Noname manuscript No.
(will be inserted by the editor)

A contribution to the reduction of the dynamic power dissipation in the turbo decoder

Haisheng Liu[†‡]
Christophe Jego[§]
Emmanuel Boutillon[‡]
Michel Jezequel[†]
Jean-Philippe Diguet[‡]

the date of receipt and acceptance should be inserted later

Abstract In the field of mobile communication systems, the energy issue of a turbo decoder becomes an equivalent constraint as throughput and performance. This paper presents a contribution to the reduction of the power consumption in the turbo decoder. The main idea is based on re-encoding technique combined with dummy insertion during the iterative decoding process. This technique, name ”Toward Zero Path” (TZP) helps in reducing the state transition activity of the Max-Log-MAP algorithm by trying to maintain the survivor path on the 'zero path' of the trellis. The design of a turbo decoder based on the TZP technique, associated with different power reduction technique (saturation of state metrics, stoping criterium) is described. The resulting turbo decoder was implemented onto a Xilinx VirtexII-Pro FPGA in a digital communication experimental setup. Performance and accurate power dissipation measurements have been done thanks to dynamic partial reconfiguration of the FPGA device. The experimental results have shown the interest of the different contributions for the design of turbo decoders.

Keywords Turbo Codes, Max-Log-MAP algorithm, re-encoding technique, energy efficient architecture, FPGA prototyping.

1 Introduction

The field of channel coding has undergone major advances for the last twenty years. With the invention of turbo codes [1], it is now possible to approach the fundamental limit of channel capacity within a few tenths of a decibel. This family of Forward Error Correction (FEC) codes is particularly attractive for mobile communication systems and have been included as part of the channel coding standards for high data rate as known UMTS and CDMA2000 (3^{th} generation) or 3GPP-LTE (the last step toward

[†] Institut Telecom/Telecom Bretagne, CNRS Lab-STICC, Brest, France.
email{haisheng.liu, michel.jezequel}@telecom-bretagne.eu
[‡] Université de Bretagne Sud, CNRS Lab-STICC, Lorient, France.
email{emmanuel.boutillon, jean-philippe.diguet}@univ-ubs.fr
[§] IPB/Enseirb-Matmeca, CNRS IMS, Bordeaux, France.
email:christophe.jego@ims-bordeaux.fr

the 4^{th} generation) standards. However, the coding gain is obtained with an increase in operational complexity due to the iterative nature of the decoding algorithm. Moreover, in the field of mobile communication, the energy consumption has to be considered as an equivalent constraint as performance or throughput. The adoption of turbo codes has consistently increased the share of channel decoding in the total receiver energy budget from around 30% to almost 50%. This means that the channel decoder is becoming the main energy bottleneck in the mixed-signal receiver [2].

Several techniques have been proposed to decrease the operational complexity and hence to improve the energy efficiency of turbo decoding. The techniques include stopping criterion in the iterative decoding process to avoid unnecessary iterations [3], window approach to reduce the total memory required in the decoder [4], and encoding soft information to reduce the number of bits necessary to represent the full range of extrinsic information [5]. Moreover, a previous investigation [6] concerning the power consumption of turbo decoders reported that the memory power dissipation accounts for more than 50% of the entire power consumption.

This paper describes an efficient way to reduce the dynamic power dissipation. Indeed, dynamic power P_{dyn} and static (leakage) power P_{lkg} are the two sources of power dissipation. They can be defined by equation (1), where V_{DD}, f, α, C_L and I_{lkg} denote the power supply voltage, the clock frequency, the activity factor, the load capacitance and the leakage current, respectively. Traditionally, dynamic power was considered as the dominant source of power dissipation. The proposed technique is based on a dynamic re-encoding of the received messages. The idea is to decrease the state transition activity of the trellis-based algorithms by replacing the classical direct decoding of the random noisy codewords by an equivalent decoding of an almost "all zero" codewords in order to keep the survivor path on the "zero path".

$$P_{dyn} = V_{DD}^2 \times f \times \alpha \times C_L, \qquad P_{lkg} = V_{DD} \times I_{lkg} \tag{1}$$

The remainder of the paper is organized as follows. The principles of a scarce state transition error-trellis decoding are described in Section II. A brief description of turbo codes, the re-encoding technique and dummy insertion in trellis are successively presented. An iterative decoding of turbo codes based on re-encoding with dummy insertion is then detailed in Section III. A comparison in terms of complexity and power dissipation is carried out in Section IV thanks to the successful design of turbo decoder FPGA prototypes. Conclusions are drawn in the last Section.

2 Scarce state transition error-trellis decoding

In this section, we first recall the principle of the turbo encoder and the associated iterative decoding process. Then we described the proposed TZP technique.

