An All–Digital Single–Chip Symbol Synchronizer and Channel Decoder for DVB

M. Vaupel, U. Lambrette, H. Dawid, O. Joeressen, S. Bitterlich, and H. Meyr
Lehrstuhl für Integrierte Systeme der Signalverarbeitung

F. Frieling and K. Müller
Siemens AG, Munich, Germany

Aachen University of Technology, ISS–611810, Templergraben 55, D–52056 Aachen, Germany, vaupel@ert.rwth-aachen.de

Abstract

In this contribution, design process and implementation of a single-chip timing and carrier synchronizer and channel decoder for digital video broadcasting over satellite (DVB-S) is described. The device consists of an A-to-D-converter with AGC, timing and carrier synchronizer including matched filter, Viterbi decoder including node synchronization, byte and frame synchronizer, convolutional de-interleaver, Reed Solomon decoder, and a descrambler. The system was designed in accordance with the DVB specifications. It is able to perform Viterbi decoding at data rates up to 56 Mbit/s and to sample the analog input values with up to 88 MHz. The chip allows automatic acquisition of the convolutional code rate and the position of the puncturing mask. The synchronization to the variable sample rates is performed fully digital by means of interpolation and controlled decimation. Hence, no external analog clock recovery circuit is needed. For algorithm design, system performance evaluation, and co-verification of the building blocks an advanced design methodology was used. This guarantees both short design time and high reliability. The chip has been fabricated in a 0.5 μm CMOS technology with three metal layers. A die photograph is presented.

1 INTRODUCTION

A modulation and channel decoding system for digital multi-program television broadcasting is standardized in the digital video broadcasting (DVB) standard (European Telecommunications Institute 1994). The satellite system is intended to provide direct-to-home services for consumer integrated receiver decoders (set-top boxes), as well as cable television head-end stations. For the consumer market inexpensive and reliable implementation solutions are required. Therefore, the goal of the system designer is to implement as much functions as possible on a single chip and to avoid the use of unreliable and expensive analog components. This has been taken into account during the development of the system which is presented here.

