Fig. 3 *Measured S_{21} of bandstop filter with different PET tuning voltages*

Lung-Hwa Hsieh and Kai Chang (*Department of Electrical Engineering, Texas A&M University, College Station, Texas, 77843-3128, USA*)

E-mail: chang@ee.tamu.edu

References

1 YEO, K.S.K., and LANCASTER, M.J.: 'The design of microstrip six-pole quasi-elliptic filter with linear phase response using extracted-pole technique', *IEEE Trans. Microw. Theory Tech.*, 2001, **49**, (2), pp. 321–327

2 BELL, H.C.: 'Single-passband single-stopband narrow-band filters', *IEEE Trans. Microw. Theory Tech.*, 2000, **48**, (12), pp. 2472–2475

3 HONG, J.-S., LANCASTER, M.J., GREED, R.B., JEDAMZIK, D., MAGE, J.-C., and CHALOUPKA, H.J.: 'A high-temperature superconducting duplexer for cellular base-station applications', *IEEE Trans. Microw. Theory Tech.*, 2000, **48**, (8), pp. 1336–1343

4 YUN, T.-Y., and CHANG, K.: 'A low loss time-delay phase shift controlled by piezoelectric transducer to perturb microstrip line', *IEEE Microw. Guid. Wave Lett.*, 2000, **10**, (3), pp. 96–98

5 BUCHANAN, R.C. (Ed.): 'Ceramic material for electronics' (Marcel Dekker, New York), Chap. 3

Parallelised max-Log-Map model

K.K. Loo, K. Salman, T. Alukaidey and S.A. Jimaa

A parallelised max-Log-MAP model (P-max-Log-MAP) that exploits the sub-word parallelism and very long instruction word architecture of a microprocessor or a digital signal processor (DSP) is presented. The proposed model reduces considerably the computational complexity of the max-Log-MAP algorithm; and therefore facilitates easy implementation.

Introduction: The simplified logarithm domain of maximum *a posteriori* (MAP) algorithm [1], the max-Log-MAP algorithm [2], work on a large number of add-compare-select (ACS) operations which are the basic and most intensive operations. The ACS is executed in sequence to compute trellis metrics recursively for each trellis state in the forward and backward manner over a huge data volume underlying the trellis. It can be shown that the max-Log-MAP algorithm lends itself to parallelism. In this Letter, the proposed P-max-Log-MAP fully exploits the sub-word parallelism (SWP) and very long instruction word (VLIW) architecture of a microprocessor or a digital signal processor (DSP) to achieve high level data parallelism. The SWP allows single instructions for the processing of several different data in parallel within a defined data width. With the combination of VLIW architecture, at least two SWP instructions can be executed in parallel, i.e. a greater data parallelism can be achieved.

Parallelised max-Log-MAP: Consider a max-Log-MAP algorithm for decoding 3GPP turbo codes with constraint length $K = 4$ and polynomial generator, $g(D) = 15/13$ [3]. We show that the algorithm's forward metrics, the backward metrics, and LLR computations can be highly parallelised. Here, assumptions are made where the forward recursion is first executed, followed by the backward recursion in the same loop with the logarithm likelihood ratio (LLR) computation. In addition, we assume that a typical SWP architecture of a processing unit is either of a microprocessor or a DSP that supports 8-bit ALU operations as minimum. With data widths of 64 bits, the SWP is capable of computing eight ACS operations over eight 8-bit different data sets individually, in parallel. The computations of eight forward/backward metrics usually require 16 add/sub operations and eight max operations. However, using the single VLIW's SWP instruction, 16 add/sub operations can be performed in one cycle, half cycles for each add and sub operation, and eight max operations in one cycle. Our model also requires that the data positions within a defined field to be arranged for a proper match of computation. For example, an arbitrary vector $Z_s = \{Z_0, Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7\}$ may be arranged to $\tilde{Z}_s = \{Z_4, Z_7, Z_3, Z_6, Z_2, Z_0, Z_5, Z_1\}$ which matches another arbitrary vector $V_s = \{V_4, V_7, V_3, V_6, V_2, V_0, V_5, V_1\}$ for possible computations. The arrangement can be done by a simple mapping of $Z_{p(s)} \mapsto Z_s$ where the permutation index, p, is related to Z_s.

