Full-parallel architecture for turbo decoding of product codes

C. Jégo, P. Adde and C. Leroux

A full-parallel architecture for turbo decoding, which achieves ultra-high data rates when using product codes as error correcting codes, is proposed. This architecture is able to decode product codes using binary BCH or m-ary Reed-Solomon component codes. The major advantage of our architecture is that it enables the memory blocks between all half-iterations to be removed. Moreover, the latency of the turbo decoder is strongly reduced. The proposed architecture opens the way to numerous applications such as optical transmission and data storage. In particular, the block turbo decoding architecture can support optical transmission at data rates above 10 Gbit/s.

Introduction: In recent years turbo codes [1] have been adopted by several digital communication applications. They are particularly attractive to increase transmission rates and/or to guarantee the Quality of Service (QoS). Currently, research is under way to use turbo codes to protect data stored on hard drive or DVD and in fibre optical transmission. The earliest FEC for optical communication [2] employed the well-known Reed-Solomon (RS) codes to recover the errors. This solution is attractive to increase transmission rates and the data rate. However, it is necessary to process these data by row and column decoding. Let \( m \) be the number of decoding iterations. In this case, the latency between row and column decoding is null.

Previous work: Many block turbo decoder architectures have previously been designed. The classical approach involves decoding all the rows or all the columns of a matrix before the next half-iteration. When an application requires high-speed decoders, an architectural solution is to cascade soft input soft output (SISO) elementary decoders for each half-iteration. In this case, the memory blocks are necessary between each half-iteration to store channel data and extrinsic information. Each memory block is composed of four memories of \( qn^2 \) symbols where \( q \) is the number of bits to quantify the matrix symbols. Thus, duplicating a SISO elementary decoder \( (e_{dec}) \) results in duplicating the memory block which is very costly in terms of silicon area. In 2002, a new architecture for turbo decoding product codes was proposed [7]. The idea is to store several data at the same address and to perform parallel decoding to increase the data rate. However, it is necessary to process these data by row and by column. Let us consider \( m \) adjacent rows and \( m \) adjacent columns of the initial matrix. The \( m^2 \) data constitute a word of the new matrix that has \( m^2 \) times fewer addresses. This data organisation does not require any particular memory architecture. The results obtained show that the turbo decoding throughput is increased by \( m^2 \) when \( m \) elementary decoders processing \( m \) data simultaneously are used and its latency is divided by \( m \). The area of the elementary decoders (\( m-e_{dec} \)) is increased by \( m^2/2 \) while the memory is constant.

Full-parallel turbo decoding principle: The codewords of all rows (or all columns) of a matrix can be decoded in parallel. If the architecture is composed of \( n \) elementary decoders, an appropriate treatment of the matrix enables the elimination of the reconstruction of the matrix between each decoding. Let \( i \) and \( j \) be the indices of a row and a column of the \( n^2 \) matrix. In full-parallel processing, the row decoder \( i \) begins the codeword decoding by the symbol in the \( ij \)th position. Moreover, each row decoder processes the codeword symbols by increasing the index by one modulo \( n \). Similarly, the column decoder \( j \) begins the codeword decoding by the symbol in the \( j \)th position. In addition, each column decoder processes the codeword symbols by decreasing the index by one modulo \( n \). Therefore only one time cycle is necessary between two successive matrix decoding operations. The full-parallel decoding of a \( n^2 \) product code matrix is detailed in Fig. 1. A similar strategy was previously presented in [8]. In this case, the conflicts of \( n \) independent RAM memories are eliminated by the appropriate treatment of the matrix. The elementary decoder latency, \( L_e \), can be defined as the symbol number processed by the decoder during the decoding of one symbol. This latency depends on the structure of the elementary decoder and the codeword length \( n \). As the reconstruction matrix is removed, the latency between row and column decoding is null.