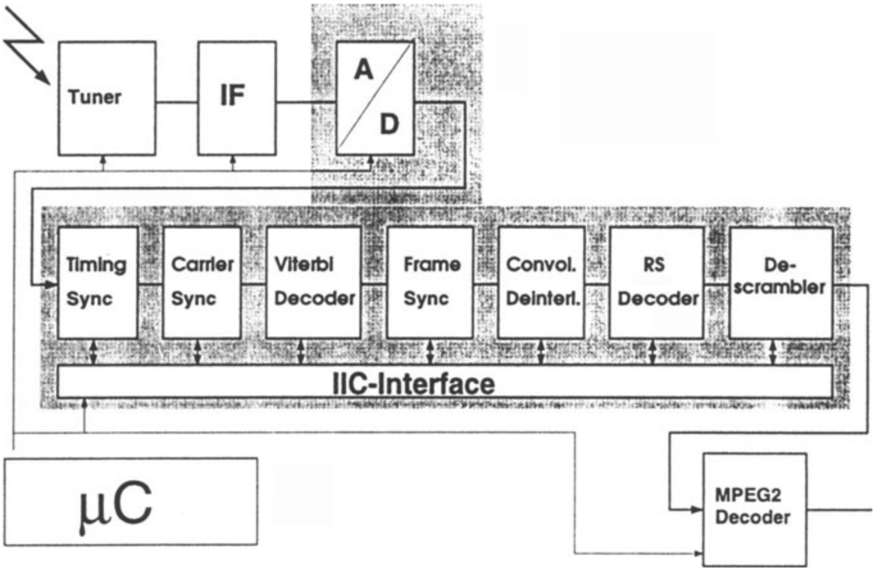

Figure 1 Block diagram of the system

A block diagram of the system is shown in Figure 1. The received LNB-output signal is prefiltered and down converted to the second IF (480 MHz). Filtered with an SAW filter this signal is fed into the demodulator. The demodulated analog I and Q signals are A-to-D converted on chip. Within the timing synchronizer timing offset correction and adaption of the sample rate to the (varying) symbol rate is performed. Carrier phase and frequency offsets are compensated for in the carrier synchronizer unit. The output of the matched filter is input of the de-puncturing unit of the Viterbi decoder which is controlled by a node synchronization unit. After byte and MPEG transport multiplex packet synchronization, the de-interleaved byte stream is fed into the Reed-Solomon (RS) decoder. The decoded information bytes are de-scrambled and put out to the MPEG2 decoder.

For an inner code rate of 1/2 and an E_b/N_0 of 4.2 dB a BER of $2\text{x}10^{-4}$

(behind the Viterbi decoder) is specified in the standard. This corresponds to quasi-error-free operation (BER of 10^{-11}) behind RS-decoding.

Loop parameters, acquisition and tracking performances of all synchronizing units, and even acquisition strategies are configurable via a standardized IIC-bus interface. In addition, internal states and important system information can be read out.

2 SYNCHRONIZATION OF QPSK SIGNALS

The synchronization of PAM signals is a topic well explored and known to the research community. An overview over the subject is given e.g. in (Meyr, Moeneclaey & Fechtel 1997).

In this project, we used two feedback loops for timing and carrier synchronization, which were completely separated. This approach leads to a simpler acquisition strategy, eases the specification of algorithmic parameter ranges and quantization, and increases design robustness. The timing synchronization error feedback loop preceeds the carrier synchronization loop in which the matched filter is embedded. Timing and carrier synchronization are described in more detail in the next two sections.

2.1 Timing Synchronization

Timing synchronization for continuous data streams is performed by a feedback loop consisting of a timing error detector, a loop filter and either a controllable VCO or a digital interpolator. The latter solution has several advantages: It allows to minimize the interaction between analog and digital circuitry (and hence reduces design time and test complexity); it allows to use cheaper analog components and its design is easier to handle as no joint analog and digital modeling technique has to be employed.

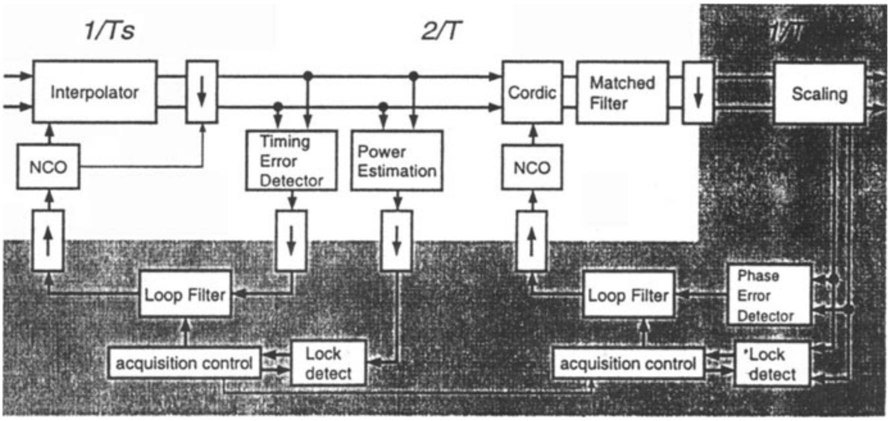

Figure 2 Block diagram of the timing and carrier recovery

In order to achieve a virtually carrier independent timing acquisition, the Gardner Timing Error Detector (Gardner 1986) which is known to produce an error estimate that will lead to timing estimates approaching the Cramer-Rao bound (CRB) (Oerder & Meyr 1987) is used in conjunction with a sampling-rate conversion NCO and a digital interpolator. A more detailed discussion of the developed algorithm for timing synchronization of variable sample rates can be found in (Lambrette, Langhammer & Meyr 1996, Lambrette, Langhammer & Meyr 1997).

