Simplified Log-MAP Algorithm for Very Low-Complexity Turbo Decoder Hardware Architectures

Simplified Log-MAP Algorithm for Very Low-Complexity Turbo Decoder Hardware Architectures

Maurizio Martina, *Member, IEEE*, Stylianos Papaharalabos, *Member, IEEE*, P. Takis Mathiopoulos, *Senior Member, IEEE*, and Guido Masera, *Senior Member, IEEE*

***Abstract*— Motivated by the importance of hardware implementation in practical turbo decoders, a simplified, yet effective, n-input max* approximation algorithm is proposed with the aim being its efficient implementation for very low-complexity turbo decoder hardware architectures. The simplification is obtained using an appropriate digital circuit for finding the first two maximum values in a set of n data that embeds the computation of a correction term. Various implementation results show that the proposed architecture is simpler by 30%, on average, than the constant logarithmic-maximum *a posteriori* (Log-MAP) one, in terms of chip area with the same delay. This comes at the expense of very small performance degradation, in the order of 0.1 dB for up to moderate bit error rates, e.g., 10^{-5}, assuming binary turbo codes. However, when applying scaling to the extrinsic information, the proposed algorithm achieves almost identical Log-MAP turbo code performance for both binary and double-binary turbo codes, without increasing noticeably the implementation complexity.**

***Index Terms*— Digital circuit, logarithmic maximum *a posteriori* (Log-MAP), Max-Log-MAP, turbo codes.**

I. Introduction

THE max* operation [1] is the computational kernel of the logarithmic maximum *a posteriori* (Log-MAP) algorithm employed in the decoding of turbo and low-density parity-check codes. These codes have become a reference for several applications, such as magnetic disk reliability and telemetry [2], [3]. For this reason, several research efforts have been devoted to reduce the complexity of the max* operation both at algorithmic and implementation level, to obtain low-complexity very large scale integration architectures [4]–[10]. Since the publication of [1], the computation of the n-input max* operation has been implemented by recursively applying the two-input max* operation (over $n-1$ times) as

$$\max^*\{x_1, x_2\} = \max\{x_1, x_2\} + f_c\left(|x_1 - x_2|\right) \tag{1}$$

where $f_c(x) = \log(1 + \exp\{-x\})$ been usually referred to as correction term. Indeed, many research works, including [4]–[8], [11], concentrate on low-complexity implementations of (1). Even if low-complexity algorithms for the computation of the two-input max* operation lead to these previously published efficient architectures, in this paper it is shown that further hardware complexity can be saved by analyzing the n-input max* operation as a whole.

Specifically, this paper is motivated by [12], where it was shown that by exploiting the Chebyshev inequality the n-input max* operation can be approximated as

$$\max^*\{\mathcal{X}\} \approx y_1 + \log[(1 + K_1 \exp\{-\delta\})] + K_2 \tag{2}$$

where $\mathcal{X} = \{x_1, \ldots, x_n\}$ is the set of the n input values, $y_1 = \max\{\mathcal{X}\}$, $y_2 = \max\{\mathcal{X} \setminus y_1\}$ are the first and second maximum values in $\mathcal{X}$, respectively, $\delta = y_1 - y_2$, $K_1 = (n-1)/n$, and $K_2 = \log[2n/(n+1)]$. To the best of our knowledge, the only implementations of the n-input max* operation presented in [12], which are available in the open technical literature, are the ones proposed in [13] and [14]. In particular, [13] describes the design and implementation of a radix-16 modified Log-MAP decoder architecture based on [12] that achieves a maximum throughput of 502 Mb/s and outperforms, in terms of chip power efficiency, previously published MAP processors available in the open technical literature. Furthermore, in [14], it has been shown that both the Log-MAP approximation of (2) and the constant Log-MAP algorithm [4] achieve nearly an optimal bit error rate (BER) performance. However, the Log-MAP architecture presented in [14] for approximating (2) was more complex than $n-1$ instances of the constant Log-MAP architecture [4]. Thus, the efficient hardware implementation of (2) leading to a very low-complexity turbo decoding architecture is still an open problem.