Forward recursion: The forward metrics $\alpha_x = \alpha_{0, \ldots, 7}$ are initialised are stored in a register as shown in Fig. 1. The branch metrics, $\gamma_{11}\gamma_{10}$, are retrieved from the memory and arranged according to the trellis branch transitions. The branch metrics of other polynomial generators within $K = 4$ can be arranged according to the new branch transitions with a corresponding permutation index, p. The metrics $\alpha_x = \alpha_{0, \ldots, 7}$ at time index $k - 1$ will add/sub with the corresponding branch metrics individually to yield α_s^+ and α_s^- that represent the metrics related to information '1' and '0', respectively. After the operation, the position of α_x^- is arranged to match α_x^+, then a comparison is applied to select the survival metrics as the new metrics, $\hat{\alpha}_x$, at index k. The $\hat{\alpha}_x$ are arranged and updated to register α_s and a copy is stored in the memory where they will be retrieved when computing the LLR.
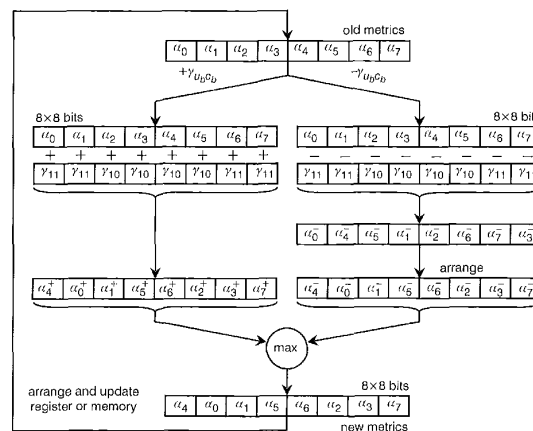

Fig. 1 *Computational flow of forward recursion*

Backward recursion: Fig. 2 shows the computational flow of the backward recursion which is identical to the forward recursion. Therefore, the backward metrics can be computed using the forward recursion method with the data collected form the backward trellis trace. The backward metrics are computed in the same loop with the LLR computation. As the new backward metrics are being computed, the metrics β_x^+ and β_x^- that are related to information '1' and '0' are kept in the register for LLR computation. The reference 'A' in Fig. 2 marks this situation. The new metrics are updated to β_s and are used for computing the next set of metrics while executing the LLR computation. To eliminate extra memory, the metrics are not required

to be stored in the memory. Next, we discuss the LLR computation which completes the backward recursion operations.
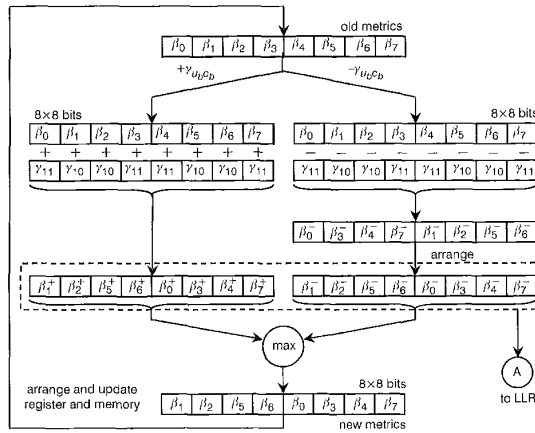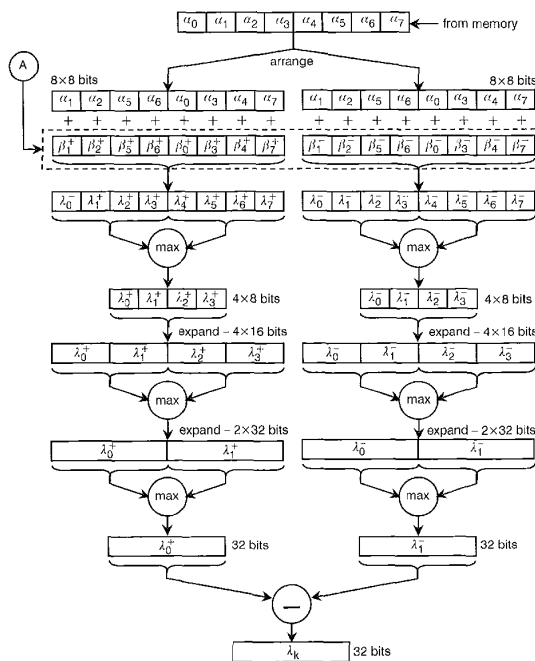