The structure of the timing synchronizer loop is depicted in the left-hand side of Figure 2. After interpolation and consecutive decimation, a loop-in-lock criterion is computed within the blocks 'power estimation' and 'lock detection'. This in-lock criterion is evaluated in the acquisition control unit. Steered by the acquisition control unit, the loop filter processes the output samples of the timing error detector. The output of the loop filter is connected to a numerically controlled oscillator (NCO) that provides the interpolator filters with filter coefficients and controls the decimation.

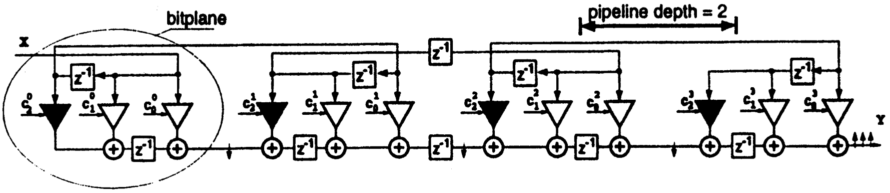

Figure 3 Principle of modified bitplanes

The interpolator consists of two independently operating FIR filters with variable coefficients. These filters are implemented according to the modified bitplane approach (Noll 1987) which yields a small silicon real estate in conjunction with a high sample rate. The number of full adder cells between two consecutive pipeline register cells has been chosen to be 2 in order to gain the smallest possible area while fulfilling the constraints on the data rate. This results in a modified structure (Vaupel & Meyr 1994) compared to (Noll 1987). The principle of the structure is depicted in Figure 3 exemplified by a filter with three taps and a coefficient word length of four bit. By means of re-ordering the add-operations the partial product with the smallest possible values are added up first leading to a smaller word length of the intermediate results. In order to increase efficiency, modified booth encoding of the coefficients is applied.

2.2 Carrier Recovery

Carrier Recovery (Meyr et al. 1997) is based on an NDA phase error detector that feeds a second order loop filter whose output is then passed to a phase rotator. Carrier recovery itself is performed at symbol rate, carrier frequency and phase correction is carried out before the matched filter and hence runs at the sample rate of $2/T$.

The structural block diagram of the carrier synchronizer can be seen in the right-hand side of Figure 2. The output samples of the interpolator are rotated in a CORDIC (Volder 1959, Dawid & Meyr 1996) processor and consecutivly filtered in a matched filter, an FIR filter with fixed coefficients. Since the Viterbi decoder requires a sign-magnitude representation at the input, the two's complement encoded samples are converted in the scaling block. Additionally, the output samples of the matched filter are scaled according to the input requirements of the phase error detector and the carrier lock detection unit. Parameters of the carrier loop filter are set by the carrier acquisition control unit mainly. The output of the loop filter is accumulated in a numerically controlled oscillator that provides the CORDIC processor with the rotation angle.

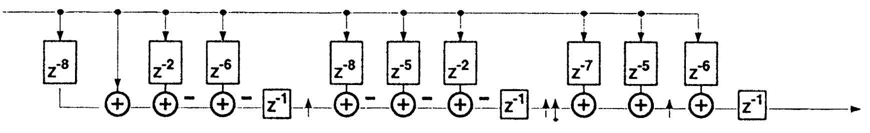

Figure 4 Block diagram of one branch of the matched filter

The complex-valued matched filter is implemented as two identical but independent FIR filters with fixed coefficients which are encoded in canonical signed digit (CSD) format in order to increase efficiency. Exploiting a carry-save representation as internal data format, the filters are implemented as rows of adder cells (bitplanes). Since investigations led to an optimum pipeline depth (the number of additions between two registers) of three, a re-ordering of the bitplanes similar to (Vaupel & Meyr 1994) has been applied to reduce silicon real estate. In order to provide the adder cells with the correctly delayed values, the input samples are delayed in one shift register chain. Figure 4 shows the structural principle.

