IP CELL FOR ERROR CORRECTION SYSTEMS IN MINIDISC

A. De Gloria, D. Grosso, M. Oddicini and F. Sciutto
Department of Biophysical Electronic Engineering (DIBE)
University of Genoa ITALY
E-mail: adg@dibe.unige.it

Abstract: *An error correction system based on the Advanced Cross Interleave Reed Solomon Code (ACIRC) has been studied and implemented. We modelled the whole structure of a Minidisc (MD) player through a mathematical simulator. We tested our hardware implementation of the ACIRC decoder inserted in the global structure through cosimulations.*

The final result has been the design of a re-usable firm core implementing the functions of the ACIRC decoder block.

1 Introduction

The MINIDISC (MD) is one of the last development in digital audio; it was introduced in 1992 to replace the tape recorder system. It combines the recording features of the traditional cassette with the sound quality and the random access capabilities of the Compact Disc (CD). Thanks to the magneto-optical technology, recordable MDs allow up to 1 million recording cycles without any deterioration of the sound quality. In addition the MD system with the use of a solid state semiconductor memory overcomes the main weakness of the CDs that is their high sensitivity to vibrations; this fact makes MD players well suited for use in outdoor environment.

Each MD contains the same amount of audio data as a standard audio CD in a very smaller physical support; this result is obtained by exploiting the ATRAC (Adaptive TRansform Acoustic Coding) [1] compression technique designed to be used in MD.

One of the factors that affect the performance of the minidisc system is the presence of errors that occur in reading data from the disc. In order to prevent and correct these errors a system of correction is used in MD. It consists of the combination of the Eight to Fourteen Modulation (EFM) [2,3,4] with an Advanced version of the Cross Interleave Reed Solomon Code (ACIRC) [2,3,4].

This paper deals with the system level modeling and simulation of a MD player , with particular care to the error correction system, and

Contributed Paper
Original manuscript received November 15, 1999

presents an efficient VLSI implementation of the ACIRC decoder block that represent the main factor of hardware and performance cost. The final result of our work was the design of a re-usable firm core implementing the functions of the ACIRC decoder block.

In the first part of the project the system functionality has been verified through behavioural simulations.

Then a mixed level simulation has been realized by inserting the optimized VHDL code of the ACIRC decoder in the behavioural model. At this point, we developed an exhaustive verification of our design by exploiting the cosimulation feature of the SPW product. When a cosimulation is initiated the code of the inserted block runs on the HDL simulator while SPW processes the regular proprietary blocks. In this way we verified the correct functionality of our digital implementation of the ACIRC block running together with the behavioral modules in the modelled environment. Moreover the cosimulation allows one to evaluate the effect of non-ideality introduced when the module is implemented in VHDL.

The second part of the project was concerned with the synthesis to the gate level of the structural VHDL description of the ACIRC decoder by means of a standard logic synthesizer (Synopsys tool set). The design was mapped on an industrial 0.35 μm CMOS standard-cell technology[†].

2 Technical Background

The MD is a system of recording and reproduction of digital audio data. The input of the whole system is the analog audio data that is A/D converted and then compressed by the ATRAC encoder in order to obtain a compression ratio of about 1 to 5. At this point the ACIRC/EFM encoder processes the compressed data to make it suited to be recorded. In case of premastered disc the data is recorded only once and it is represented by pits on the reflective layer of the disc; in recordable MD the data is stamped through the magnetic recording head that is represented by the north or south magnetization.

Anyway the EFM/ACIRC encoder performs the same signal processing in both the above cases.

The ACIRC allows the MD player to correct most errors that occurs on the disc. This encode method performs interleaving and adds parity symbols

[†] Corelib HCMOS6 from ST Microelectronics.

to the data stream. It consists of CIRC with additional interleave used to improve the resistivity from the burst error on the disc.

