International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 4, Number 1 (2011), pp.41-48

http://www.irphouse.com

Low Power Implementation of Turbo Code with Variable Iteration

Niladri P. Maity and Reshmi Maity

Department of Electronics & Communication Engineering,
Mizoram University (A Central University), Aizawl-796004, Mizoram, India.
E-mail: maity_niladri@rediffmail.com

Abstract

The transport of information from the source to its destination has to be done is such a way that the quality of the received information's should be as close as possible to the quality of the information's. Forward Error Correction (FEC) technique is one of the finest solutions for achieving higher quality information's. A innovative coding technique has developed called "Turbo Coding" which stems from convolutional coding, has been adopted in the third generation (3G) Mobile communications system to the fourth generation (4G) due to its high coding gain and reasonable computation complexity. This paper presents the implementation of the iterative turbo decoder based on the Log-Maximum-a-Posteriori (Log-MAP) algorithm. The design of encoder using Recursive Systematic Code (RSC) with puncturing techniques is presented. Component decoders are implemented by Log-MAP algorithm and thereafter implementation of overall turbo decoder is illustrated in detail. Finally we have investigated low power design technique of the turbo decoder design with variable iteration techniques.

Introduction

Turbo code was proposed by Berrou, Glavieux and Thitimajashima [1] in 1993, represents the most important breakthrough in coding, since Ungerboeck introduced trellis codes in 1982. They have found very wide range of applications mainly in wireless communication, ranging from the 3G mobile system to deep space exploration. It is a very powerful error correcting coding technique, which enables reliable communication with Bit Error Rate (BER) close to Shannon limit; this improvement is due to the SISO (Soft-In-Soft-Out) decoding algorithm to produce soft decisions. A SISO decoder can be implemented using the MAP or Maximum-Likelihood Algorithm (MLA). The MAP algorithm is originally described in 1974

[14], was generally overlooked in favor of the less complex Viterbi algorithm.. Recently with the booming of portable devices such as cellular phones, camcorders and laptop computers, power consumption has become an important factor in VLSI design [2]-[7] for the Turbo code system and some modern designer also proposing new structures [8]-[10] of the system which gives us a better turbo code with high speed [11][12]. In this paper, a low power design of a turbo decoder with punctured technique [13] at the gate level in standard cell design environment is proposed. It starts from a behavioral VHDL description of the circuit, which is synthesized by generating a gate level design using Leonardo's Spectrum of Mentor graphics tools. The gate level can be imparted into place and route tool to generate layout design. We have introduced low power technique into a behavioral description of a turbo decoder by taking flexible number of iterations. When channel condition is good, a smaller number of iterations may be sufficient to save power. Experimental results indicate that a variable number of iterations and clock gating reduces power dissipation significantly.

MAP Algorithm

L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv [14] proposed an algorithm, known as BCJR algorithm, named after its inventors or MAP algorithm. They showed how the algorithm examines every possible path through the convolutional decoder trellis diagram and therefore initially seemed to be unfeasible complex for application in more systems architecture. Hence it was not widely used before discover of the turbo code. However, the MAP algorithm provides not only the estimated bit sequence, but also the probabilities for each bit, that it has been decoded correctly. This is essential for the iterative decoding of turbo codes proposed by Berrou [15]. The encoder structure of the turbo coding for k = 3 RSC code, k is the constraint length is shown in Fig.1.

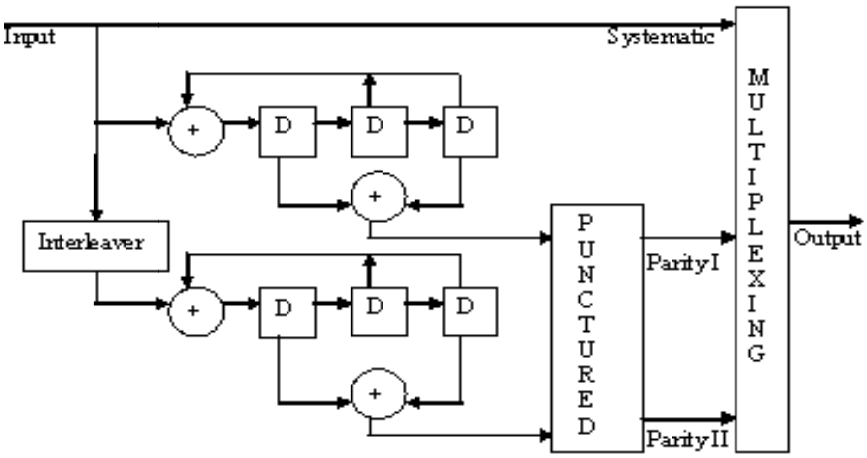

Figure 1: The overall structure of Turbo encoder.