3 VITERBI DECODER

The Viterbi decoder operates on all DVB compliant code rates (1/2, 2/3, 3/4, 5/6, and 7/8) by means of de-puncturing. It consists of the Viterbi core, a de-puncturing unit, an error correction rate (ECR) measurement unit, and a synchronization controller. The basic Viterbi decoder core consists of a transition metric unit (TMU), an add compare select unit (ACSU) and a survivor memory unit (SMU) with an implemented survivor depth of 128. The de-puncturing unit steers the input FIFO to convert the data rates according to the code rates and performs the actual de-puncturing according to the current synchronization state. It is able to perform a 90 degree rotation of the received QPSK symbol prior to the actual de-puncturing for synchronization purposes. Since up to 4 QPSK symbols belong to one de-puncturing period (for code rate 7/8) an offset is input to the unit to be able to adjust the de-puncturing sequence to possible offsets of the received sequence. The error correction rate (ECR) of the Viterbi decoder, ie the rate of different bits between hard-decisions and the re-encoded data stream, is detected. This rate is an estimate of the hard error rate of the channel and can thus be used to estimate the channel SNR. The synchronization controller performs node synchronization automatically, based on a choice of

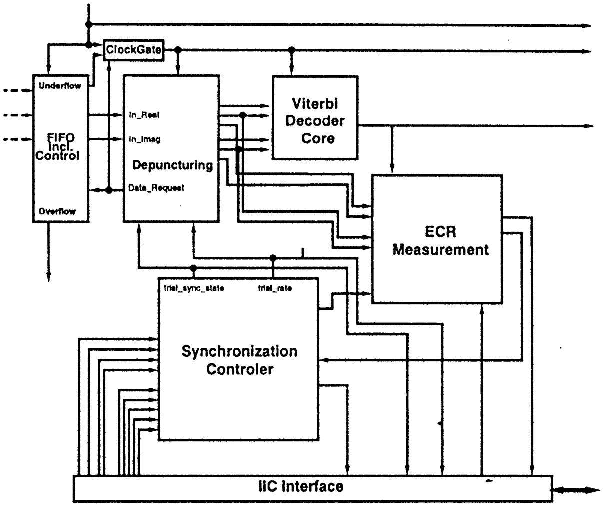

Figure 5 Structural outline of the Viterbi decoder

programmable code rates and thresholds on the correction rate which indicate out-of-sync conditions.

4 FRAME SYNCHRONIZATION AND CONVOLUTIONAL DE-INTERLEAVING

The frame structure of the interleaved data is depicted in Figure 6. An MPEG-2 transport MUX packet consists of 187 information bytes and one leading sync byte (47 hex). The RS-encoder adds 16 byte redundancy to each packet. Each eighth packet (super frame) is indicated by an inversion of the sync byte. On the transmitter side, all data bytes beside the sync bytes are scrambled prior to RS-encoding. This structure of the data stream is exploited in the frame synchronizer to perform

1) byte synchronization of the infinite bit stream
2) frame synchronization, which is needed to synchronize the deinterleaver and the RS-decoder
3) resolving the π-ambiguity of the output data stream of the Viterbi decoder

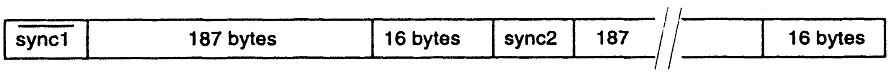

Figure 6 DVB frame structure

The acquisition and tracking performance can be controlled via the IIC bus. It depends on the bit error rate. For a typical parameter set and a BER of $2*10^{-4}$

the mean time until detecting in-sync correctly is below 0.5ms and the mean time until loss-of-sync is above 10^{50}s.