2.1 Notations used in the paper

In the rest of the paper, the following notation is used: D, $\mathbf{X}$ and $[A]$ represent respectively a binary variable, a vector of binary variables and a matrix of binary variables. The k^{th} element of a sequence of variables (or vectors) is represented by the index k (D_k, $\mathbf{X}_k$ for examples). If specified, $A_{k,1}$ and $A_{k,2}$ represent the k^{th} element in the natural and the interleaved order, X^i represents the i^{th} coordinate of vector $\mathbf{X}$ and $\mathbf{X}^T$ the tranposed version of $\mathbf{X}^T$. Finally, $\tilde{D}$ represent a soft value, or Log Likelihood Ratio (LLR) of variable D given an observation: $\tilde{D} = LLR(D/observation) = log(P(D = 1/observation)/P(D = 0/observation))$. For simplicity, it can be assume that $\tilde{D}$ represent a couple $\tilde{D} = (s(\tilde{D}), |LLR(\tilde{D})|)$, where $|LLR(\tilde{D})|$ is the absolute value of $LLR(\tilde{D})$ and $s(\tilde{D})$ is the hard-decided value of $\tilde{D}$, where $s(\tilde{D}) = 1$ if $LLR(\tilde{D}) \geq 0$, $s(\tilde{D}) = 0$ otherwise. The operation: $\tilde{C} = \tilde{A} + B$ implies a binary XOR between $s(\tilde{A})$ and B, i.e. $s(\tilde{C}) = s(\tilde{A}) + B$, the absolute value of LLR remains unchanged, i.e. $|LLR(\tilde{C})| = |LLR(\tilde{A})|$. The operation: $\tilde{C} = \tilde{A} + \tilde{B}$ means that $LLR(\tilde{C}) = LLR(\tilde{A}) + LLR(\tilde{B})$.

2.2 Turbo codes

Convolutional codes are linear codes over the field of one-sided infinite sequences. This code family can be represented by means of a trellis diagram. A Recursive Systematic Convolutional (RSC) code is one in which the information bits are included in the codeword and the redundancy is obtained by feeding back its encoded output to its input. The principle of trellis termination is generally adopted to avoid the degradation of spectral efficiency of the transmission introduced by edge effects. This involves forcing the initial and final states to values known by the decoder. In order to find these states, a well-known technique called $tail-biting$ is usually applied. It ensures that the initial and the final states of the encoder are identical by making the decoding trellis circular. The application of the $tail-biting$ technique to RCS codes requires to determine the circulation state. These codes are then called Circular RCS (CRSC) codes. Fig. 1 shows the structure of a turbo encoder made up of two CRSC encoders concatenated in parallel thanks to an interleaver. Each CRSC code has a coding rate $r_{CRSC} = 1/2$, a codeword length $n = 2k$ and a constraint length $\nu = 4$. The overall code rate of the turbo code is $r = 1/3$. At each time k, the information bit (or systematic bit) D_k and two redundancies (or parity bits) $R_{k,1}$ and $R_{k,2}$ corresponding to the contributions of each CRSC code are provided by the turbo encoder. The first encode received bit $D_{k,1} = D_k$ in the natural order while the second encoder received an interleave version of $D_{k,2} = D_{\Pi(k)}$ thanks to an interleaver Π.

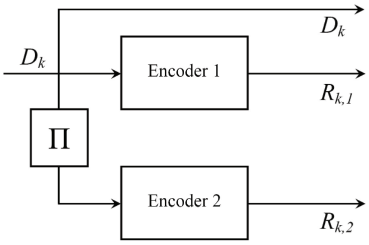

Fig. 1 turbo encoder made up of two CRSC encoders

A turbo decoder is composed of two Soft-Input Soft-Output (SISO) decoders that exchange some information thanks to the interleaver (Π) and the deinterleaver (Π^{-1}) as shown in Fig. 2. Each SISO decoder considers three different inputs: the first two inputs are coming from the channel observation (intrisinc information): the Log Likelihood Ratio (LLR) of the noisy received systematic bit $\tilde{D}_{k,1}$, and the LLR of the noisy received parity bits produced by the correspoding component decoder ($\tilde{R}_{k,1}$ or $\tilde{R}_{k,2}$). The third information is the LLR ($\tilde{Z}_{k,1}$ or $\tilde{Z}_{k,2}$) computed by the other component decoder, known as *a priori* information of the systematic bit. During each iteration, a component decoder computes from its inputs the associated probabilities of each information bit called as *extrinsic information.* They are used as the *a priori* information of the systematic bits by the other SISO component decoder. This information enables to increase the probabilities associated to the information bits and also to improve the error correction efficiency.

