ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Simplified Turbo Decoder Design for Energy Efficient Wireless Sensor Networks

Monisha.V, S. Angeline Grace Stella

Department of ECE, K.L.N College of Engineering, Pottapalayam, Sivagangai, India.
Department of ECE, K.L.N College of Engineering, Pottapalayam, Sivagangai, India.

ABSTRACT---Turbo codes offer low transmission energy consumption hence they are used in energy constrained wireless communication applications. BCJR algorithms are used in the Turbo Decoder. Energy consumption is an important factor in the wireless sensor networks which has to be minimum. Hence we use LUT log BCJR architecture. SBGT-BCJR algorithm is implemented in this paper . In the SBGT algorithms the complexity of computations increase but the memory requirement reduces. SBGT is modified BCJR algorithm which reduces the energy consumption than other BCJR architectures and also increases the throughput.

KEYWORDS—Energy-efficient, error-correcting code(ECC), SBGT-BCJR algorithm, turbo code.

I. INTRODUCTION

Wireless sensor networks are a collection of sensor nodes that move and interact with physical environment and have the ability of sensing, computing and communication[3]. Sensor networks consist of large number of sensors, equipments, low power transmitters and receivers. Energy is a very important factor in wireless sensor networks. To increase the network lifetime energy must be saved in the hardware as well as the software. It is observed that negligible energy is consumed in the process of data acquisition and data processing. More energy is consumed in the process of communication between two nodes. Sensors energy consumption is dominated by transmission energy. Turbo codes, a class of high performance forward error correction which approaches shannon capacity[2]is used in wireless sensor network applications like 3GPP and LTE. These Turbo codes are used as its coding gain facilitates reliable communication using reduced transmission energy. Turbo code is formed by the parallel concatenation of 2 Recursive Systematic Codes (RSC) [7] separated by an interleaver.The interleaver scrambles the input bits in a random predetermined fashion. At the decoder side BCJR Bahl, Cocke, Jelinek, Raviv) algorithm was introduced. The BCJR algorithm is also called the MAP(Maximum A Posterior) algorithm. Higher throughputs excess of 100 Mbits/sec is achieved by employing the Max-log-BCJR algorithm.LUT(look up table) log BCJR algorithms is used than Max-Log-BCJR algorithm and therefore does not suffer from the coding gain degradation. Various LUT-log-BCJR architectures are designed for high throughput. A modified version of the BCJR-MAP algorithm is the BGT-MAP (Berrou,Glaviex,Thitmajshima). The BGT-MAP algorithm was further updated to reduce complexity hence PB-MAP (Peitrobon Barbulescu) algorithm was introduced[4]. To further simplify the complexity the BCJR Algorithm was further modified which is called the Simplified BGT-MAP (SBGT) algorithm. It uses recursive procedures in it.

II. RELATED WORK

In [1] energy is saved in wireless sensor networks using LUT log BCJR architecture in turbo decoder. They facilitate a low transmission energy and reduce the overall energy consumption. In [2] an introduction related to Turbo codes was introduced which offers Shannon capacity in channels that is it offers maximum channel capacity in a digital communication system. A detail on wireless sensor networks is given in [3] The different types of wireless sensor networks and their working is explained in this paper . Different algorithms used in the Turbo decoder are presented in [4]. BCJR, SBGT-BCJR, DUAL SBGT, PB MAP algorithms are explained. The Turbo decoding structure is implemented in FPGA in [5]. The BER of a BCJR decoder is compared with Xilinx log MAP turbo decoder. The Turbo decoder structure for low energy mobile wireless communication is shown in [6]. The structure for the calculation of forward metrics , backward metrics and the structure of the LLR calculation unit is implemented here. The Complete operation of turbo encoder and decoder is explained in [7]. An example is given which shows the Trellis structure and shows the calculation of the metrics for all the iterations until the input is decoded.

III. TURBO CODING

Turbo codes capable of achieving close to Shannon capacity and amenable to hardware efficient implementation, are used in many wireless applications and wireless communication standards like LTE, 3GPP, HSDPA. The decoders here use an iterative method to decode until correct input sequence is decoded hence the name Turbo. As it's a FEC method the decoder itself tries to correct the errors[9].