The error protected packets of 204 bytes are interleaved in the transmitter. Therefore, a deinterleaver has to process the byte stream before the RS-decoder is able to decode the packets.

In principle, the deinterleaver is a convolutional interleaver with $I = 12$ branches (Forney 1971, Ramsey 1970). Each branch consists of a shift register with $M * (11 - j)$ cells ($M = 17$, j branch index). Each register has a wordlength of eight bit. The data are (de)interleaved byte–wise. For synchronization purposes, the (inverted) sync bytes are always routed to branch "0" of the deinterleaver (see Figure 7). Due to the large consumption of silicon real estate, implementing the deinterleaver using register cells would be very inefficient. Instead, a RAM-based solution was implemented. In order to obtain the minimal possible memory size, an addressing scheme was developed that allows in-place updating.

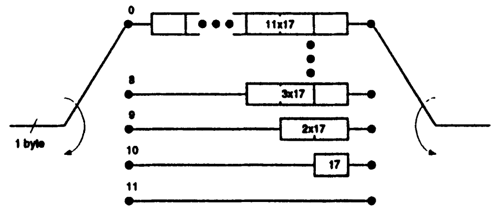

Figure 7 De-interleaving scheme

5 REED SOLOMON DECODER

The DVB standard specifies a shortened (204, 188) Reed Solomon (RS) code. One codeword consists of 204 bytes, separated into 188 information bytes and 16 parity check bytes. Since errors-only decoding is employed (no erasure processing), the RS decoder is able to detect and correct up to $t = 8$ byte errors per codeword (a byte error specifies an erroneous byte, independent of the number of corrupted bits), which can be arbitrarily distributed within the data and check locations in a codeword. This code is designed to achieve QEF (quasi error free) performance. The code is characterized by the code generator polynomial

$$g(z) = \prod_{i=m_0}^{m_0+d-2} (z - \alpha^i)$$

with $m_0 = 0$ as specified in the DVB standard. The DVB Reed Solomon Decoder (RS) uses a finite field GF(2^8) which is specified in the DVB standard by the field generator polynomial $f(x) = x^8 + x^4 + x^3 + x^2 + 1$. For the DVB application the "classical" method, given by syndrome calculation in the frequency domain and calculation of the error locator and evaluator polynomials using the Bérlekamp-Massey algorithm, is considered to be optimal.

The whole decoding process, which has to be performed for each codeword, can then be coarsely divided into the following steps:

- Syndrome calculation
- Calculation of the Error Locator and Evaluator Polynomials
- Chien Search (Determination of the roots of the Error Locator Polynomial)
- Calculation of the correction values
- Correction and output of the codeword

These steps are reflected in the top level structure which is shown in Figure 8. Due to the high throughput requirements, every block is implemented as a separate hardware unit.

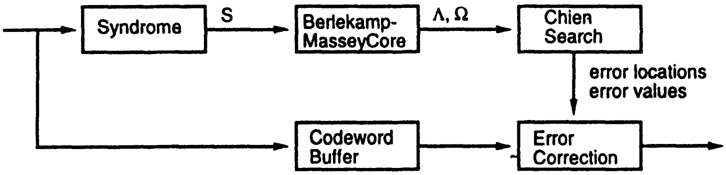

Figure 8 Block diagram of RS decoding architecture

Given a syndrome, a time budget of 204*4 = 816 clock cycles is available for solving the key equation using the Berlekamp–Massey Algorithm. In order to minimize area consumption while meeting this throughput constraint, a special ALU supporting Galois field arithmetic was developed (see Figure 9).

