1-1-2005

Box-minus operation and application in sum-product algorithm

S Tong
Xidian University, sheng@uow.edu.au

P Wang
Xidian University

D Wang
Xidian University

X Wang
Xidian University

Box-minus operation and application in sum-product algorithm

Abstract

A new formula for box-minus operation, which separates the box-minus operation into sign and reliability operations, was derived. Its application in the sum-product algorithm (SPA) was also investigated. Both box-plus operation and box-minus operation were divided into sign operation and reliability operation. The results show that on using the box-minus operation, SPA can be implemented with less decoding latency and complexity.

Keywords

product, algorithm, sum, application, operation, minus, box

Disciplines

Engineering | Science and Technology Studies

Publication Details

S. Tong, P. Wang, D. Wang & X. Wang, "Box-minus operation and application in sum-product algorithm," Electronics Letters, vol. 41, (4) pp. 197-198, 2005.

Box-minus operation and application in sum–product algorithm

S. Tong, P. Wang, D. Wang and X. Wang

A new expression for box-minus operation, i.e. the inverse of box-plus operation, is derived, with which the box-minus operation can be implemented by a small look-up table. Its application in the sum–product algorithm is investigated.

Introduction: For the practical application of low density parity check (LDPC) codes [1], the implementation of the associated decoding algorithm, i.e. the sum–product algorithm (SPA), should be carefully considered. As for this problem, there have already been some results [2–5]. An efficient parallel implementation in the log-likelihood ratio (LLR) domain for SPA is proposed in [4] leading to less decoding complexity and latency, which involves two core operations, i.e. box-plus operation [6] and its inverse, called box-minus operation.

In this Letter we derive a new expression for the box-minus operation, which separates the box-minus operation into sign and reliability operations and is more suitable for practical implementation.

Box-plus and box-minus operations: Denote the LLR of a binary random variable $u(\in\{\pm 1\})$ as $L(u)=\log(P(u=+1)/P(u=-1))$. Then, for two independent binary random variables u and v, the box-plus operation is defined as follows [6, 4]:

$$\begin{aligned}L(w)=L(u\oplus v)=L(u)\boxplus L(v)&=\log\frac{1+e^{L(u)}e^{L(v)}}{e^{L(u)}+e^{L(v)}}\\&=\mathrm{sign}(L(u))\,\mathrm{sign}(L(v))\min\{|L(u)|,|L(v)|\}\\&\quad+\log(1+e^{-|L(u)+L(v)|})-\log(1+e^{-|L(u)-L(v)|})\end{aligned}\tag{1}$$

where $w=u\oplus v$ and $\oplus$ denotes binary XOR operation. Equation (1) can be further written as

$$\begin{aligned}L(w)=L(u)\boxplus L(v)&=\mathrm{sign}(L(u))\,\mathrm{sign}(L(v))\times\{\min\{|L(u)|,|L(v)|\}\\&\quad+\log(1+e^{-(|L(u)|+|L(v)|)})-\log(1+e^{-||L(u)|-|L(v)||})\}\end{aligned}\tag{2}$$

In (2), the term next to the multiplication sign is non-negative, and thus the box-plus operation is divided into sign and reliability operations. Then the box-plus operation reduces to the computation of the function $g(x)=\log(1+e^{-|x|})$, which can be implemented by a small look-up table [4].

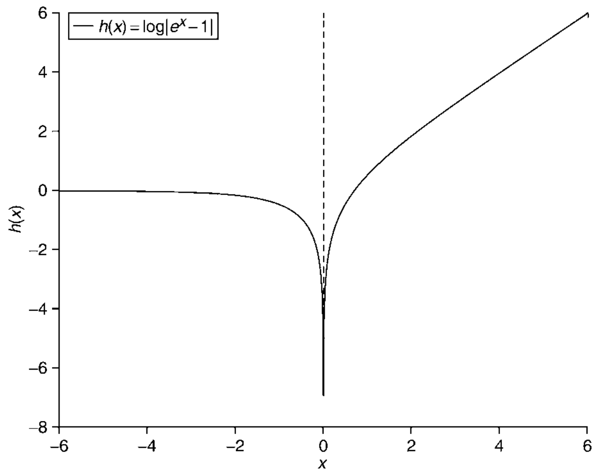

Fig. 1 *Function* $h(x)=\log|e^x-1|$

From (1), the box-minus operation can be defined as

$$L(u)=L(w)\boxminus L(v)=\log\frac{1-e^{L(w)}e^{L(v)}}{e^{L(w)}-e^{L(v)}}\tag{3}$$

Similar to the derivation of (2), the box-minus operation can be divided into sign and reliability operations as follows:

$$\begin{aligned}L(u)=L(w)\boxminus L(v)&=\mathrm{sign}(L(w))\,\mathrm{sign}(L(v))\times\{|L(w)|\\&\quad+\log(1-e^{-(|L(w)|+|L(v)|)})-\log(1-e^{-(|L(v)|-|L(w)|)})\}\end{aligned}\tag{4}$$

Since $|L(w)|<|L(v)|$ (see (2)), the term next to the multiplication sign in the above expression is non-negative. Thus, the box-minus operation reduces to the function $\hat{h}(x)=\log(1-e^x)$, $(x<0)$. A similar function $h(x)=\log|e^x-1|$ is defined in [4] for the implementation of box-minus operation, which is plotted in Fig. 1. It can be seen that as $x\to+\infty$, $h(x)\to+\infty$. Hence, $h(x)$ is not suitable to be implemented with a look-up table. However, $\hat{h}(x)$ is only the case of $h(x)$ when $x<0$, which can be easily implemented by a small look-up table.