A turbo encoder [7] consists of a parallel concatenation of two convolutional codes and a interleaver. The turbo encoder specified in the LTE standard is illustrated in the figure 1 . It consist of a convolutional encoder and uses a QPP (Quadratic permutation Polynomial) interleaver. The interleaver splits the incoming bits and it permutes low weight code words in one encoder and high weight code words for another encoder. It also shuffles the input data bits in a pseudo random manner. The Turbo encoder code rate is 1/3 where for each input bit the output is given as 3 bits. N input bits will be coded to 3N data bits. The encoders are RecursiveSytematicCode(RSC)

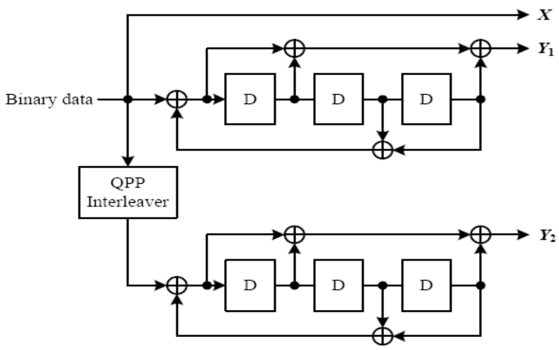


Fig. 1Turbo Encoder Schematic

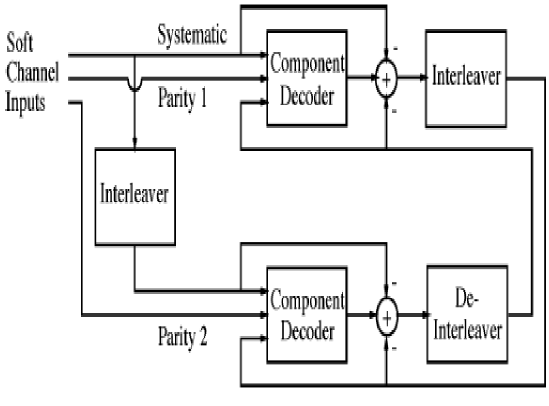


Fig. 2 Turbo Decoder Schematic

The decoder as shown in the figure 2 [9] is composed of a concatenation of interleavers and soft decoders where input values can be other than 0 or 1 indicating reliability and output refer to the fact that each bit in the decoded output takes values indicating reliability .The decoder operates in a iterative way and decoder stops when reliability is reached. Each decoder module produces a better correction term called the extrinsic information which is used as priori information to the other decoder and another input to the decoder is the log likelihood ratios/(LLR) of the received symbols[6]. The LLR is the ratio of the probability of occurrence of symbol 1 to the probability of the occurrence of symbol 0 [8]. The extrinsic information is calculated depending on the algorithm class. They all use a forward recursion and a backward recursion and can be extended toward a sliding window approach. High throughput Turbo code decoder architectures have to split a received data block into so called windows for parallel decoding. The extrinsic information is calculated based on alpha and beta state metrics. In forward recursion the transition metrics and State metrics alpha are found and in the backward recursion transition metrics and state metrics beta are obtained