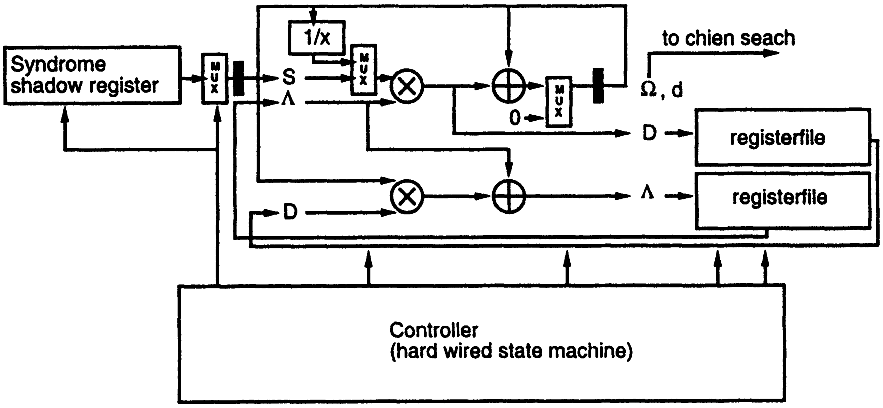

Figure 9 Galois Field ALU

The polynomial coefficients are stored intermediately in two register files, one for the Ω and one for the Λ polynomial. A large hard-wired state machine steers the operations in the ALU and the register files. This design approach leads to a highly efficient implementation of the Berlekamp-Massey algorithm, implementing exactly the amount of parallel processing necessary to meet the given throughput constraint. The input data which is stored in a dual port RAM (the codeword buffer) is finally read out and corrected.

6 DESIGN METHODOLOGY

All algorithm design and system performance evaluation was performed using the system level design tool COSSAP (Synopsys 1996). The performance of the frame, carrier and timing synchronizer was calculated partly in conjunction with Matlab (The MathWorks 1994, Lambrette, Schmandt, Post & Meyr 1995). Only the synchronizers for timing, carrier, node sync and frame start required specification of algorithms and fixing algorithm–specific parameters (loop–bandwidth, threshold values), designing the remaining blocks did not require major algorithmic investigations.

For each of the building blocks the environment was modeled. System simulations paved the way from floating point to quantized integer models. The VHDL descriptions of the components were verified against the corresponding system level COSSAP blocks using a coupling of the system simulator and the VHDL simulator (Zepter 1993*b*, Zepter 1993*a*). Therefore, no VHDL testbench had to be written in the course of this project. This led to significant savings in design time compared to a more conventional HDL–based methodology.

COSSAP was well suited to the modeling of the dynamic data-dependend data flow, imposed by the controlled decimation in the timing synchronizer. In hardware, this dynamic dataflow was realized using gated clocks. Corresponding to the three different sample rates 1/T$_S$, 2/T, and 1/T (see Figure 2) we have three different clock domains in the symbol synchronizer. (A fourth clock domain drives the viterbi decoder and the consecutive units.) For synchronization purposes between these domains adjacent clocks are negated. That means that each transition from 'low' to 'high' of the clock corresponding to $2/T$ occurs on a falling edge of the clock corresponding to $1/T_S$ only. The COSSAP system model takes this into account. Therefore, for the presented system a hierarchical COSSAP model exists which is bittrue and cycle–true identical to the VHDL model.

Using the COSSAP design flow, seamless design verification was possible throughout all design stages. Synopsys' Design Compiler was used for logic synthesis of the RTL VHDL code. Test pattern for the resulting gate level netlist and even for post production testing were also generated from COSSAP.

7 PERFORMANCE

For a symbol rate of R = 33MHz, frequency offsets of ±12.5% normalized to the symbol rate and a typical parameter setting, the acquisition time of the carrier synchronizer is below 20 ms and the acquisition time of the timing synchronizer below 2ms. The mean acquisition time for the frame synchronizer is about 0.5 ms.

In order to assess the performance of the synchronizer, bounds must be established for the performance of the synchronizers as well as for the overall system.