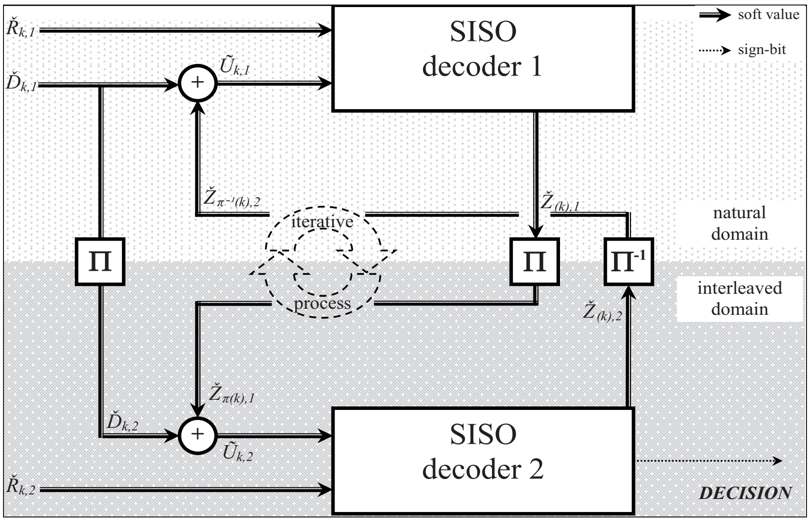

Fig. 2 Block diagram of a turbo decoder

In 1974, a new trellis decoding algorithm capables of associated a probability with the binary decision was presented [7]. It is now referred to *Maximum A Posteriori* (MAP) algorithm in the literature. Rather complex to implement in its initial version, a simplified version called Max-Log-MAP [8] with an acceptable loss of performance is widely used in the trellis decoding of convolutional codes and more particularly turbocodes. This algorithm is performed on a block of n received symbols which corresponds to a trellis with a finite number of stages equal to n. Interested reader are refered to [9] for the description of the MAP algorithm.

2.3 Re-encoding principle

The idea of re-encoding a received codeword was first introduced in the Welch-Berlekamp (WB) algorithm [10] for the decoding of algebraic block codes such as Reed-Solomon (RS) codes. The WB algorithm concentrates on finding a codeword by translating an error pattern into another more likely error pattern in the same coset. A first error pattern is obtained by re-encoding. It corresponds to the error pattern which assumes that all message symbols are correct and that all errors occurred in the check symbols. The same year, a scarce state transition (SST) approach was introduced in [11] to reduce the power consumption of the Viterbi decoding systems for convolutional codes. The received data is precoded before decoding by a conventional Viterbi decoder. The similar re-encoding technique was further extend to the Soft Output Viterbi Algorithm (SOVA) for the turbo code decoding [12] and to the soft-decision Koetter-Vardy algorithm to transform the complex decoding problem of RS codes into an easier one [13]. To our best knowledge, no previous study was done to apply this technique for decoding turbo codes based on the Max-Log-MAP algorithm.

This technique can be explained by using the state space representation of the encoder [14]:

$$\mathbf{X}_{k+1} = [A] \cdot \mathbf{X}_k + [B] \cdot D_k \tag{2a}$$

$$\mathbf{V}_k = [C] \cdot \mathbf{X}_k + [D] \cdot D_k \tag{2b}$$

The mapping between the information sequence D_k and the code sequence $\mathbf{V}_k$ at time k depends on the state of the encoder $\mathbf{X}_k$. Let us consider a CRSC code defined by the generator polynomial G=(1,5/7) octal as shown in Fig. 3.

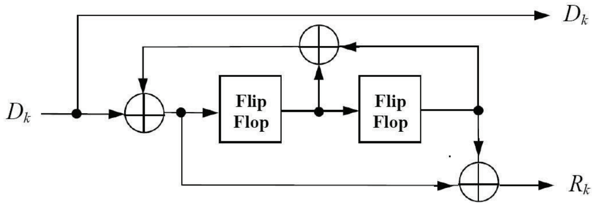

Fig. 3 CRSC code defined by the generator polynomial G=(1,5/7) octal

The corresponding state matrices can be expressed as:

$$[A] = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}; \qquad [B] = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3a}$$

$$[C] = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}; \qquad [D] = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{3b}$$

The code sequence is:

$$\mathbf{V}_k = \begin{bmatrix} D_k \\ R_k \end{bmatrix} \tag{4}$$

where D_k represent the systematic part of the codeword and R_k the redundancy. The state space representation of the encoder becomes:

$$\begin{bmatrix} X^1_{k+1} \\ X^2_{k+1} \end{bmatrix} = \begin{bmatrix} 1\,1 \\ 1\,0 \end{bmatrix} \cdot \begin{bmatrix} X^1_k \\ X^2_k \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} D_k \tag{5a}$$

$$\begin{bmatrix} D_k \\ R_k \end{bmatrix} = \begin{bmatrix} 0\,0 \\ 1\,0 \end{bmatrix} \cdot \begin{bmatrix} X^1_k \\ X^2_k \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} D_k \tag{5b}$$

If we consider that the initial state of the encoder is equal to zero ($\mathbf{X}_0 = [0,0]'$) then the current state $\mathbf{X}_k$ can be expressed as:

$$\mathbf{X}_k = \sum_{i=1}^{k} [A]^{i-1} \times [B]\, D_{k-i} \tag{6}$$

At the receiver side, the demodulator computes the soft values (or LLR values) $(\tilde{\mathbf{V}}_k = [\tilde{D}_k, \tilde{R}_k]')_{k=1..n}$ from the noisy received symbol. The objective of the decoding process is then, form $(\tilde{\mathbf{V}}_k)_{k=1..n}$ to evaluate the error pattern $(E_k)_{k=1..n}$ introduced by the channel on the sequence D_k, the final hard decoded codeword is thus $(\bar{D}_k)_{k=1..n} = (s(\tilde{D}_k) + E_k)_{k=1..n}$.