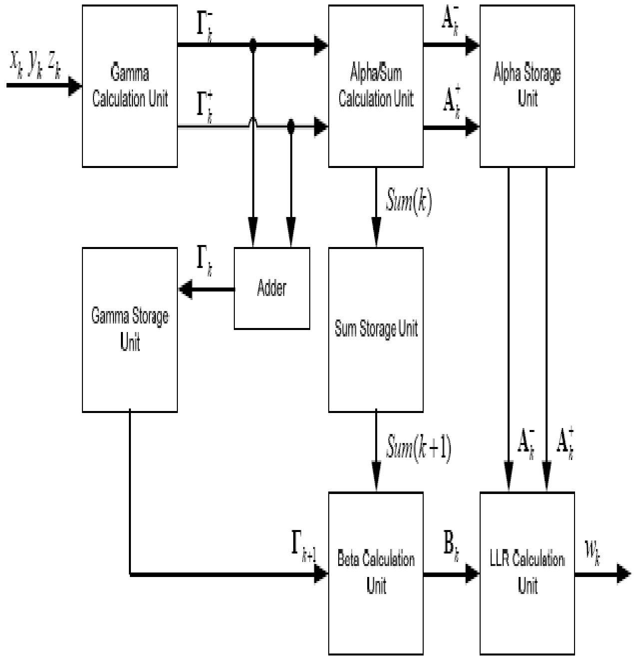


Fig.3 Hardware structure of the Component Decoder

The figure above shows the hardware structure [5] of the Component decoder in a Turbo Decoder. It consists of Gamma calculation unit which is the calculation of branch metrics which are found on the branches in a trellis. Alpha calculation unit is used to find the state metrics in the forward recursion. The structure of the alpha calculation unit is shown in figure 4. and beta calculation unit is used to find the state metrics in the backward recursion mode. The beta calculation unit also called backward calculation unit is shown in figure 5. By storing these values in the storageunit we can obtain the log likelihood ratio(LLR) [6] using the LLR calculation unit. The structure of the LLR metric calculation unit is shown in figure 6.

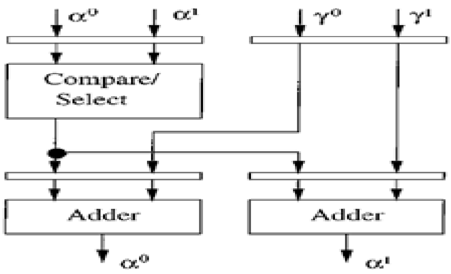


Fig. 4 Forward metric calculation unit

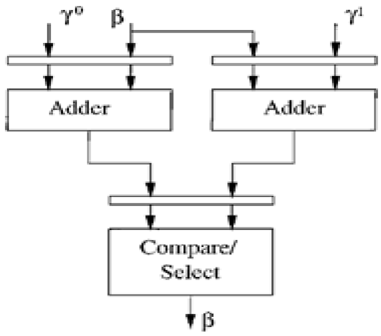


Fig. 5 Backward calculation unit

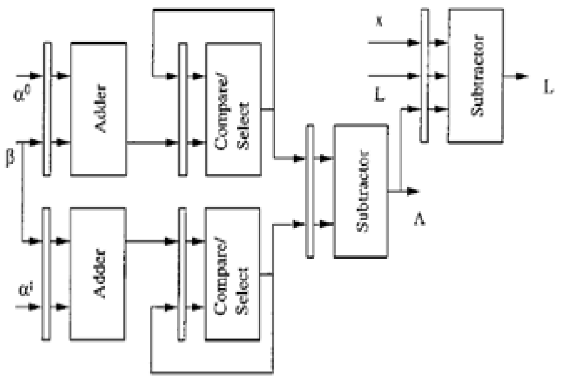
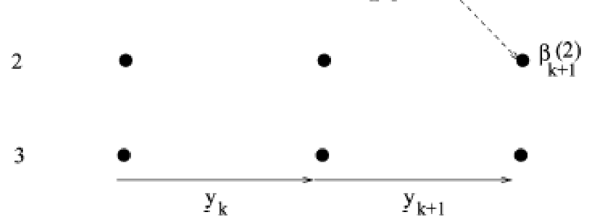


Fig. 6 LLR metric calculation unit

Fig. 7, which depicts a section of a four-state trellis for an RSC code having a constraint-length of K =3, shows this split of the received channel output sequence. In this Figure, solid lines represent transitions as a result of a '1' input bit, and dashed lines represent transition resulting from a '0' input bit. This shows an example of how α values are calculated recursively using α(s') and τ(s',s). Notice that, as we are considering a binary trellis, only two previous states S_{k-1} =0 and S_{k-1}=1, have paths to the state S_k=0. The values on the left hand side are taken as α which are computed using the forward recursion method and the values on the branches are the transition metrics. The transition metrics are the branch metrics denoted as . The values taken from the right hand side are β which are computed using



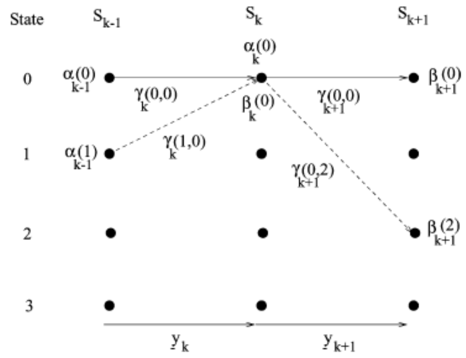


Fig. 7 Recursive calculation of α , β and γ in a trellis

IV. SBGT ALGORITHM

In this section, we derive the SBGT MAP algorithm from the BCJR MAP algorithm [4].