If previous state S_{k-1} and present state S_k are known, the value of z_k which cause the transition between these two states will be known. The received sequence can be split up by three sections: received code word associated with the present transition y_{rk}, the received sequence prior to the present transition $y_{rj<k}$ and the received sequence after the present transition $y_{rj>k}$. Then the LLR becomes,

$$LLR\ (Z_k / Y_r) = \ln \frac{\sum_{Z_k=+1} \alpha_{k-1}(s')\theta_k(s',s)\beta_k(s)}{\sum_{Z_k=-1} \alpha_{k-1}(s')\theta_k(s',s)\beta_k(s)}$$

Where,

$$\alpha_{k-1}(s') = P(s_{k-1} = s', y_{rj<k}),$$

The probability that the trellis is in state s' at time k-1 the received channel sequence is $y_{rj<k}$,

$$\beta_k(s) = P(y_{rj>k} / s_k = s),$$

The probability that is in state at time k, the future received channel sequence is $y_{rj>k}$,

$$\theta_k(s',s) = P\{(y_{rk}, s_k = s) / s_{k-1} = s'\},$$

The probability that the trellis was in state s' at time k-1, it moves to state s, the received channel sequence for this transition is y_{rk}. Fig. 2 gives the idea about the recursive calculation $\alpha_{k\backslash}(s)$, $\theta_k(s)$, $\gamma_k(s', s)$.

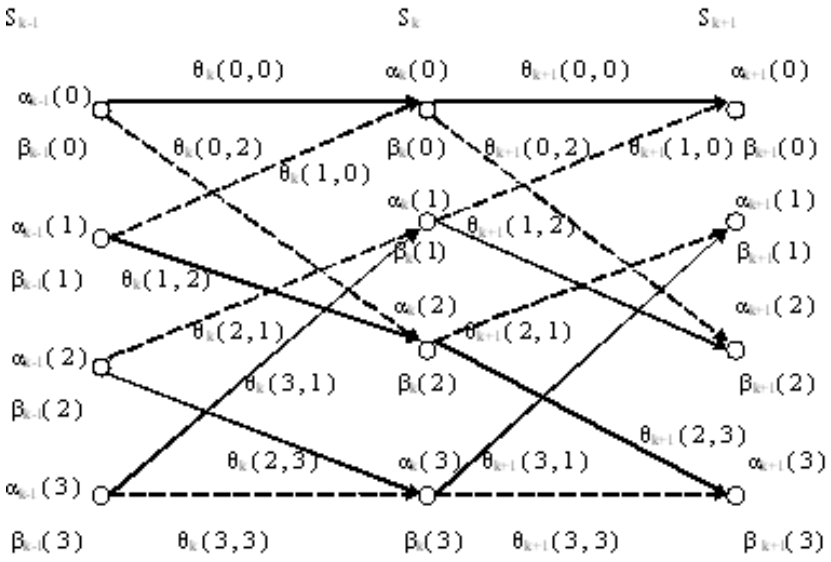

Figure 2: Calculation of $\alpha_k(s)$, $\beta_k(s)$, $\theta_k(s', s)$.

Iterative Design Technique

Use of the system in practice has shown if we subtract the LLR of the a-priori information after each constituent decoder and make a number of decoding iterations. The general turbo decoder structure is shown in Fig. 3.