A re-encoding step can be applied to the hard decided systematic part $(s(\tilde{D}_k))_{k=1..n}$ to generate a new codeword $(\tilde{\mathbf{V}}'_k)_{k=1..n}$. Now, using the linearity of the code, decoding $\tilde{\mathbf{V}}''_k = (\tilde{V}_k + V'_k)_{k=1..n}$ gives the same error pattern $(E_k)_{k=1..n}$ than the direct decoding of $(\tilde{\mathbf{V}}_k)_{k=1..n}$. By construction, the systematic part of $s(\tilde{V}''_k)$ is $s(\tilde{D}_k) + s(\tilde{D}_k) = 0$. In others word, the systematic part of $\tilde{V''}_k$ is composed of the all zero sequence associated with the absolute value of the LLR of $\tilde{D}_k$. In terms of decoding, the all zero input symbols leads, in first glance, the decoder staying in the all-zero path of the trellis.

Note that if the decoder provided a soft value of extrinsic value, the same re-encoding process can be performed with the same effect on the path. In the rest of the section, we analyze more deeply the effect of this re-encoding process considering only the sign of the soft information. Considering that the initial state of the re-encoder is equal to zero ($\mathbf{X}'_0 = 0$) then the current state $\mathbf{X}'_k$ can be expressed, using (6):

$$\begin{aligned} \mathbf{X'}_k &= \sum_{i=1}^{k} [A]^{i-1} \times [B]\, s(\tilde{D}_{k-i}) \\ &= \sum_{i=1}^{k} [A]^{i-1} \times [B]\, D_{k-i} + \sum_{i=1}^{k} [A]^{i-1} \times [B]\, E_{k-i} \end{aligned} \tag{7}$$

Combining equation (6) and (7), we obtained the state $\mathbf{X"}_k$ of a new codeword:

$$\mathbf{X}''_k = \sum_{i=1}^{k} [A]^{i-1} \times [B]\, E_{k-i} \tag{8}$$

This new codeword, by construction, implies only the sign error on the information bits. At time k, the observation $\mathbf{V}''_k$ is then equal to:

$$\mathbf{V}''_k = \begin{bmatrix} 0 \\ X''^2_k \end{bmatrix} \tag{9}$$

and the soft vector $\tilde{\mathbf{V}}''_k$ used by the decoder will be equal to $\tilde{D}''_k = (0, |LLR(\tilde{D}_k))$ for the systematic part, $\tilde{R}''_k = (X''^2_k, |LLR(\tilde{R}_k))$ for the redundant part.

It means that the sequence of the systematic part of $\mathbf{V"}_k$ is equal to zero. The value of the redundancy part of $\mathbf{V"}_k$ at time k is equal to the sum of the noise with the accumulation of the noise for the systematic part from the initial state of the system. Note that this re-encoding technique can also be applied if the code is not systematic. However, a previous decoding step has to be applied to regenerate the information sequence as suggested in [11].

2.4 Dummy insertion in trellis

In order to verify the efficiency of the re-encoding technique for turbo decoding, a study of the survivor paths has been done. Unfortunately, this study shows that the survivor paths do not always remain on the zero-path during a soft decoding process. The recursive propertie of the RCS encoders is the reason of this observation. Simulation results showed that the survivor path remains on the zero-path for 19 % and 68 % of the states at BERs of 10^{-1} and 10^{-6} if the re-encoding technique is applied. Actually, some survivor paths run along a particular periodic path.

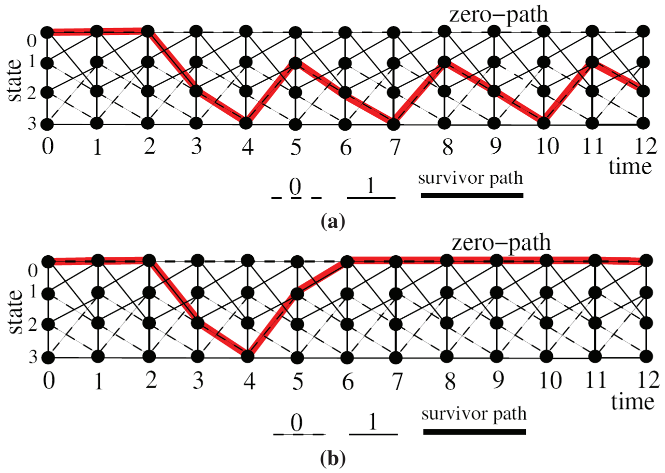

Fig. 4 trellis of convolutional code defined by the generator polynomial G=(1,5/7) octal based on re-encoding technique (a) and based on re-encoding combined with a dummy insertion (b)