For i = 0, 1 and $1 \leq t \leq \tau$, let the probability sequence

$$\alpha_t^i(m) = \sum_{(m,m')\in B_{t,i}} \alpha_{(t-1)}(m')\gamma_t(m',m)$$

For m = 2^v encoder states

The APP (a posterior probability) sequence is given as

$$\Lambda(d_t) = \log \frac{\sum_{m=0}^{2v-1} \alpha_t^1(m)\beta_t(m)}{\sum_{m=0}^{2v-1} \alpha_t^0(m)\beta_t(m)}$$

Since

$$\frac{\sum_{(m,m')\in B_{t,1}} \alpha_{t-1}(m)\gamma_t(m,m')\beta_t(m)}{\sum_{(m,m')\in B_{t,0}} \alpha_{t-1}(m)\gamma_t(m,m')\beta_t(m)}$$

=

$$\frac{\sum_{m=0}^{2^v-1} \beta_t(m) \sum_{(m',m)\in B_{t,1}} \alpha_{t-1}(m')\tau_t(m',m)}{\sum_{m=0}^{2^v-1} \beta_t(m) \sum_{(m',m)\in B_{t,0}} \alpha_{t-1}(m')\tau_t(m',m)}$$

=

$$\frac{\sum_{m=0}^{2v-1} \alpha_t^1(m)\beta_t(m)}{\sum_{m=0}^{2v-1} \alpha_t^0(m)\beta_t(m)}$$

$\alpha_t^i(m)$ Admits the probabilistic interpretation

$$\alpha_t^i(m) = \sum_{(m',m)\in B_{t,i}} \alpha_{t-1}(m)\gamma_t(m',m)$$

$$\alpha_t^i(m) = \sum_{(m',m)\in B_{t,i}} \Pr\{S_{t-1} = m'; Y_1^{t-1}\} *$$

$$\Pr\{S_t = m; Y_t\, S_{t-1} = m'\}$$

$$= \Pr\{d_t = i; S_t = m; Y_1^t\}$$

$\alpha_t^i(m)$ can be computed using the following Forward recursions

$$\alpha_0^0(0) = \alpha_0^1 = 1$$

$$\alpha_t^i(m) = \sum_{m'=0}^{2^v-1}\sum_{j=0}^{1} \alpha_{t-1}^j(m')\gamma_i(Y_t, m', m)$$

$1 \le t \le \tau \quad I = 0,1 \quad 0 \le m \le 2^v - 1$

$\beta_\tau(m)$ can be computed using the following backward recursions as follows.

$$\beta_\tau(m) = 1 \qquad 0 \le m \le 2^v - 1$$

$$\beta_t(m) = \sum_{m'=0}^{2^v-1}\sum_{j=0}^{1} \beta_{t+1}(m')\gamma_i(Y_{t+1}, m, m')$$

$1 \le t \le \tau\text{-}1, \quad 0 \le m \le 2^v - 1$

Branch metrics $\tau_t(j,m)$ is computed using the equation

$\tau_t(j,m) =$

$$\exp\left(j(La(dt) + Lc\, r_t(1)) + \sum_{p=2}^{n} Lc\, r_t^p Y_{p-1}(j,m)\right)$$

$L_C = \frac{2}{\sigma^2}$ is called the channel reliability coefficient.

La (dt) is the priori information sequence of the input data sequence dt.