After ACIRC encoding the Eight to Fourteen Modulation is performed; it encodes the data in such a way that it can be properly recovered from the disc. The EFM belongs to the Run Length Limited (RLL) codes [8,9] designed to limit the distance between transitions in a digital word. It takes as input an 8 bit word and convert it to a 14 bits codeword that is compliant to the following constraints

- 2 is the minimum number of '0' code bits between adjacent '1'
- 10 is the maximum number of '0' code bits between adjacent '1'.

The first constraint is set to avoid the presence of small pits that cannot be correctly detected by the optical pick-up. The second constraint is set to ensure the proper extraction of the read clock. In fact, during readout the only available information is the audio signal while the channel bit rate has to be reproduced by using digital PLL. The second constraint is necessary to ensure that synchronizing information for the digital PLL is regularly available in the waveform extract from the MD.

The same optical pickup unit is used to read out data from both premastered and recordable disc. In fact the optical detecting unit is able to detect the pit signal as well the magnetic signal. The recovered data is then EFM demodulated and error decoded in the ACIRC decoder. Finally from the ATRAC decompression and from the A/D conversion the original analog audio signal can be recovered.

In the next sections we present our study about the MD player's error correction system; this work has been performed through the development of a model and of the exhaustive simulation of it. The final result has been the design of a re-usable firm core implementing the functions of the ACIRC decoder block.

3 System modeling and simulation

The data path structure of a MD player is shown in Fig. 1.

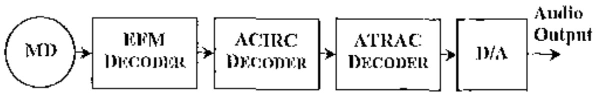

Figure 1 Block diagram of the MD player system

The model created with SPW describes the behaviour of the EFM demodulator, of the ACIRC decoder and the ATRAC decoder block and provides a simple representation of the disk through the model of the typical reading errors.

The EFM demodulator takes as input the bit stream coming from the optical pickup and creates the codeword for the ACIRC decoder by performing the 14 to 8 bits conversion.

We focus our attention on the ACIRC decoder because it represent the main factor of hardware and performance cost. Moreover the ACIRC decoder is an advanced version of that used in CD drive [4] whereas the EFM decoder has the same specifications used in error correction codes for CD drive. The algorithm performed by this decoder tries to compensate eventual errors that can occur during data reading; its input is called frame and consists of a set of 32 symbols (8 bits per symbol): 24 information symbols and 8 symbols of parity. This decoder is able to correct up to 16 consecutive wrong frames; its block diagram, shown in Fig. 2, is composed of five main parts: three stages of interleave and two Reed-Solomon decoders, called C1 and C2. Interleaving improves the resistance of coded information to the burst errors. by performing the scrambling of the symbols and then converting burst errors to random errors. In this application the Reed Solomon decoder takes as input symbols defined by the Galois Field ($GF(2^8)$) [2,3] generated by $F(x)=x^8+x^4+x^3+x^2+1$ [4].

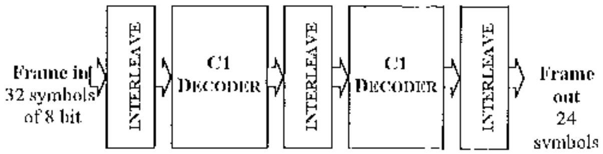

Figure 2 Block diagram of ACIRC decoder

The C1 decoder uses four symbols of parity (P-parity) to locate and correct up to one error over the 28 information symbols (Reed Solomon (32,28)). This decoder is able to correct up to two errors but in this application, according to the standard of CD Audio [4], it corrects only one error. The error correction involves the following operations: syndromes calculation, locator and corrector calculation and single-error correction. The syndromes Sj_p, that are used to determine if data contains errors are computed as shown in (1).

$$Sj_p=\sum_{i=0}^{31}\alpha^{j(31-i)}W_i \qquad (1)$$

where

W_i : inputs of the C1 decoder

α : primitive element of the Galois field

j = 0,1,2,3

If no errors occur all the syndromes are equal to zero; if one or more is not equal to zero, there have been one or more errors. If only one error occurs, (1) can be solved and in particular, from the first equation (j=0) one can get the corrector $E_i=S_{0p}$ and from the second (j=1) one can get the locator *i*. In this case the corrected symbol is found with :

$$W_{i,correct}=W_i+E_i \qquad (2)$$

If more than one error occurs, (1) cannot be solved and it is not possible to perform error correction. In this case the C1 decoder marks all the symbols as wrong. The output of the C1 decoder is scrambled and then passed to C2.