Fig.4-a illustrates this observation by an example of survivor path when the noise error sequences are $E_{(k=1..12)} = 001000000000$ and $s(\tilde{R}_{(k=1..12)}) = 000000000000$ for a RCS code defined by the generator polynomial G=(1,5/7) octal. This survivor path runs along the states 1, 2 and 3 in a periodic way. The produced parity bit sequence $R_k{}' = 001110110110$ is composed of a binary pattern 110 with a periodicity equal to three. This binary pattern can be obtained from the code features. Moreover, its periodicity just depends on the constraint length of the code. So, it is possible to find this binary pattern after a re-encoding step and especially to bring the survivor path back to the zero-path. This technique is called *dummy insertion* in this paper. It consists in adding some bits to the sign-bits of the transmitted soft information sequence $\tilde{D}$ for removing the cyclic binary patterns in the parity bit sequence. These dummy bits are then removed after the decoding process thanks to an Xor operation. Fig.4-b shows the new surviving path after the insertion of a dummy bit in the 6^{th} position. As expected, the new sequence $R_k{}'' = 001111000000$ is brought back to the zero-path. If the re-encoding technique is combined with the dummy insertion approach then the survivor path remains on the zero-path for 40 % and 89 % of the states at BERs of 10^{-1} and 10^{-6} , respectively. By reducing the state transition activities, the scarce state transition turbo decoding helps to design some architectures more efficient in terms of energy consumption. The scarce state transition turbo decoding based on re-encoding technique combined with a dummy insertion approach has to be extended to the circular codes. The challenge is about the determination of the circular state. A simple solution is to set the state zero for the initial state. So, the aim of termination is to lead the encoder towards state zero by following one of the paths in the trellis. This technique of treillis termination requires the insertion of zero bits at the end of the frame. The number of zero bits that has to be added is equal to the constraint length of the code.

3 Iterative decoding of turbo codes based on re-encoding with dummy insertion

In this section, we apply the propose TZP algorithm in the context of a turbo decoder. The main difference is that, each decoder should also take into account the extrinsic information generated during the previous half iteration. The analysis of the result show that TZP should be completed with a mechanism of insertion of dummy bit in order to force the decoder toward the zero path.

3.1 Turbo-decoder with re-encoding combined with dummy insertion

The proposed turbo decoding scheme is composed of two systematic convolutional encoders and two Max-Log-MAP SISO decoders as shown in Fig.5. First, the extrinsic information $\tilde{Z}_{\Pi^{-1}(k),2}$ is added to the soft information sequence $\tilde{D}_{k,1}$ in the natural domain: $\tilde{U}_{k,1} = \tilde{D}_{k,1} + \tilde{Z}_{\Pi^{-1}(k),2}$. In parallel, the interleaved extrinsic information $\tilde{Z}_{\Pi(k),1}$ is added to the soft interleaved information sequence $\tilde{Z}_{k,2}$ in the interleaved domain: $\tilde{U}_{k,2} = \tilde{D}_{k,2} + \tilde{Z}_{\Pi(k),1}$. Then, for each decoder, the sign-bits of the obtained sequences $s(\tilde{U}_k)_{k=1..n}$ are only considered for the re-encoding step. The produced parity bit sequences R'_k are XORed with the sign-bits of the soft decision symbol sequences $\tilde{R}_k$. The bits value of soft information sequences are thus replaced by sequences of zero. Traditional Max-Log-MAP algorithms are used to produced the extrinsic information $\tilde{Z}''_k$. The correct value of $\tilde{Z}_k$ are simply obtained using $\tilde{Z}_k = s(\tilde{U}_k) + \tilde{Z}''_k$. After several half-iterations, the iterative process converges as in traditionally Max-Log-MAP decoder and the hard decisions are taken. The linearity of the code implies that the proposed approach has no impact on the BER performance. The transitions inside the trellis during the forward and the backward recursions of the Max-Log-MAP algorithm can thus considerably be reduced. Indeed, if the survivor path is maintained on the

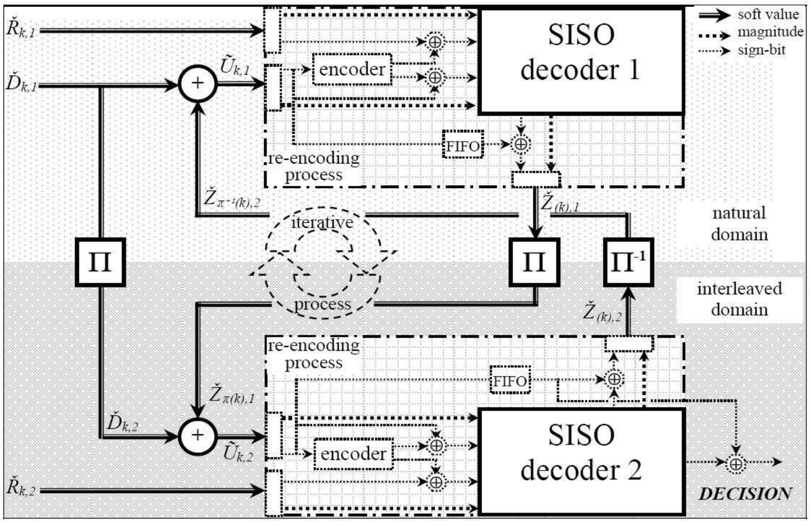

Fig. 5 Block diagram of a turbo decoder with re-encoding technique

zero-path (the path that goes through the states 0) of the trellis then most of the state transitions can be avoided. Note that the metric value of state 0 is turned into a zero by subtracting the corresponding transition metric from the other transition metrics.

3.2 Scarce state transition SISO decoder

The architecture for the two domains (natural and interleaved) of the turbo decoding process is detailed in Fig.6. The main innovation compared to the architecture presented in Fig.5 is about the dummy insertion process.