Motivated by the above, this paper has the following contributions: 1) it shows that (2) can be further simplified without compromising its near optimal turbo code BER performance and 2) it obtains a very low-complexity Log-MAP architecture, which is significantly simpler than the constant Log-MAP architecture. The design of this novel n-input max* approximation is based upon: 1) the search of the first two maximum values among the n possible inputs and 2) the use of a constant correction term added to the first maximum value. As it will be shown in the subsequent Section IV of this paper, the proposed architecture not only requires lower area than constant Log-MAP algorithm [4] (e.g., 30% area savings, on average,

Manuscript received February 25, 2013; revised May 29, 2013; accepted July 14, 2013. Date of publication September 24, 2013; date of current version February 5, 2014. The Associate Editor coordinating the review process was Dr. Jesus Urena.

M. Martina and G. Masera are with the Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Torino 10129, Italy (e-mail: maurizio.martina@polito.it).

S. Papaharalabos is with the Institute for Astronomy Astrophysics Space Applications and Remote Sensing, National Observatory of Athens, Athens 152 36, Greece.

P. T. Mathiopoulos is with the Institute for Astronomy Astrophysics Space Applications and Remote Sensing, National Observatory of Athens, Athens 152 36, Greece, and also with the Department of Informatics and Telecommunications, National and Kapodestrian University of Athens, Athens 157 84, Greece.

Digital Object Identifier 10.1109/TIM.2013.2281554

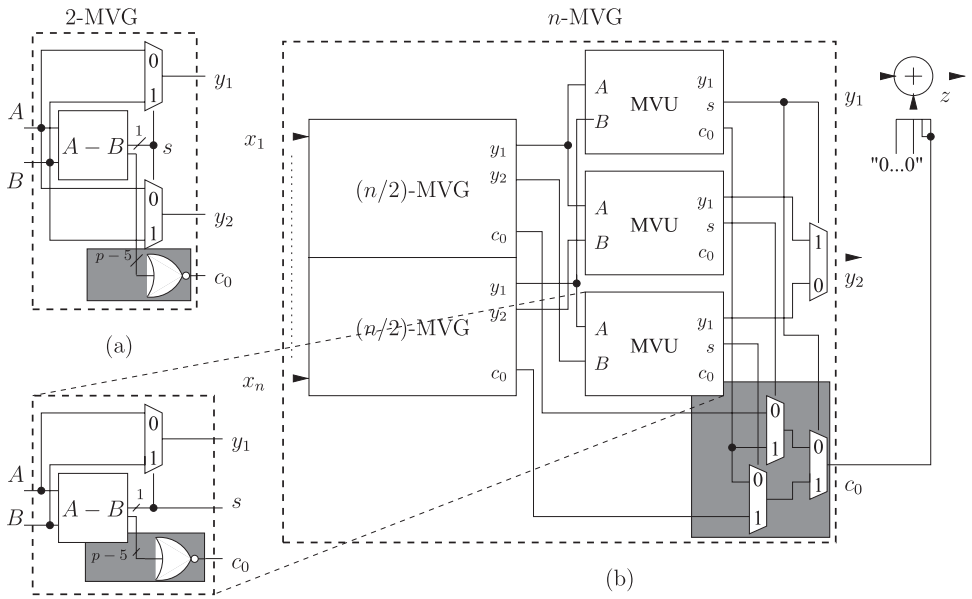

Fig. 1. Block diagram of modified architectures shown by the gray-shaded blocks. (a) Two MVG and (b) n-input max* approximation.

with the same delay for both architectures) but it is also simpler than the solutions presented in [13] and [14]. Furthermore, when applying scaling to the extrinsic information almost identical Log-MAP turbo code BER performance is obtained with the proposed algorithm for both binary and double-binary turbo codes. Hardware complexity results are also provided for some practical implementations of MAP decoders based on the universal mobile telecommunication system-long-term evolution (UMTS-LTE) [15], Consultative Committee for Space Data Systems (CCSDS) [3], and Worldwide Interoperability for Microwave Access (Wi-MAX) [16] turbo codes. It is shown that for these codes, the proposed n-input max* architecture features the lowest area among the Log-MAP implementations.