Application of box-minus operation in SPA: Consider a check node w of degree n. Denote its n neighbouring variable nodes as $\{u_k\}_{k=1}^{n}$. In the horizontal step of SPA, the n LLRs of

$$\sum_{k\neq i}\oplus u_k,\quad (i=1,2,\ldots,n)$$

are calculated [2–4]. Obviously, these LLRs can be calculated as follows:

$$\begin{aligned}L\left(\sum_{k\neq i}\oplus u_k\right)&=\sum_{k\neq i}\boxplus L(u_k)=L(w)\boxminus L(u_i)\\L(w)=L\left(\sum_{k=1}^{n}\oplus u_k\right)&=\sum_{k=1}^{n}\boxplus L(u_k)\end{aligned}\tag{5}$$

In the conventional implementation, a forward–backward algorithm is used for the horizontal step, which involves $3(n-2)$ box-plus operations [4]. However, by first calculating $L(w)$ and then box-minusing $L(u_i)$ $(i=1,\ 2,\ \ldots,\ n)$, respectively, to get the n LLRs, the overall computation is $(n-1)$ box-plus operations and n box-minus operations. Since box-minus operation has almost the same computation complexity as box-plus operation, by the above technique the total computation complexity is reduced. In addition, the decoding latency is reduced from $O(n)$ to $O(\log(n))$ box-plus (or box-minus) operations [4]. Here we call this technique the parallel-excluding (PE) technique, which can be viewed as a message-passing schedule [7]. Denote the SPA implemented with PE as PE-SPA. Fig. 2 shows the bit error rate performance of a randomly constructed (1008,3,6) regular LDPC code, assuming an AWGN channel and BPSK modulation. Here, the box-minus operation in PE-SPA is implemented by a 5-bit look-up table.

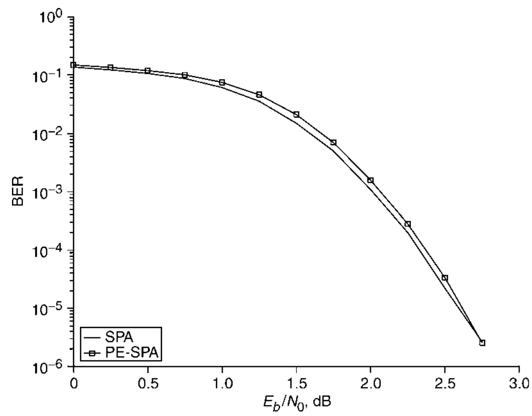

Fig. 2 *Performance of (1008,3,6) LDPC code*

Conclusion: Both box-plus operation and box-minus operation are divided into sign operation and reliability operation. By its new expression, the box-minus operation can be implemented by a small look-up table. Using the box-minus operation, SPA can be implemented with less decoding latency and complexity.

 17 October 2004

Electronics Letters online no: 20057447

doi: 10.1049/el:20057447

S. Tong, P. Wang, D. Wang and X. Wang (*State Key Lab of ISN, Xidian University, Xi'an, 710071, People's Republic of China*)

E-mail: ts_xd@163.com

References

1 Gallager, R.G.: 'Low-density parity-check codes' (MIT Press, Cambridge, MA, 1963)
2 Ping, L., and Leung, W.K.: 'Decoding low density parity check codes with finite quantization bits', *IEEE Commun. Lett.*, 2000, **4**, pp. 62–64
3 Leung, W.K., Lee, W.L., Wu, A., and Ping, L.: 'An efficient implementation technique of LDPC decoder', *Electron. Lett.*, 2001, **37**, (20), pp. 1231–1232
4 Hu, X.-Y., Eleftheriou, E., Amold, D.-M., and Dholakia, A.: 'Efficient implementation of the sum-product algorithm for decoding LDPC codes'. GLOBECOM, 2001
5 He, Y.-C., Sun, S.-H., and Wang, X.-M.: 'Fast decoding of LDPC codes using quantization', *Electron. Lett.*, 2002, **38**, (4), pp. 189–190
6 Hagenauer, J., Offer, E., and Papke, L.: 'Iterative decoding of binary block and convolutional codes', *IEEE Trans. Inf. Theory*, 1996, **42**, (2), pp. 429–445
7 Kschischang, F.R., and Frey, B.J.: 'Iterative decoding of compound codes by probability propagation in graphical models', *IEEE J. Sel. Areas Commun.*, 1998, **16**, (2), pp. 219–230