The C2 decoder uses a Reed Solomon (28,24) code, with four symbols of parity (Q-parity); the syndromes are computed as shown in (3).

$$S_{jq}=\sum_{i=0}^{27}\alpha^{j(27-i)}W_i \qquad (3)$$

If no errors occur all the syndromes are equal to zero. In case of error the correction by-erasure technique [2,3] is used. The second decoder is able to recover up to N=4 wrong symbols by using the 4 correctors obtained as solutions of (3). The wrong symbols have been marked by C1 thus the information about the errors position is immediately recovered by the *test flag* block shown in Fig. 3. The *correctors* block decides if the correction can be applied by following the flow chart shown in Fig. 4; in addition it finds the 4 correctors E_i, E_j, E_k, E_h by solving (3) as shown in (4).

Finally the last block performs the correction by adding the corrector just computed to the wrong symbol as shown in (2).

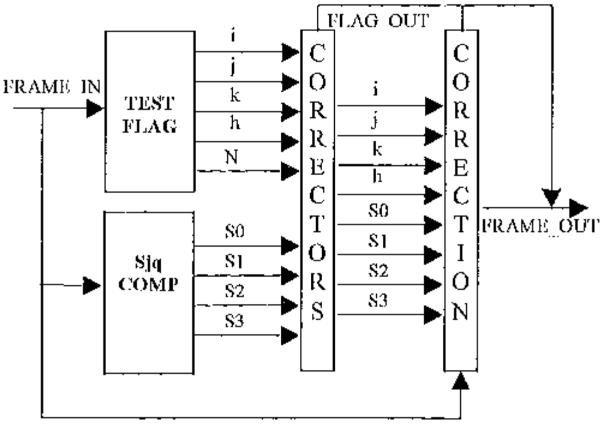

Figure 3 Block diagram of the C2 decoder

If more than 4 symbols are marked as wrong the C2 decoder is not able to perform the correction; in this case the frame is put in output without any processing. The output of the second decoder enters the last interleaving stage; this block descrambles the symbols in such a way that all the frames are subject to the same delay within the ACIRC block.

The output of the ACIRC decoder is dequantized and decompressed by the ATRAC decoder in order to provide the standard audio bit rate.

In order to simulate the whole system shown in Fig. 1 we realized a model of the physical media; in particular the mini disc has been represented as a noisy channel.

$$E_H = [A\varphi_{I,J}S3_q + C\varphi_{I,J}S2_q + \left(A\varphi_{3I,3J} + C\varphi_{2I,2J}\right)S1_q + $$
$$+\left(A\varphi_{3I+J,I+3J} + C\varphi_{2I+J,I+2J}\right)S0_q]/D$$

$$E_K = [BE_H + \varphi_{I,J}S2_q + \varphi_{2I,2J}S1_q + \varphi_{I+2J,2I+J}S0_q]/A \quad (4)$$

$$E_J = [\varphi_{I,K}E_K + \varphi_{I,H}E_H + S1_q + \alpha^{I}S0_q]/\varphi_{I,J}$$

$$E_I = E_J + E_K + E_H + S0_q$$

with

$I,J,K,H = 27\text{-}i,\ 27\text{-}j.\ 27\text{-}k.\ 27\text{-}h$

$$\varphi_{m,n} = (\alpha^m + \alpha^n)$$

$$A = \alpha^{2K}\varphi_{I,J} + \alpha^{2J}\varphi_{I,K} + \alpha^{2I}\varphi_{J,K}$$