This paper is structured as follows. After the introduction, Section II details the proposed simplified n-input max* architecture, leading to very low-complexity turbo decoding hardware architecture. Section III deals with the BER performance evaluation results of the proposed n-input max* approximation algorithm and comparison against other algorithms for both binary and double-binary turbo codes. Next, Section IV shows the hardware implementation results of the proposed simplified architecture, including some practical implementations and comparisons of MAP decoders based on the UMTS-LTE, CCSDS, and Wi-MAX turbo codes. Finally, in Section V conclusion is drawn.

II. Simplified n-Input max* Architecture

Since K_2 in the right-hand side of (2) is a positive constant it can be ignored in the iterative decoding process and for large values of n, $K_1 \approx 1$, so (2) is further simplified to

$$\max^*\{\mathcal{X}\} \approx y_1 + \log\left(1 + \exp\{-\delta\}\right) = y_1 + f_c\left(\delta\right). \quad (3)$$

The implementation of (3) relies on finding y_1, y_2, computing δ and then applying to y_1 a correction term that depends on δ. An efficient architecture, based on a tree structure of maximum-value generators (MVGs), has been proposed in [17] and generalized in [18] for finding y_1 and y_2 in a set of n elements. The structure used for n inputs (n-MVG) in [17] is derived recursively from two $(n/2)$-MVG architectures and three maximum-value units (MVUs), as shown in Fig. 1(a) and (b), where s is the sign of $A-B$. This tree structure is the best known architecture to find y_1 and y_2. Therefore, a straightforward implementation of (3) is composed of an n-MVG architecture followed by a substractor to compute δ, a lookup table (LUT), which stores precomputed $f_c(\delta)$ values and an adder. To improve this solution, it is shown here, for the first time in the open technical literature, that: 1) $f_c(\delta)$ in (3) can be effectively implemented resorting to a modified version of the approximation proposed in [4] for (1) and 2) a modified version of the MVG-based architecture is proposed to incorporate the generation of the correction term.

As a previous art baseline and for comparison purposes, the numerical results reported in [8] show that the constant Log-MAP algorithm is the lowest complexity Log-MAP approximation with near-optimal turbo code BER performance. The constant Log-MAP algorithm can be described as follows: A 2's complement p-bit fixed-point data representation with three fractional bits (i.e., $m = 3$) approximates the smallest nonzero value of $f_c(x)$ as 1/8. Thus, there exist a value $\pm\hat{x}$, such that $f_c(|\hat{x}|) = 1/8$ ($|\hat{x}| \approx 2$). Therefore, if $|x| \geq 2$ then $f_c(x) \approx 0$. On the other hand, according to [4], when $|x| < 2$, $f_c(x)$ can be approximated by its mean value in [0, 2), that is $f_c(x) \approx 1/2 \cdot \int_0^2 f_c(t)\,\mathrm{d}t$, whose best representation with $m = 3$ is 3/8. Thus, the constant Log-MAP algorithm approximates $f_c(x)$ as

$$f_c(x) \approx \begin{cases} 3/8, & -2 < x < 2 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

resulting in a simple logic circuit to selectively add a constant to the output of the maximum selection circuit [4].

In particular, the constant Log-MAP algorithm achieves area savings of 40% with a BER performance loss of 0.03 dB compared with Log-MAP turbo decoding [4]. In this paper, it is shown that the complexity of the constant Log-MAP algorithm can be reduced more by implementing the correction term $f_c(x)$ as follows.