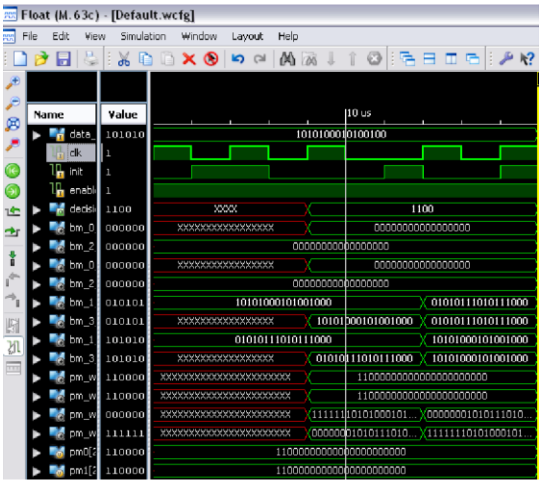
V. RESULT ANALYSIS

In this section we summarize the key characteristics of the SBGT implemented Turbo Decoder and compare it with the other BCJR implemented algorithms that is the LUT-log BCJR and Max-Log BCJR implemented turbo decoders. The implementation results arising from different technologies are scaled to give a fair comparison. As shown in the table there is an reduction in energy consumption when compared to other architectures. Power consumption is also comparatively less . Memory required is reduced by increasing the calculation complexity in the proposed SBGT –BCJR Turbo decoder.Comparison with respect to the gate count , clock frequency supply voltage and throughput is also done and its proved that proposed SBGT algorithm achieves higher performance rates than other Turbo decoder.

TABLE 1

COMPARISION OF THE IMPLEMENTED TURBO DECODER

Publication	proposed	[1]	[5]
Algorithm	SBGT	LUT-Log	Max-Log
Clock Frequency F(MHz)	333	333	390.6
Power (Mw) Consumption	2.7	4.17	788.9
Supply Voltage	1.2	1.0	1.2
Energy consumption (nJ)	0.2	0.4	0.37



Simulation results of Turbo Decoder

The simulation results of the Turbo Decoder in the figure shows the generation of the branch metrics and the path metrics which are the state metrics and or the backward recursion and the data bits are decoded after undergoing a few iterations. The Data bits are first padded on both sides by parity bits and later in the final stages the parity bits are removed and output is in the form of decisions.

VI. CONCLUSIONS

In this paper we have demonstrated the modified BCJR algorithm, SBGT-BCJR algorithm which has simplicity in the calculation and gives higher throughput and energy efficiency than other BCJR algorithms like LUT-log BCJR ,Max-Log-BCJR algorithms which are used in the turbo decoder for decoding purpose .The SBGT algorithm offers a throughput of 1.03 Mb/s which is appropriate for energy constrained applications such as in environmental monitoring Wireless Sensor Networks. The SBGT –BCJR decoder implements an approximately 75% lower energy consumption of 0.2 nj/bit/iteration.

REFERENCES

[1] Liang Li, Robert G. Maunder, Bashir M.Al-Hashimi, Fellow, IEEE, and Lajos Hanzo." A Low-Complexity Turbo Decoder Architecture for Energy Efficient Wireless Sensor Networks," IEEE transactions on VLSI SYSTEMS, VOL.21, NO.1,JANUARY 2013

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding Turbo codes", in Proc. IEEE Int.Conf. Commun., 1993, pp 1064- 1070.

[3] I.F.Akyilidiz, W.Su, Y. Sankarasubramaniam, and Cayiciri, »Wireless Sensor networks :A survey, »Computer networks : The International Journal of Computer Telecommunications Networking., Vol.52,pp. 292-422,2008.

[4] Sichun Wang and Francois Patenaude, "A Systematic Approach to modified BCJRMAP algorithms for convolutional codes," Hindawi Publishing Corporation EURASIP journal on applied signal processing Volume 2006.

[5] Onur ATAR, Murat H. SAZLI, Hakkı Gökhan ILK"FPGA implementation of Turbo Decoder Using BCJR algorithm.TUBITAK UZAY,Ankara University

[6] Sangjin Hong and Wayne .E. Stark Design and Architecture Implementation of a low complexity VLSI Turbo decoder or Low Energy Mobile Wireless communication. Journal of VLSI signal processing system 24, 43–57 (2000).

[7] "Turbo Coding ,Turbo equalization and space coding,"L. Hanzo, T.H. Liew, B.L. Year Department of Electronics and Computer Science, University of Southampton, UK

[8] M.May, T.Ilnseher, N.Wehn, and W.Raab,"A , 150Mbit/s 3GPP/LTE turbo code decoder," in proc. Design Autom. Test in Euro. Conf. Exhib. (DATE),2010,pp 1420-1425

[9] L.Hanzo, J.P. Woodard, and P.Robertson, "Turbo decoding and detection for wireless applications," Proc IEEE , Vol.95, no.6, pp.1178- 1200, June 2007