$$B = \alpha^{2H}\varphi_{I,J} + \alpha^{2J}\varphi_{I,H} + \alpha^{2I}\varphi_{J,H}$$

$$C = \varphi_{I,K}\varphi_{3I,3J} + \varphi_{I,J}\varphi_{3I,3K}$$

$$D = A(\varphi_{I,H}\varphi_{3I,3J} + \varphi_{I,J}\varphi_{3I,3H}) + CB$$

Thus through our model it is possible to insert the typical errors of the digital audio reproduction systems: random errors and burst errors of desired length. The system level simulation allowed us to verify the correct functionality of the whole data path section of the mini disc player. In addition the ability of correcting both random errors and burst errors has been tested. In this context we realized simulations in order to find the maximum length of a sequence of wrong bits (burst error) that can be corrected by the ACIRC decoder. It is useful to describe the two limit cases. When a sequence of 8820 wrong bits (15 wrong frames) enters the EFM decoder, thanks to the scrambling performed by the interleave stage between the two decoder, no more than 4 wrong symbols enters the C2 decoders at the same time; thus the correction can be applied. When a sequence of 9408 wrong bits (16 frames) is inserted more than 4 wrong symbols enters the C2 decoders at the same time; thus the decoder is not able to correct the recovered audio data. From this tests we deduced that the system presented here allows one to recovery a superficial defect of about 2 mm.

After having inserted the structural VHDL description of the ACIRC decoder in the system shown in Fig.1 the same functional tests have been repeated by performing a cosimulation. In this way we developed a first verification of our digital design; besides we performed an estimation of how the non-ideal digital parts affect
the behavior of the whole system and if their delay or the reduced precision of arithmetic operations could be critical or not.

Cosimulations are very useful because allows one to verify the behaviour of digital blocks inserted in the environment it will have to work in. With the term environment we identify all the context the system operates in and then we include different operating conditions settings and situations that can happen during system functioning.

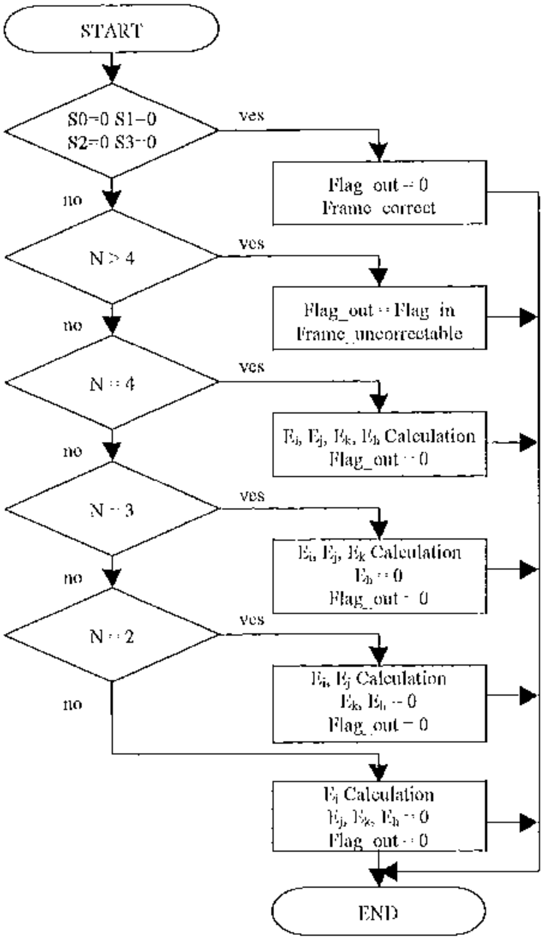

Figure 4 The strategy used to apply the correction

4 System architecture

The architectural structure of the ACIRC decoder is shown in Fig. 2. We developed the design of the two decoder and we used an external RAM for implementing the three interleaving stages.