Since $\delta \geq 0$ the implementation of $f_c(\delta)$ in (3) can be further simplified. Let c be the value taken by $f_c(\delta)$ for a given value of δ and $c = c_{p-1} \ldots c_0$, $\delta = \delta_{p-1} \ldots \delta_0$ being their p-bit fixed-point representations with $m = 3$. Then, it is observed that 1) if $c = 3/8$ then $c_{p-1} \ldots c_0 =$ "0 . . . 0.011" otherwise $c_{p-1} \ldots c_0 =$ "0 . . . 0.000" (i.e., $c_0 = c_1$ and $c_i =$ "0" for $2 \leq i \leq p-1$). 2) if $\delta < 2$ (note that $\delta \geq 0$ so $\delta_{p-1} =$ "0") then $\delta_{p-2} \ldots \delta_0 =$ "0 . . . 0x.xxx", where 'x' is a do not care bit (i.e., $\delta_j =$ '0' for $4 \leq j \leq p-2$). Thus, c_0 and c_1 are 1 only when $\delta_j =$ '0' for $4 \leq j \leq p-2$

$$\overline{c}_0 = \overline{c}_1 = \bigvee_{j=4}^{p-2} \delta_j \qquad c_i = 0, \qquad 2 \leq i \leq p-1 \tag{5}$$

where $\overline{(\cdot)}$ and $\bigvee$ represent the NOT and OR logic operations, respectively. Since δ is already computed inside the n-MVG architecture, the architectures of two-MVG and MVU blocks are modified to embed the computation of c_0 into the n-MVG architecture, so as to obtain $z = y_1 + f_c(\delta)$. To the best of our knowledge this is a novel contribution, the implementation of which is explicitly shown by the gray-shaded blocks in Fig. 1(a) and (b). Therefore, the whole implementation of (3) can be obtained by introducing a few additional gates in the n-MVG architecture. Clearly, the proposed implementation for the correction term takes advantage of $\delta \geq 0$, so that the complexity of the proposed correction term is roughly half the complexity of the correction term of the constant Log-MAP algorithm.

III. BER Performance Evaluation

In this section, the proposed n-input max* approximation method is evaluated in terms of turbo code BER performance versus E_b/N_o, where E_b is the bit energy and N_o is the one-sided power spectral density of an additive white Gaussian noise channel. In particular, performance evaluation results have been obtained for the following algorithms: 1) constant Log-MAP [4]; 2) the proposed algorithm from (3) and (4); 3) Log-MAP with 3-b LUT [1]; and 4) the least complex Max-Log-MAP algorithm [1]. A 16-states turbo code is considered with overall coding rate equal to $R = 1/2$ and generator polynomials $(1, 33/23)_o$ in octal form representing the feed-forward and backward polynomials, respectively. This turbo code is used in the CCSDS standard [3]. An information sequence of $N = 10^3$ bits is considered with the total number of transmitted frames being 10^6. To ensure the accuracy of the performance evaluation results, in the simulation experiments at least 100-b errors have been counted for each performance point, which has been obtained. A pseudorandom turbo interleaver is considered and at the receiver a maximum of 10 decoding iterations are performed. The modulation type has been selected to be binary phase-shift keying (BPSK).