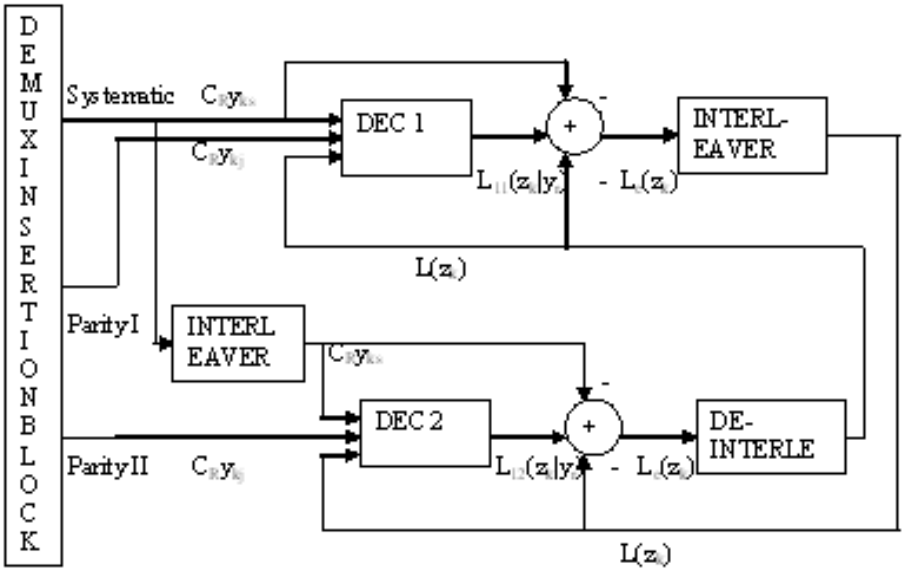

Figure 3: Iterative decoding technique.

At the first iteration, DEC1 receives the channel sequence $C_R Y_1$ (C_R is the Channel reliability) containing received version of systematic bits $C_R\, y_{ks}$ and the parity bits $C_R\, y_{kj}$ (Parity I). To obtain a half rate code, half of these parity bits will have been punctured at the transmitter, so the turbo decoder must inserts zeros in the soft channel output $C_R\, y_{kj}$ for these punctured bits. Then by calculation of θ_k (s', s), α_k (s) & β_k (s) produce its estimate $L_{11}(z_k|y)$ (first iteration of first decoder) of the conditional LLR's of the data bits z_k, k =1,2,3,----N. In this first iteration, DEC1 will have no a-priori information about the bits so, $L(z_k) = 0$, gives the a-priori probability [$P(z_k)$] of 0.5. Now DEC2 comes into operation. It receives channel sequence $C_R Y_2$ containing received version of systematic bits, $C_R y_{ks}$ by Interleaved version and parity II, $C_R y_{kj}$ by punctured version and the interleaved version of extrinsic value of DEC1 $L_e(z_k)$, this is the $L(z_k)$ for the DEC2. $L_e(z_k)$ is completely independent from all other information. By calculation of $\theta_k(s',s)$, $\alpha_k(s')$ and $\beta_k(s)$, produce it's a-posteriori value $L_{12}(z_k \mid y_r)$ and this is the end of the first iteration. Next, the final output will get by the de-interleaved version of $L_{12}(z_k \mid y_r)$ and taken hard decision. For second iteration, DEC1 again processes its received channel sequence $C_R Y_1$', now it also has a-priori LLR's $L(z_k)$ provided by the extrinsic value of DEC2 of first iteration. Then it can produce an improved a-posteriori $L_{21}(z_k|y_r)$. Finally we calculate $L_{22}(z_k|y_r)$. This iterative process continues for each iterations on average the BER will fall.

Results and Discussion

The turbo decoder was described in VHDL at behavioral level by Modelsim tool and logic synthesize tool, called Leonardo Spectrum of Mentor Graphics was used to obtain a gate level circuit from the behavioral description. CMOS circuit simulation has been done by HSPICE using 0.35µm technology with supply voltage of 3.3V. We consider the ½ rate encoder (after puncturing) with interleaver length is 1000 bits and constraint length is 3. The turbo decoder and system specifications are given in Table 1.

Table 1: Decoder and system specifications.

S. N.	Components	Specifications
1.	Decoder	Type: Log-MAP decode Interleaver length: 999 bits De-interleaver length: 999 bits System clocks: 40 MHz Date clock: 2.5 MHz
2.	System	System Clock: 40 MHz Gate count: 12,759 NAND2 gates Bit rate: 125 Kbps Frame Length: 1000 Symbols/frame

With the modulation technique of BPSK modulation and AWGN channel the Simulation results gives the number of iterations to achieve the expected BER is summarized in Table 2.

Table 2: No. of Iteration with BER.

No of Iteration	E_b/N_0
1	> 2.5 dB
2	1.2 dB, 2.5 dB
3	0.9 dB, 1.2 dB
4	0.7 dB, 0.9 dB
5	0.5 dB, 0.7 dB

The performance of our decoder can be further improved by employing a large number of bits for Internal [4] data and large interleaver length. But this decoder requires more complex hardware.