The main clock of the ACIRC decoder is called *frame_clk* (T_{frame_clk}=136μs); both the C1 and the C2 decoder computes the syndromes during the high semi-period of *frame_clk* and performs the symbol correction during the low semi-period of *frame_clk*. Both these operations are performed in 32 steps; thus we use a second signal (*clk*), at least 64 times faster than *frame_clk*, in order for these operations to be clocked.

Fig. 5 shows the circuit used to compute the syndromes [7] by following (1).

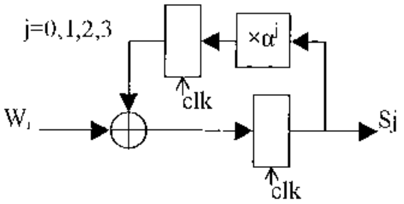

Figure 5 Circuit for syndromes computation

The input of the circuit is the sequence of symbols belonging to a frame; every *clk* a new symbol W_i is added to the partial result stored in the register; after 32 clocks cycles the syndromes have their definitive value.

According to the rules of the Galois Field, the sum operation is made with a bit to bit exor. The multiplication $\times\alpha$, $\times\alpha^2$, $\times\alpha^3$, are implemented by using the circuits shown in Fig. 6, directly derived from the polynomial generator of the code

$F(x)=x^8+x^4+x^3+x^2+1$.

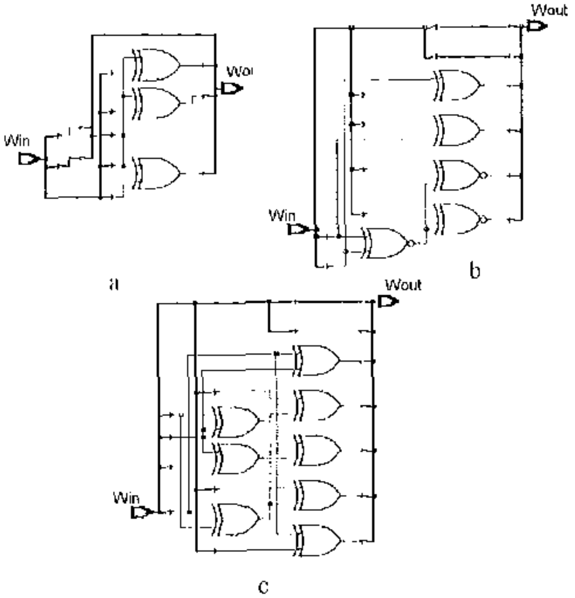

Figure 6 Multiplier circuits: a) ×α; b) ×α²; c)×α³

Once verified that at least one syndrome is not zero thus that there is at least one error the C1 decoder begins applying the correction. It gets the corrector from the first equation of (1) and the locator *i* through (5) derived from (1). We implemented equation (5) by using the circuit shown in Fig. 7.

$$Sj_p = \alpha^{ji} S0 \qquad (5)$$

with j=1,2,3.

The signal *start* selects S_0 during the first *clk* and then it always selects the input from the feedback signal. During each *clk* cycle this circuit verifies (5) for a new value of *i (i = 0...31);* the locator is found when *error=1.*

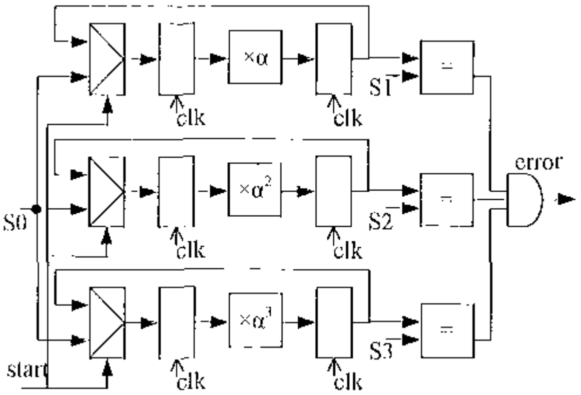

Figure 7 Circuit to find the locator

If after 32 *clk i* that satisfy (5) it has not been found, it means that more than one error has occurred and thus the frame is marked as wrong.