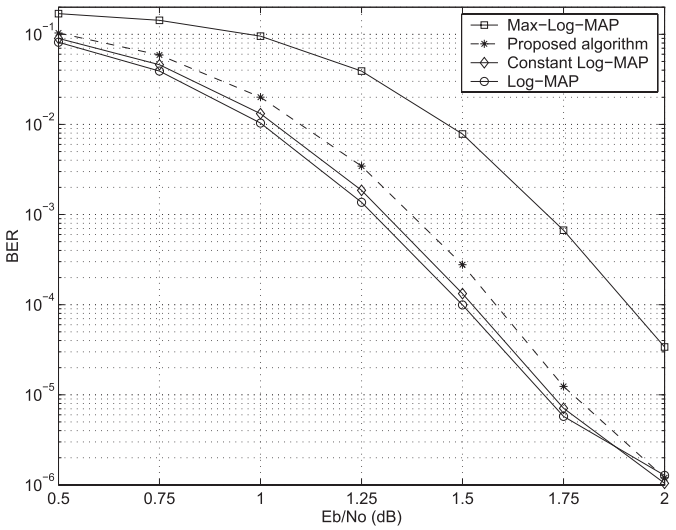

Fig. 2. BER performance comparison. Max-Log-MAP, constant Log-MAP, Log-MAP, and the newly proposed algorithm. A binary turbo code is assumed as in the CCSDS [3] standard.

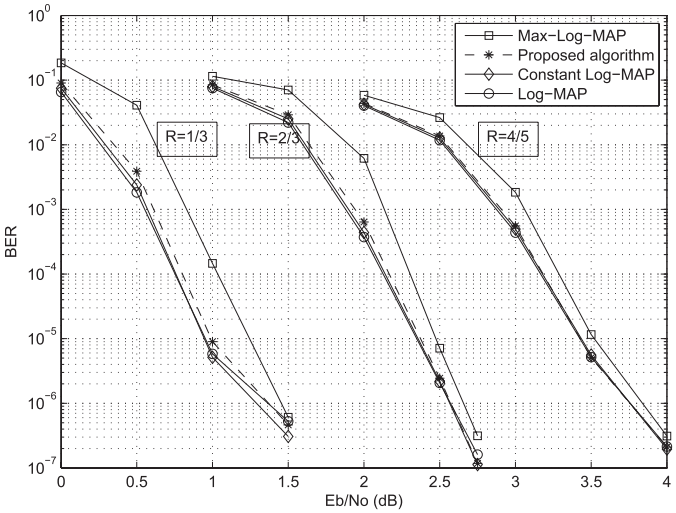

Fig. 3. As in Fig. 2, but considering a QPSK modulation format and employing a double-binary turbo code as in the Wi-MAX standard [16] with different coding rates R.

Additionally, a double-binary turbo code is considered having eight-states component convolutional codes as proposed in Wi-MAX standard [16] with the overall coding rates $R = 1/3$, 2/3, and 4/5, and information sequence length of $N = 752$ bit couples (MPEG packets). The modulation type in this case is quadrature phase-shift keying (QPSK). At the receiver, a maximum of 10 decoding iterations are performed. Notice that in computer-based simulations, BPSK/QPSK modulation is implemented with an antipodal baseband signaling representation, i.e., carrier frequency is not considered.

As shown in Figs. 2 and 3, the proposed algorithm shown with dashed lines always achieves the near-optimal BER performance, which is significantly better than the equivalent performance achieved by the Max-Log-MAP algorithm. The proposed algorithm is slightly inferior to the performance of the constant Log-MAP algorithm up to moderate BERs (e.g.,

TABLE I
REQUIRED E_b/N_o (IN dB) AT BER $= 10^{-5}$ FOR BINARY TURBO CODE AND AT BER $= 10^{-6}$ FOR DOUBLE-BINARY TURBO CODE WITH CONSTANT SCALING DENOTED AS sc. R DENOTES OVERALL CODING RATE

Decoding Algorithm	Binary Turbo Code $R = 1/2$	Double-binary Turbo Code $R = 1/3$	Double-binary Turbo Code $R = 2/3$	Double-binary Turbo Code $R = 4/5$
Max-Log-MAP	1.7 ($sc = 0.65$)	1.3 ($sc = 0.75$)	2.55 ($sc = 0.75$)	3.75 ($sc = 0.75$)
Proposed algorithm	1.62 ($sc = 0.8$)	1.25 ($sc = 0.85$)	2.45 ($sc = 0.85$)	3.65 ($sc = 0.85$)
Constant Log-MAP	1.6 ($sc = 0.85$)	1.25 ($sc = 0.9$)	2.45 ($sc = 0.9$)	3.65 ($sc = 0.9$)
Log-MAP	1.6 ($sc = 0.9$)	1.25 ($sc = 0.9$)	2.45 ($sc = 0.9$)	3.65 ($sc = 0.9$)