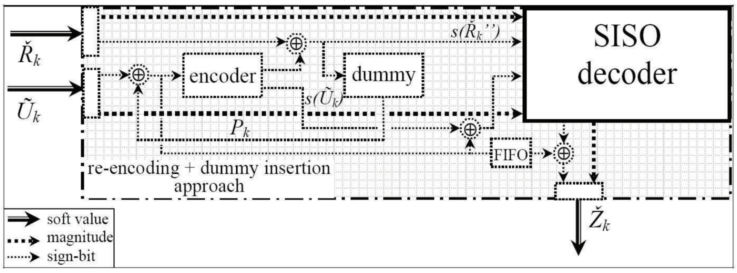

Fig. 6 Block diagram of the scarce state transition SISO decoder

The dummy block given in Fig.7, contains logic elements (Flip-Flop, Xor and And) for detecting the binary pattern but also for constructing the P_k dummy signal. Finally, an Xor logic gate is necessary to add the dummy-bits to the sign-bit of $\tilde{U}_k$. It means that the additional hardware cost of the re-encoding process is very low. However, the

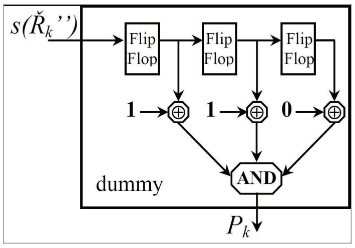

Fig. 7 Block diagram of the dummy block for the CRSC code defined by the generator polynomial G=(1,5/7) octal

dummies inserted during the re-encoding of the soft information sequence in natural domain have to be memorized. This constraint involves an additional cost in terms of memory that can have an impact on the total power dissipation.

4 Design and prototyping of a turbo decoder based on the re-encoding

The design of a turbo decoder based on the TZP algorithm has has been done. The TZP technique has also been combined with a technique of state metric saturation proposed by the authors [15]. Indeed, the precision of state metrics has a significant impact on memory size and thus on total power consumption of a turbo decoder. Using the proposed direct rescaling renormalization of state metrics, the number of bits of the state metrics can been reduced from 7 bits downto 4 bits. Simulations have shown that this optimization results in a loss of 0.1 dB at a BER of 10^{-6}, when the received symbols are 4-bit quantized and the extrinsic information are 6-bit quantized. Compared with the previous state metric quantization, the 4-bit quantization is the better quantization of state metrics so far to the best of our knowledge.

The number of iterations for decoding process depends on the channel characteristic which changes from block to block. Stopping criterion in the iterative process can avoid unnecessary iterations and thus improve the energy efficiency of the turbo decoder. For this reason, a combination of two stop criteria [16] has also been added to the turbo decoder. It gives good BER performance with minimal implementation overhead compared to other stopping criteria. The two criteria operate on the sum and minimum of the absolute reliability values for low and high SNR, respectively. After each iteration, the decoding process is stopped if the condition is true.

4.1 Turbo decoder prototyping

The final objective is to measure the total power dissipation of turbo-decoder architectures to demonstrate the interest of our approach. Usually, an ASIC design is the best way to measure the energy efficiency of an architecture. Indeed, FPGAs have been slower, less energy efficient and generally achieved less functionality than their fixed ASIC counterparts. However, a shorter implementation time, ability to re-program in the field to fix bugs, and lower non-recurring engineering costs relative to an ASIC design are the reasons why we have decided to first implement the different turbo decoder architectures onto an FPGA device. The experimental prototype is composed of a board that includes one Xilinx Virtex2Pro FPGA. All the components of a digital communication setup have been implemented onto the FPGA. A Pseudo Random Generator (PRG) sends a pseudo random data stream at each clock period. Then, a turbo encoder processes the data stream. The designed turbo encoder is composed of two CRSC encoders concatenated in parallel thanks to an interleaver. Each CRSC code has a code rate $R = 1/2$ and a constraint length $\nu = 4$. Moreover, the experimental setup includes a channel emulator and BER measurement facilities in order to verify decoding performance of the turbo decoder by plotting some BER curves. The channel emulator is obtained from an AWGN generator of variables. The AWGN generator is achieved using the design described in [17]. The turbo decoder is composed of two memory blocks (data and extrinsic information) and one SISO decoder. It means that all the iterations in natural and interleaved domains are done by only one decoder.

Different versions of the SISO decoder have been designed for the experimentation. Each version contains the characteristics of the previous one and an innovation. Finally, five versions have been implemented: a classical version, a version with stopping criterion, a version with proposed state metric quantization [15], a version based on re-encoding technique and a final version based on re-encoding combined with dummy insertion. All the decoder versions are clocked at $f_0 = 50MHz$. Logic Syntheses were performed using the Xilinx tool ISE to estimate the complexity of SISO decoders. The classical SISO decoder occupies 633 slice LUTs, 1398 slice Flip-Flops and 4 BlockRAMs of 18Kbs. A comparison in terms of complexity of the different SISO decoders is done in Fig.8.