This architecture allows one to re-use the same multipliers already used in the first part of *frame_clk* for the syndromes calculation; this results in a good save of area.

Once known the locator *i* and the corrector E_i the correction can be applied by following (2).

The correction stage of C2 decoder is more complex, due to the necessity of computing four correctors (E_i, E_j, E_k, E_h) as shown in (4).

In the high semi period of *frame_clk*, we compute the syndromes and we test the flag associated to the input symbols in order to extract the number of errors in the current frame and to find their position.

During the low semi period of *frame_clk* a high number of operations between elements of

Galois Field has to be performed. In this case we chose to use the logarithmic technique to perform multiplications and divisions In this technique the operands are converted into exponents of the corresponding Galois Field elements through two look-up tables (Log table); these are module 255 summed (multiplication) or subtracted (division) and then converted with the inverse table (Antilog table) into the element that represent the result of the operation. Form Fig. 8, showing the structure of a multiplier, one can understand that at least two clock cycles are required to perform either a multiplication or a division.

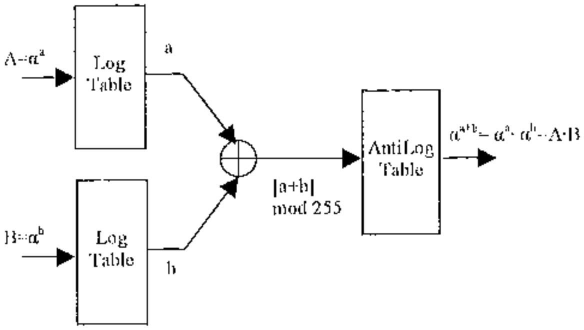

Figure 8 Scheme of the Galois Field multiplication

In order to perform the big number of operations necessary to solve (4) in a single *frame_clock* cycle we implemented a pipelined architecture. In this way during each *clk* cycle a new multiplication is performed and (4) can be solved in 46 *clk* cycles. At this point the correction can be applied by following (2).

The architecture implemented for this stage uses 4 tables (2 log and 2 antilog tables), 2 addition and 1 subtraction circuits for the operation between exponents and, finally, 21 pipeline registers to store the partial results.

5 Results and conclusions

We proceeded our development works with the final target of development of MD error correction system. The target was reached starting with the system level modelling and simulation of the whole MD player and passing through a mixed level model; this methodology allowed us to perform a precise and reliable validation of our digital implementation. The final result is a re-usable intellectual property that can be re-mapped to any customer's technology and included in a larger custom design. The core architecture has been placed and routed automatically to obtain a sample implementation layout in 0.35 um CMOS.

The core layout is composed of 21 equivalent K gates and occupies less than 2 mm^2 .

6 Acknowledgment:

We are grateful to ST Microelectronics for the technology and CAD support.

References

[1]. J. Maes, "*The Mini Disc*", Focal Press Nov. 1996

[2]. Ken C.Pohlmann, "*Principle of Digital Audio*", 3rd Edition, Mc Graw Hill 1995.

[3]. John Watkinson, "*The Art of Didital Audio*", 2nd Edition Focal Press 1994.

[4]. "*IEC 908 standard for digital audio on compact discs*" Feb. 1999 http//:www.iec.ch.

[5]. *Signal Processing Worksystem User's Guide*, Alta Group, Cadence http//:www.cadence.com.

[6]. Synopsys Inc., Command Reference Manual, Version 98.08 http//:www.synopsys.com

[7]. T. K. Matsushima, T. Matsushima, S. Hirasawa, "*Parallel Encoder and Decoder Architecture for Cyclic Codes*", IEICE Trans. Fundamentals, vol. E79-A, no. 9 September 1996

[8]. P.H. Siegel, "*Recording Codes for Digital Magnetic Storage*" IEEE Trans. on Magnetics, vol. MAG-21, no. 5, Sept. 1985.

[9]. P.A. Franasezek, "*Sequence State Methods of Run-Length-Limited Coding*" IBM J. of R&D, vol. 14, July 1970.