Fig. 2 *Computational flow of backward recursion*

Fig. 3 *Computational flow of LLR computation*

LLR computation: Fig. 3 shows the computational flow of the LLR which is a continuation from the reference 'A' in Fig. 2. Both the backward metrics, β_s^+ and β_s^- will be added to the forward metrics α_s in a matched position, yielding the *a posteriori* probability soft values for information '1' and '0' represented as $\lambda_{s=0,\ldots,7}^+$ and $\lambda_{s=0,\ldots,7}^-$, respectively. The rest of the LLR computation is to find the maximum values of $\lambda_{s=0,\ldots,7}^+$ and $\lambda_{s=0,\ldots,7}^-$ using a unique search procedure as shown in Fig. 3. Finally, the LLR soft value, λ_k, is calculated by finding the ratio between the two maximum values as in (1):

$$\lambda_k = \max[\lambda_{s=0,\ldots,7}^+] - \max[\lambda_{s=0,\ldots,7}^-].\qquad(1)$$

Implementation results: The proposed P-max-Log-MAP was implemented on analogue devices TigerSHARC dual-cores DSP. The input data is quantised into a minimum of 8-bit signed integer format. In addition to saving a great deal of memory resources, multiple 8-bit operations can also be used. For a single-core operation of the TigerSHARC, 16 8-bit data can be processed in parallel on the single VLIW's SWP instruction, as shown in Table 1, which also depicts the implementation results of the P-max-Log-MAP.

Table 1: Cycle count

P-max-Log-MAP	Latency (cycle)	Memory (bit)
Branch metrics	$7 + (N/16) \times 8$	$2N \times 8$
Forward metrics	$8 + (N \times 5)$	$8N \times 8$
Backward metrics	$10 + (N \times 20)$	–
LLR		$N \times 16$
Total	$25 + 25.5N$	$96N$
Turbo decoder 2 × P-max-Log-MAP	$50 + 51N$	$96N$

Conclusion: The result show that the proposed model implementing the 3GPP turbo decoder can decode, with two iterations, 3G data traffic channels at a rate beyond 2 Mbit/s on the 250 MHZ Tiger-SHARC DSP.

K.K. Loo, T. Alukaidey and S.A. Jimaa (*Department of ECEE, University of Hertfordshire, AL10 9AB, United Kingdom*)

E-mail: k.k.loo@herts.ac.uk

K. Salman (*NC A&T State University, USA*)

E-mail: ksalman@ncat.edu

References

1 BAHL, L.R., COCKE, J., JELINEK, F., and RAVIV, J.: 'Optimal decoding of linear codes for minimising symbol error rate', *IEEE Trans. Inf. Theory*, 1974, pp. 284–287
2 ROBERTSON, P., and HOEHER, P.: 'Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding', *Eur. Trans. Telecommun.*, 1997, **8**, (2), pp. 119–125
3 '3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Multiplexing and Channel Coding (FDD)', 3GPP TS25.212 v3.6.0, July 2001

Pilot symbol initiated sub-optimal sequence estimation algorithm for QAM signals

Jong-Ho Lee, Seong-Cheol Kim and Jae Choong Han

A sub-optimal decision directed decoding algorithm based on the expectation-maximisation algorithm is proposed. The algorithm performs iterative sequence estimation assuming quadrature amplitude modulation signals in a frequency non-selective fading channel. The iteration begins using periodically inserted pilot symbols, and it was observed that the algorithm converges mostly within two iterations. The bit error rate performances of the proposed algorithm are evaluated using computer simulation. The results show that the proposed algorithm performs better than conventional schemes.

Introduction: Applying the M-ary QAM scheme for mobile communications is attractive due to its inherent spectral efficiency. However, channel fading introduces amplitude as well as phase distortion. Thus, a fading compensation technique is required for coherent decoding. Fading compensation techniques using periodically inserted pilot symbols in the data stream have been studied in the literature [1]. It is shown that the technique is simple to implement while providing good performance. The technique uses pilot symbols to produce fading estimates, which are used for coherent decoding of QAM symbols. The performance of the technique can be further improved if decoded symbols are used for channel estimation via some iterative algorithm such as the expectation maximisation (EM) algorithm [2]. It is well known that the EM algorithm provides maximum likelihood (ML) estimate under some conditions. The application of the EM algorithm for a fading channel is introduced in [3]. However, implementation of the EM algorithm for QAM scheme is too complicated. In this Letter, the EM algorithm is investigated for QAM signals, and a sub-optimal pilot symbol initiated decision directed decoder is proposed.