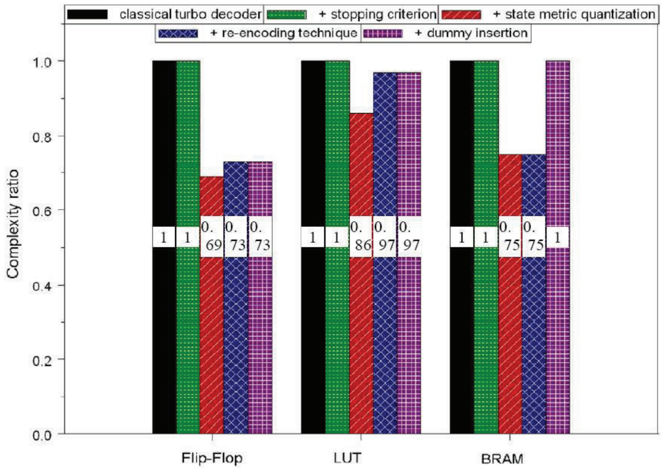

Fig. 8 Comparison in terms of complexity of the different SISO decoders

The classical version is considered as the reference for this comparison. As expected the proposed state metric quantization enables to decrease the numbers of slice LUTs, slice Flip-Flops and BlockRAMs by 31%, 14% and 25%, respectively. On the other hand, the addition of re-encoding technique and in particular a bidirectional CRSC encoder that can process soft information sequence in natural and interleaved domains takes up 22 slice LUTs and 251 slice Flip-Flops. One BlockRAM is also necessary for the memorization of the dummy inserted during the re-encoding of the soft information sequence in natural domain. It means that the final SISO decoder that contains the proposed state metric quantization related to a re-encoding technique with dummy insertion is only less complex in terms of slice LUTs by comparison with a classical SISO decoder. Note that the re-encoding process has a very low impact on the turbo decoder architecture latency. Indeed, the additional latency depends on the convolutional code used to build the turbo code. In our work, a CRSC code (cf. Fig. 3) was considered. It introduces an additional latency of only two period times during the soft decoding of one frame in each domain (natural or interleaved).

4.2 Results in terms of dynamic power consumption

Dynamic Partial Reconfiguration (DPR) is going to make design more flexible by changing a part of the device while the rest of an FPGA is still running. Moreover, the modular design flow allows a designer to work on different modules of a design independently and then to merge them onto one FPGA design. As explained in the previous sub-section, we have designed five versions of the SISO decoder. For this reason, we have experimented the module-based partial reconfiguration in FPGA design. This process permitted to reconfigure only the turbo decoder component of the digital communication setup implemented onto the FPGA. A special bus macros has been added to the digital communication setup to ensure the data exchange between the static part and the dynamic part. The dynamic part contained the turbo decoder. The rest of the digital communication setup has been implemented onto the static part. So, the impact in terms of dynamic power consumption of the different versions of the turbo decoder can be observed while the digital communication setup is still operating. Similarly, the experimental prototype permitted to measure the BER performance of each turbo decoder.

The BER curves obtained by hardware emulation using our experimental setup are equivalent to those obtained by Monte-Carlo simulation on a Gaussian channel. No degradation is introduced by the insertion of the stopping criterion that confirmed the results reported in literature. In [15], simulation results show the effect of the proposed state metric quantization. An equivalent degradation around 0.1 dB has been observed by prototyping. As expected, the performance in terms of BER of a turbo decoder based on re-encoding technique combined with dummy insertion is identical with the traditional turbo decoder.

Turbo codes have been adopted by a wide range of applications. The power consumption cost in a turbo decoder has to be investigated for significant SNR values. In a satellite communication system, SNR values can be around 0 dB if the mobile system is near the limit of the coverage area. But, high SNR values have also to be considered because the same mobile system can be in the middle of the coverage area. For this reason, we have taken the measurements of dynamic power consumption for a relatively large range of SNR values, from 0 dB to 7 dB. The dynamic power consumption of the different versions of the turbo decoder in function of SNR are depicted in Fig.9. Note that each turbo decoder contains the characteristics of the previous one and an innovation. First, the classical turbo decoder has a constant dynamic power consumption equal to 71 mW. The insertion of a stopping criterion enables the major decrease of the dynamic power consumption for all SNR values. The consumption is thus divided by two when the SNR values are higher than 3.5 dB. However, the proposed state metric quantization combined with a stopping criterion increases the dissipation for low SNR values. But, it enables to decrease the dissipation if the SNR value is higher than 3,5 dB. The gain in terms of dynamic power dissipation of the final turbo decoder is very small by comparison of the version that only combined a stopping criterion with the proposed state metric quantization. This result can be explained by the additional complexity that is necessary for the insertion of the re-encoding technique associated with dummy insertion. Besides, it must be noted that current FPGAs are less energy efficient and also have a very important static power dissipation that has an influence on our experimentations. It is most likely that an ASIC design should be more in favour of the re-encoding technique.

