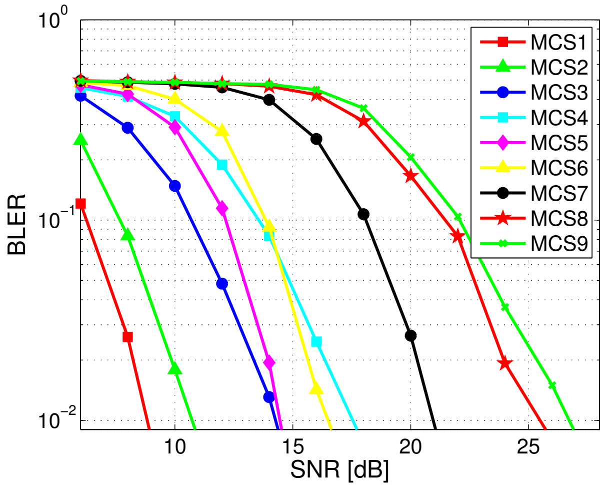

Fig. 9. Receiver block error-rates in various modulation and coding schemes (MCS) of EDGE.

VI. Conclusion

EDGE enables 2G data services all over the world with data rates suitable for many cellular applications. The introduction of 8PSK modulation improves the bandwidth utilization, but significantly increases equalizer complexity. Latest 65 nm signal processors provide the computational power required to process the digital baseband signal processing for EDGE. However, power and silicon area of such implementations exceed what is desirable for a low-power, low-cost 2.5G solution required in modern cellular phones.

In this work a combined architecture for GSM/GPRS/EDGE has been presented. We have shown how the digital baseband can be mapped efficiently to dedicated hardware by using suitable algorithms and architectures to share hardware resources with little overhead. A certain amount of flexibility allows to support the specified transmission modes with different modulation and coding schemes. Our approach to reduce complexity in the most crucial block, the channel equalizer, is to maximize the effort in the pre-processing FIR filtering, in order to be able to reduce DFSE complexity.

Technology	**0.13 μm CMOS**
Supply Voltage V_{DD}	**1.2 V-0.6 V**
Core Size	**1.0 mm^2**
Gate Count	**97 kGE**
Total Memory	**20.3 kb**
Max. System Clock Frequency	**172 MHz**
Leakage current @ V_{DD}=1.2 V (0.6 V)	**0.49 mA (0.15 mA)**
Average power measured during continuous burst reception @ target frequency f=40 MHz, V_{DD}=1.2 V (0.6 V)	
GSM CS1 (GMSK)	**2.4 mW (0.6 mW)**
EDGE MCS9 (8PSK)	**5.2 mW (1.3 mW)**

TABLE I
Key characteristics of the receiver ASIC implementation.

With 1.0 mm^2 and only 1.3 mW average power consumption our ASIC shows that cost-effective multi-mode digital baseband receiver accelerators for GSM/EDGE can be realized at ultra low-power.

Acknowledgments

The authors would like to thank A. Burg and L. Bruderer for their support. This project was funded by CTI, Switzerland in collaboration with Advanced Circuit Pursuit AG.

References

[1] "GSM Association, Market Data, www.gsmworld.com, August 2009."

[2] J. Olivier *et al.*, "Efficient equalization and symbol detection for 8-PSK EDGE cellular system," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 3, pp. 525–529, May 2003.

[3] W. Gerstacker and R. Schober, "Equalization concepts for EDGE," *IEEE Transactions on Wireless Communications*, vol. 1, no. 1, pp. 190–199, Jan. 2002.

[4] L. Szczecinski and I. Soto, "Is turbo equalization useful in EDGE systems?" *Vehicular Technology Conference*, vol. 1, no. 6-9, pp. 80–84, May 2002.

[5] C. Laot, R. Le Bidan, and D. Leroux, "Low-complexity MMSE turbo equalization: a possible solution for EDGE," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 965–974, May 2005.

[6] *3GPP; Tech. Spec. Group GSM/EDGE Radio Access Network; Channel Coding*, 3GPP Organizational Partners TS 45.003, Rev. 8.2.0, June 2009.

[7] *3GPP; Tech. Spec. Group GSM/EDGE Radio Access Network; Modulation*, 3GPP Organizational Partners TS 45.004, Rev. 8.0.0, Dec. 2008.

[8] *3GPP; Tech. Spec. Group GSM/EDGE Radio Access Network; Radio transmission and reception*, 3GPP Organizational Partners TS 45.005, Rev. 8.5.0, June 2009.

[9] J. Proakis, *Digital Communications, 4th Edition*. New York, NY, USA: McGraw-Hill Higher Education, 2001.

[10] M. Schmidt and G. Fettweis, "Fractionally-spaced prefiltering for reduced state equalization," vol. 5, 1999.

[11] W. Gerstacker *et al.*, "On prefilter computation for reduced-state equalization," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, Oct. 2002.

[12] J. O. III Smith, *Introduction to Digital Filters: with Audio Applications*. W3K Publishing, 2004.

[13] G. Golub and C. Van Loan, *Matrix computations*. Johns Hopkins Univ Pr, 1996.