performance degradation less than 0.1 dB at BER $= 10^{-5}$ assuming the binary turbo code of Fig. 2) since the former algorithm is a simplified version of the latter algorithm. It is also observed from Fig. 2 that at low BERs, e.g., 10^{-6} the proposed algorithm achieves essentially identical performance with both constant Log-MAP and Log-MAP algorithms. As additionally shown in Fig. 3, the BER performance degradation between the proposed algorithm and the constant Log-MAP becomes even smaller for the double-binary turbo code since for this turbo code the BER performance difference between the Log-MAP and Max-Log-MAP is smaller than in the case of binary turbo code [19]. Especially, for $R > 1/2$ it is observed that the proposed algorithm achieves an identical performance with both constant Log-MAP and Log-MAP algorithms at BER $= 10^{-6}$ and lower.

To further improve the BER performance, constant scaling to the extrinsic information has been applied [19], [20], which adds a very small increase in the implementation complexity, i.e., two extra multiplications per decoding iteration. This is denoted as sc in Table I. The best performing values for scaling were found by trial and error for all the investigated algorithms, although they can be computed analytically off-line using extrinsic information transfer charts at the expense of small complexity increase [21]. The obtained performance evaluation results for both binary and double-binary turbo codes with constant scaling are summarized in Table I. From this table, it is noticed that scaling improves BER performance and the proposed algorithm achieves an essentially optimal BER performance for both binary and double-binary turbo codes.

IV. HARDWARE IMPLEMENTATION

Apart from the detailed BER performance evaluation results presented earlier, in this section, the proposed n-input max* approximation method is evaluated in terms of hardware complexity. In Table II, the proposed n-input max* architecture (denoted as $\mathcal{C}$) is compared in terms of area (A) and delay (D) with two tree structures being implemented as: 1) two-input Log-MAP with 3-b LUT architecture (denoted as $\mathcal{A}$) [1] and

TABLE II
POSTSYNTHESIS RESULTS: AREA (EQ. GATES) AND DELAY (ns) COMPARISON AMONG: TWO-INPUT LOG-MAP ARCHITECTURE WITH 3-b LUT ($\mathcal{A}$) [1], TWO-INPUT CONSTANT LOG-MAP ARCHITECTURE ($\mathcal{B}$) [4], AND THE PROPOSED n-INPUT max* ARCHITECTURE ($\mathcal{C}$)

n,p	$\mathcal{A}$		$\mathcal{B}$		$\mathcal{C}$		
	$A_{\mathcal{A}}$	$D_{\mathcal{A}}$	$A_{\mathcal{B}}$	$D_{\mathcal{B}}$	$A_{\mathcal{C}}$	$D_{\mathcal{C}}$	$A_{\mathcal{C}}$ @ $D_{\mathcal{B}}$
4,8	1096	1.40	568	1.20	484	1.05	379 (-33.3%)
4,12	1445	1.50	769	1.45	712	1.20	605 (-21.3%)
4,16	2140	1.55	1245	1.45	1075	1.20	895 (-28.1%)
8,8	2427	2.05	1432	1.80	1299	1.35	961 (-32.9%)
8,12	3596	2.10	2255	2.05	1980	1.55	1475 (-34.6%)
8,16	4615	2.20	3145	2.15	2533	1.60	2053 (-34.7%)
16,8	5416	2.60	3213	2.35	2610	1.75	2160 (-32.8%)
16,12	7920	2.70	4907	2.65	4078	2.00	3406 (-30.6%)
16,16	11068	2.85	7312	2.80	5768	2.00	4515 (-38.3%)