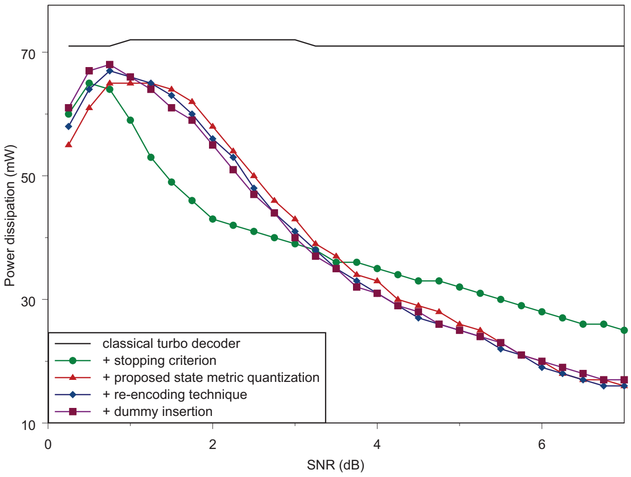

Fig. 9 Dynamic power consumption of turbo decoders in function of SNR

5 Conclusion

The paper described a contribution to the reduction of the dynamic power dissipation in the turbo decoder. The main idea is to decrease the state transition activity of the Max-Log-MAP algorithm by maintaining the survivor path on the 'zero path' thanks to a re-encoding combined with dummy insertion. The paper also discusses FPGA design and prototyping of some turbo decoders based on proposed techniques. Performance and power dissipation measurements have been done thanks to dynamic partial reconfiguration of the FPGA device. The measured gain in terms of dynamic power dissipation obtained by the re-encoding technique combined with dummy insertion is small by comparison of the version that only combined a stopping criterion with the proposed state metric quantization. However, it is important to note that the turbo decoder architectures were designed for turbo codes with a low code rate equal to $\frac{1}{3}$. Experimentations of proposed TZP technique on turbo codes with high code rates would be more profitable in terms of dynamic power consumption. Moreover, a more efficient way to measure the impact of our study on the total power dissipation of a turbo decoder is an ASIC design. However, FPGA design enables to quickly validate by measurements in a real time context algorithm and architecture investigations presented in this paper.

References

1. C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, IEEE International Conference on*, volume 2, pages 1064–1070 vol.2, May 1993.
2. B. Bougard. *Cross-layer energy management in broadband wireless transceivers.* PhD thesis, Katholieke Universiteit Leuven, 2006.
3. J. Hagenauer, E. Offer, and L. Papke. Iterative decoding of binary block and convolutional codes. *Information Theory, IEEE Transactions on*, 42(2):429–445, Mar 1996.
4. S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. A soft-input soft-output app module for iterative decoding of concatenated codes. *Communications Letters, IEEE*, 1(1):22–24, Jan 1997.
5. D. Garrett, Bing Xu, and C. Nicol. Energy efficient turbo decoding for 3g mobile. In *Low Power Electronics and Design, International Symposium on, 2001.*, pages 328–333, 2001.
6. C. Schurgers, F. Catthoor, and M. Engels. Memory optimization of map turbo decoder algorithms. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 9(2):305–312, Apr 2001.
7. L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate (corresp.). *IEEE Trans. on Inf. Theory*, 20(2):284–287, Mar 1974.
8. P. Robertson, E. Villebrun, and P. Hoeher. A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain. *ICC'95, Seattle*, 2:1009–1013, Jun 1995.
9. E. Boutillon, C. Douillard, and G. Montorsi. Iterative decoding of concatenated convolutional codes: Implementation issues. *Proceedings of the IEEE*, 95(6):1201 –1227, june 2007.
10. L. R. Welch and E. R. Berlekamp. Error correction for algebraic block codes. U.S. Patent # 46,33,470, December 1986.
11. S. Kubota, K. Ohtani, and S. Kato. High-speed and high-coding-gain viterbi decoder with low power consumption employing sst (scarce state transition) scheme. *Electronics Letters*, 22(9):491–493, 24 1986.
12. Lang Lin, Chi Ying Tsui, and R.S. Cheng. Low power soft output viterbi decoder scheme for turbo code decoding. In *Circuits and Systems, 1997. ISCAS '97., Proceedings of 1997 IEEE International Symposium on*, volume 2, pages 1369–1372 vol.2, Jun 1997.
13. W.J. Gross, F.R. Kschischang, R. Koetter, and R.G. Gulak. A vlsi architecture for interpolation in soft-decision list decoding of reed-solomon codes. In *Signal Processing Systems, 2002. (SIPS '02). IEEE Workshop on*, pages 39–44, Oct. 2002.
14. C. Weiss, C. Bettstetter, S. Riedel, and Jr Costello, D.J. Turbo decoding with tail-biting trellises. In *Signals, Systems, and Electronics, 1998. ISSSE 98. 1998 URSI International Symposium on*, pages 343–348, Sep-2 Oct 1998.
15. Haisheng Liu, Jean-Philippe Diguet, Christophe Jego, Michel Jezequel, and Emmanuel Boutillon. Energy efficient turbo decoder with reduced state metric quantization. In *Signal Processing Systems, 2007 IEEE Workshop on*, pages 237–242, Oct. 2007.
16. F. Gilbert, F. Kienle, and N. Wehn. Low complexity stopping criteria for umts turbo-decoders. In *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, volume 4, pages 2376–2380 vol.4, April 2003.
17. J.-L. Danger, A. Ghazel, E. Boutillon, and H. Laamari. Efficient fpga implementation of gaussian noise generator for communication channel emulation. In *Electronics, Circuits and Systems, 2000. ICECS 2000. The 7th IEEE International Conference on*, volume 1, pages 366–369 vol.1, 2000.