The most important measure for the performance is the bit error ratio which is measured behind the Viterbi decoder that follows the synchronizer. Usually, an ideal implementation reaches the theoretical bounds of the error ratio. Any degradation from this bound is due to implementation effects like quantization or clipping. A detailed performance analysis also relating the bittrue synchronizer performance to the Cramer–Rao bound can be found in (Lambrette et al. 1996, Lambrette et al. 1997, Meyr et al. 1997).

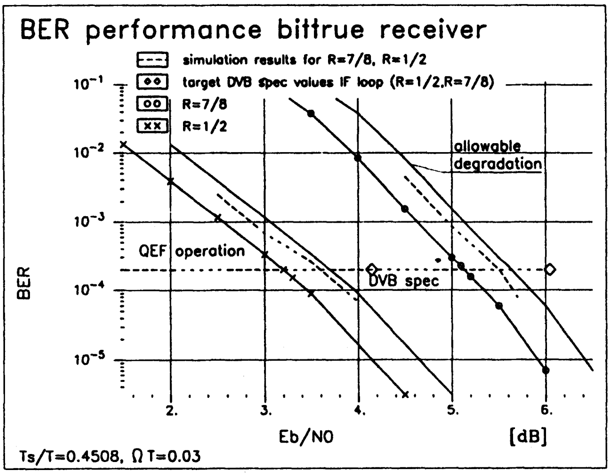

Figure 10 Bittrue Receiver Model Performance

Figure 10 displays the resulting Bit Error Ratio as a function of the E_b/N_0. In the figure, the ideal implementation (no synchronization errors, no implementation loss due to survivor path truncation or quantization) is compared to the bittrue model of the receiver also including impairments of the A/D converter. The overall degradation is about 0.4dB, leaving 0.6dB implementation loss for the other analog circuitry.

8 IMPLEMENTATION

The chip was implemented in a 0.5 μm CMOS technology with three metal layers. The single supply voltage is 3.3 V. The power consumption amounts to 1.2 W at a maximum sampling rate of the analog input values of 88 MHz. The maximum output bit rate figures up to 56 Mbit/s. In Table 1 the relative standard cell areas of the main components and the normalized silicon areas including analog components and RAM are summarized:

Figure 11 shows a chip photograph. The data flow direction is from left to right. On the upper left corner the two clock synthesizer Plls for the synchronizer and the channel decoder are located. Below these, the A/D-converters for in-phase and quadrature component were placed. The memory blocks in the middle are the RAM's of the survivor memory unit which enclose the Viterbi decoder. On the right hand side, the memories for the deinterleaver (at the top) and the Reed-Solomon decoder (bottom) can be seen.

Table 1 Cell and chip areas

Component	accum. cell area	silicon area
Synchronizer	32 %	9 %
Viterbi	40 %	48 %
RS-decoder*	28 %	17 %
A/D-converter		4 %
Clock-Pll		1 %
Pad frame		21 %
Sum	100 %	100 %

*incl. deinterleaver, frame sync, and descrambler

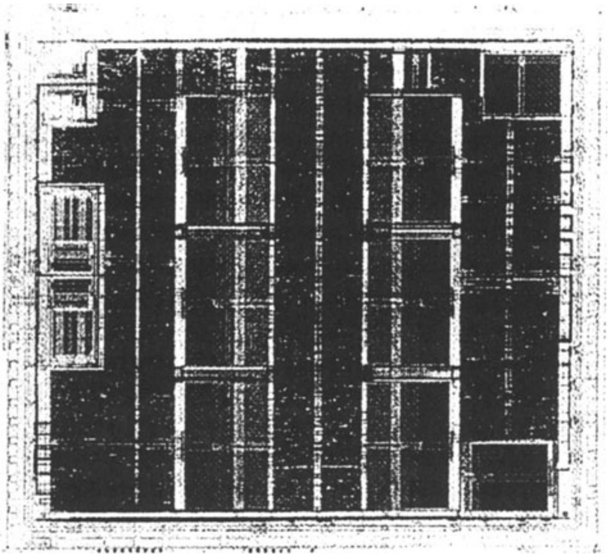

Figure 11 Chip photograph

9 CONCLUSION

The implementation of a single-chip timing and carrier synchronizer and channel decoder for digital video broadcasting over satellite (DVB-S) was described. Due to the digital timing and carrier synchronization, the number of external components has been minimized. The chip is fully compliant with the DVB-standard and allows automatic acquisition of variable symbol rates and convolutional code rates. The design methodology presented ensures both short time to market and high design integrity.