A low power design of a turbo decoder at the gate level in standard cell design environment is proposed. The standard cell design procedure starts from behavioral VHDL descriptions of the circuit. The standard cell based designs are more advantageous then full custom design for faster turn around time, accurate modeling and ease in verification of the circuit. Due to switching transient current, charging and

discharging of load capacitance dynamic power dissipation is main source for full static CMOS circuits. The supply voltage and clock frequency are determined at the system level and they are beyond of control of a circuit designer. It is possible to reduce C_{Load} by using less hardware and reducing wire capacitance. Here we measured power dissipation of three different mode of operation for the taken decoder during a typical frame, which is shown in Fig. 4.

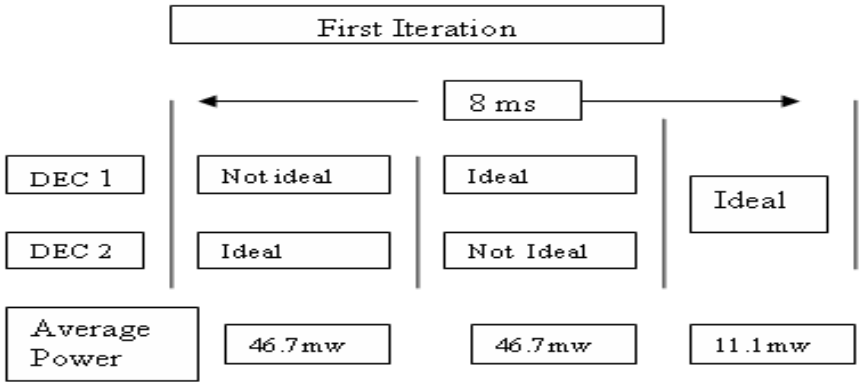

Figure 4: Power dissipation.

Each component decoder's power dissipation is about 36.2 mw and total amount of energy consumed for each iterations is 76.4946×10^{-6} J. Fig. 5 gives the idea of energy consumed for various numbers of iterations for one time.

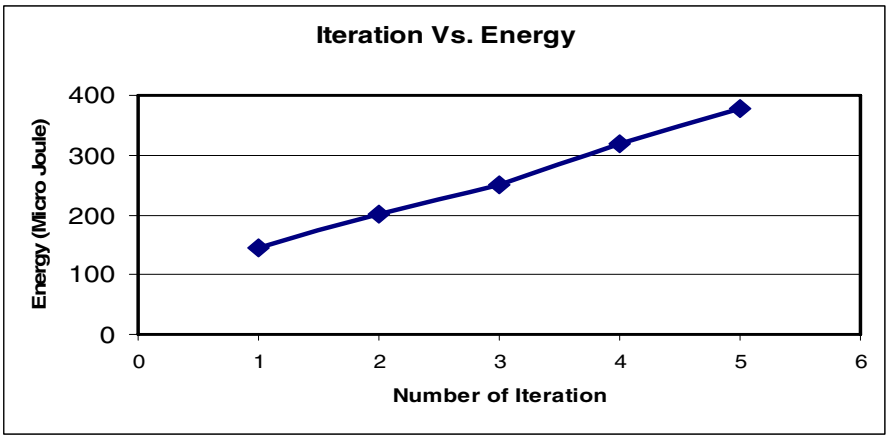

Figure 5: Number of Iteration Vs. Energy consumed.

Our aim in the low-power design of turbo decoders is to reduce the dynamic power dissipation in the standard cell design environment. We considered two methods, variable number of iterations and clock gating techniques. The number of

iterations is decided dynamically according to the channel condition. The clock-gating method is applied to the two component decoders. When one of the decoders is working, the other is idle (see Fig. 4). The clock of the idle decoder is gated to save power. We also block inputs of the idle decoder to prevent propagation of spurious inputs. We have got the BER performance of our turbo decoder and see after five iterations, the BER of our turbo decoder reaches 7.8×10^{-5}, which is near the Shannon Theory. Although direct comparison with other turbo decoders is difficult, we believe that the performance of our turbo decoder is reasonably good. We have also calculated the critical delay in our turbo decoder which is 17.34 ns, and the total number of NAND2 gates is 12579. The system clock of the circuit is set to 40 MHz and the maximum clock frequency is 57 MHz. The dynamic power dissipation of the circuit was measured based on the annotated switching activities from the gate level simulation. The variable number of iterations and clock gating of idle component decoders were proposed to reduce power dissipation. The results show that the proposed methods save the power significantly.