Full-parallel turbo decoder for product codes: The major advantage of our full-parallel architecture is that it enables the memory block of \( 4m^2n^2 \) symbols between each half-iteration to be removed. However, the codeword symbols exchanged between the row and column decoders have to be switched. One solution is to use a connection network for this task. In our case we have chosen an Omega network. The Omega network is one of several connection networks used in parallel machines [9]. It is composed of \( n \) stages, each having \( n/2 \) exchange elements. In fact, the Omega network complexity in terms of number of connections and of \( n \) switch transfer blocks is \( n \log_2 n \) and \( (n \log_2 n)/2 \), respectively, e.g. the equivalent gate complexity of a \( 32 \times 32 \) network can be estimated to be 200 per exchange bit. The proposed full-parallel architecture for product codes is presented in Fig. 2. It is composed of cascaded modules for the block turbo decoder. Each module is dedicated to one iteration. However, it is possible to process several iterations by a same module. In our approach, \( 2n \) elementary decoders and two connection networks are necessary for one module. In fact, the full-parallel turbo decoder complexity essentially depends on the complexity of the elementary decoder. Table 1 gives a comparison of our architecture proposal with the previous solutions in terms of mean features. The features depend on different parameters: codeword length \( n \), number of decoding iterations \( it \), elementary decoder throughput \( D_{th} \), elementary decoder latency \( L_e \), number of symbol quantisation bits \( q \) and number of adjacent symbol groups \( m \). The \( e_{dec} \) and \( m-e_{dec} \) architecture types correspond to the classical solution and the solution in [7], respectively.

Towards implementation of architectures for ultra-high rates: By applying the full-parallel decoding principle, block turbo decoders using BCH component codes have been implemented. An architecture of BCH(32, 26)\( 2^p \) product codes with single correction power was synthesised. The decoding algorithm is chosen with \( q = 4 \) quantisation levels, eight test vectors, one competitor and \( it = 4 \) iterations. The elementary decoding of a codeword is split into three pipelined phases. Each phase requires \( 32/m \) clock periods and the elementary decoder latency is equal to \( 64/m \) clock periods. Synthesises were
performed using the Synopsys tool with an STMicroelectronics 0.09-vm CMOS process target library. Two architecture types were chosen: e\textsubscript{dec} as the reference and 4-e\textsubscript{dec} where m = 4 symbols are simultaneously processed by an elementary decoder. Elementary decoders have a clock period equal to 2 ns, which corresponds to a frequency of 500 MHz. The estimated area complexity in terms of equivalent gates for the two elementary decoders are: 4400 for BCH(32, 26)\textsubscript{e} and 5700 for BCH(32, 26)\textsubscript{4-e}. This complexity includes all the elementary decoder elements (processing and memorisation). The processing unit gate numbers of the block turbo decoders are equivalent between previous and proposed architectures: 1.13 and 1.45 million for e\textsubscript{dec} and 4-e\textsubscript{dec}, respectively. The latency in terms of symbol number is strongly reduced for the proposed architecture. It decreases from 270336 to 512 for e\textsubscript{dec} and from 5120 to 128 for 4-e\textsubscript{dec}. The memory complexity of the previous architecture in terms of equivalent gates is 126400. It corresponds to 10% of BCH(32, 26)\textsubscript{2} block turbo decoder complexity. On the other hand, the equivalent gate complexity of connection networks is only 5600 for the proposed architecture of BCH(32, 26)\textsuperscript{2} block turbo decoder.

The latency in terms of symbol number is strongly reduced for the ultra-high-speed FEC as block turbo codes opens up new opportunities for the next generation of optical communication systems.

© The Institution of Engineering and Technology 2006
11 July 2006

References


Table 1: Features of different architectures for block turbo decoding

<table>
<thead>
<tr>
<th>Feature</th>
<th>Previous architectures</th>
<th>Proposed architectures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput (Gbit/s)</td>
<td>n * (2it * n\textsuperscript{2} + 2it * L)</td>
<td>n * (2it * D\textsubscript{ref})</td>
</tr>
<tr>
<td>Memory size (Kb)</td>
<td>n * m / (m * 2it)</td>
<td>n * m / (m * 2it + D\textsubscript{ref})</td>
</tr>
<tr>
<td>Number of network connections</td>
<td>2it + 4it\textsubscript{2}</td>
<td>2it – 1</td>
</tr>
</tbody>
</table>

Conclusion: A full-parallel turbo decoding architecture for product codes is proposed. This architecture enables the memory blocks between all half-iterations to be removed. Moreover, the latency of the turbo decoder is strongly reduced. The ultra-high-speed FEC architectures obtained meet demands for ever-higher data rates. In particular, our architectural solution can support optical transmission at data rates above 10 Gbit/s. In this context, using more powerful FEC as block turbo codes opens up new opportunities for the next generation of optical communication systems.

E-mail: christophe.jego@enst.bretagne.fr

Electronics Letters 31st August 2006 Vol. 42 No. 18