REFERENCES

Dawid, H. & Meyr, H. (1996), 'The Differential CORDIC Algorithm: Constant Scale Factor Redundant Implementation without correcting Iterations', *IEEE Transactions on Computers* **45**(3).

European Telecommunications Institute (1994), Digital Broadcasting System for Television, Sound and Data Services; Framing Structure, Channel Coding and Modulation for 11/12 GHz Satellite Services, Draft DE/JTC-DVB-6, ETSI Secretariat, 06921 Sophia Antipolis - France.

Forney, G. D. (1971), 'Burst-correcting codes for the classic bursty channel', *IEEE Transactions Communications* **COM-19**, 772–781.

Gardner, F. (1986), 'A BPSK/QPSK Timing-Error Detector for Sampled Receivers', *IEEE Transactions on Communications* **COM–34**(5), 423–429.

Lambrette, U., Langhammer, K. & Meyr, H. (1996), Variable Sample Rate Digital Feedback NDA Timing Synchronization, *in* 'Proceedings of the IEEE Global Telecommunications Conference GLOBECOM'.

Lambrette, U., Langhammer, K. & Meyr, H. (1997), 'An aliasing-free Receiver with Variable Sample Rate Digital Feedback NDA Timing Synchronization', *J. Wireless Personal Communciations* .

Lambrette, U., Schmandt, B., Post, G. & Meyr, H. (1995), COSSAP - MATLAB Cosimulation, *in* 'Proc. Int. Conf. on Signal Processing Application and Technology'.

Meyr, H., Moeneclaey, M. & Fechtel, S. (1997), *Digital Communication Receivers*, John Wiley & Sons. in preparation.

Noll, T. G. (1987), Semi-Systolic Maximum Rate Transversal Filters with Programmable Coefficients, *in* W. M. et.al, ed., 'Systolic Arrays', Adam Hilger, Bristol, pp. 103–112.

Oerder, M. & Meyr, H. (1987), 'Derivation of Gardner's timing error detector from the maximum likelihood principle', *IEEE Transactions on Communications* **COM-35**, 684–685.

Ramsey, J. L. (1970), 'Realization of Optimum Interleavers', *IEEE Transactions on Information Theory* **IT-16**(8), 338–345.

Synopsys (1996), *COSSAP Overview and Documentation Roadmap*, Synopsys, Inc., 700 E. Middlefield Rd., Mountain View, CA 94043, USA.

The MathWorks (1994), *MATLAB Reference Guide*, Cochituate Place, 24 Prime Park Way, Natick, Mass. 01760. Documentation of MATLAB 4.2.

Vaupel, M. & Meyr, H. (1994), High speed FIR-filter architectures with scalable sample rates, *in* 'Proceedings of the IEEE International Symposium on Circuits and Systems', IEEE, London, pp. 4.127–4.130.

Volder, J. E. (1959), 'The CORDIC trigonometric computing technique', *IRE Trans. Electronic Computing* **EC–8**(3), 330–34.

Zepter, P. (1993*a*), Kopplung eines VHDL Simulators an einen Simulator für Signalverarbeitungsalgorithmen, *in* D. Seitzer, ed., 'GME Fachberichte 11 Mikroelektronik', VDE Verlag, pp. 127–132. in german.

Zepter, P. (1993*b*), Simulator Coupling: COSSAP - Synopsys VSS, Internal Memo 715/16, ISS, RWTH Aachen.