Conclusion

An ultimate error correcting coding, which is known as turbo codes, generates tremendous interest in channel coding of digital communication systems in modern technology in last decade due to its high error correcting capability and better performance comparison with other error correcting coding system. Parallel concatenated encoding and iterative decoding with interleaving are two key design innovations of the turbo codes. We have investigated a low power implementation of a turbo decoder in this contribution. The log - MAP algorithm is adopted to implement component decoders for the design, which is basically equivalent to the original MAP algorithm that is optimal for estimating the information bits. The complexity of the hardware implementation of the log-MAP algorithm is significantly reduced through computing in the logarithm domain.

References

[1] C. Berrou, C., Glavieux, A., and Thitimasjshima, P., 1993, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," Proc. IEEE International Conference on Communication, Switzerland, pp. 1064-1070.

[2] Yang, W., Tsui, Chi-Ying., and Cheng, R. S., 2000, "A Low Power VLSI Architecture of SOVA-based Turbo-code decoder using Scarce State Transition Scheme," Proc. IEEE International Symposium on Circuits and Systems, Geneva, Switzerland, pp. 283-286.

[3] Hong, S., Yi, J., and Stark, W. E., 1998, "VLSI design and implementation of low-complexity adaptive turbo-code encoder and decoder for wireless mobile communication applications," IEEE Workshop on Signal Processing Systems, Cambridge, pp.233-242.

[4] Hong, S., and Stark, W. E., 1998, "VLSI circuit complexity and decoding performance analysis for low-power RSC turbo-code and iterative block decoders design," Proc., IEEE Military Communication Conference, Boston, 3, pp.708-712.

[5] Masera, G., Piccinini, G., Roch M. R., and Zamboni, M., 1999, "VLSI Architectures for Turbo Codes," IEEE Transactions on VLSI systems, 7(3), pp. 369-379.

[6] Wang, Z., and Li, Q., 2007, "Very Low-Complexity Hardware Interleaver for Turbo Decoding", IEEE Transaction on circuits and systems-II: Express Briefs, 54(7), pp. 636-640.

[7] Nabipoor, M., Khodaian, S. A., Sedaghati-Mokhtari, N., Fakhraie, S. M., and Jamali, S. H., 2006, "A High-Speed Low-Complexity VLSI SISO Architecture," IEEE APCCAS, Singapore, pp. 1512-1515,

[8] Del Ser, J., Crespo, P. M., Esnaola, I., and Garcia-Frias, J., 2010, "Joint Source-Channel Coding of Sources with Memory using Turbo Codes and the Burrows-Wheeler Transform," IEEE Transactions on Communications, 58(7), pp. 1984-1992.

[9] Haghighat, J., Hamouda, W., and Soleymany, M. R., 2006, "Design of Lossless Turbo Encoders" IEEE Signal Processing Letters, 13(8), pp. 453-456.

[10] Ishibashi, K., Ishii, K., and Ochiai, H., 2010, "Dynamic Coded Cooperation Using Multiple Turbo Codes in Wireless Relay Networks, " IEEE Journal of Selected Topics in Signal Processing, PP(99), pp. 1-1.

[11] Boyer, P., and Giordano, A., 2009, "Turbo decoder SNR estimation with RAKE reception," IEEE Transactions on Communications, 57(2), pp. 430-439.

[12] Gazi, O., and Yilmaz, A. O., 2007, "Fast Decodable turbo Code," IEEE Communications Letters, 11(2), pp.173-175.

[13] Chatzigeorgiou, I., and Wassell, I. J., 2008, "Revisiting the Calculation of effective free distance of turbo codes," Electronics Letters, 44(1), pp. 43-44.

[14] Bahl, L. R., Cocke, J., Jelinek, F., and Raviv, J., 1974, "Optimal decoding of linear codes for minimizing symbol error rate," IEEE Transactions on Information Theory, 20, pp. 284–287.

[15] Berrou C., and Glavieux, A., 1996, "Near optimum error correcting coding and decoding: Turbo-codes," IEEE Transactions on Communications, 44(10), pp. 1261-1271.