2) two-input constant Log-MAP architecture (denoted as $\mathcal{B}$) [4]. Postsynthesis results have been obtained via Synopsys Design Compiler on a 90-nm standard cell technology for $n = 4, 8, 16$ and $p = 8, 12, 16$ being the number of inputs for the max* operation and the data width, respectively. All the implementations rely on $m = 3$ fractional bits, for this reason $p - m$ integer bits are assumed. Various experimental results have confirmed that the proposed architecture is not only significantly smaller than the ones obtained by recursively applying the Log-MAP LUT-based and constant Log-MAP approximation, but it can also run at higher clock frequency as it features lower delay. As an example, assuming $n = 16$ and $p = 16$ in Table II, the proposed n-input max* architecture requires 5768 equivalent gates for a minimum delay of 2 ns, whereas the two-input constant Log-MAP architecture requires 7312 equivalent gates for a minimum delay of 2.8 ns. Thus, the proposed n-input max* architecture features 21% less area and has 28.5% lower delay than the two-input constant Log-MAP architecture [4]. However, to have a fair area comparison the architectures should have the same delay. Thus, the delay constraint of the proposed architecture has been relaxed, namely the logic synthesizer has been constrained to achieve $D_{\mathcal{B}}$ that is the same minimum delay D of the constant Log-MAP architecture $\mathcal{B}$. The corresponding implementation results are shown in the last column of Table II. As it can be observed, the complexity saved by the proposed architecture is approximately 30%, on average, as compared with the complexity of the constant Log-MAP architecture, under the condition to achieve the same delay $D_{\mathcal{B}}$.

In [13], the n-input max* operation presented in [12] is adopted for the design of a power efficient Log-MAP decoder. Unfortunately, in [13] only complexity results of the whole MAP decoder are given. Thus, for a fair comparison the n-input max* architecture proposed in [13] that is sized for $n = 16$, $p = 11$ and relies on three stages, has been also implemented. The first stage contains a dynamic range selector

TABLE III
POSTSYNTHESIS AREA RESULTS [μm^2] FOR TURBO CODE USED IN UMTS-LTE [15] AND CCSDS [3] STANDARDS WITH A TARGET CLOCK FREQUENCY OF 200 MHZ

	UMTS-LTE		CCSDS	
	APO	MAP decoder	APO	MAP decoder
Proposed architecture	12151.83 (68.2%)	32192.28 (85.0%)	26228.56 (61.7%)	63687.45 (79.7%)
Log-MAP Delta [14]	13299.13 (74.6%)	33339.58 (88.1%)	31088.03 (73.2%)	68546.92 (85.7%)
Constant Log-MAP	12318.35 (69.1%)	32358.80 (85.5%)	29709.29 (69.9%)	67168.18 (84.0%)
Log-MAP	17820.63 (100%)	37861.08 (100%)	42483.47 (100%)	79942.36 (100%)

TABLE IV
AS IN TABLE III, BUT FOR TURBO CODE USED IN Wi-MAX [16] STANDARD. α AND β: FORWARD AND BACKWARD METRICS, RESPECTIVELY

	Wi-MAX		
	α/β	APO	MAP decoder
Proposed architecture	15101.25 (67.6%)	28052.54 (69.6%)	71559.83 (72.8%)
Log-MAP Delta [14]	16547.72 (74.1%)	31246.06 (77.6%)	77646.29 (79%)
Constant Log-MAP	16066.20 (71.9%)	29284.50 (72.7%)	74721.69 (76.0%)
Log-MAP	22338.12 (100%)	40289.06 (100%)	98270.09 (100%)

to find b, the position of the leading 1 bit in the p bits of the n input data, where b is the most significant bit. The second stage performs a coarse search of y_1 using only three bits for each input. The third stage uses the four least significant bits of each input to refine the search of y_1, identify y_2, and compute $f_c(\delta)$. Postsynthesis results for the same 90-nm standard cell technology used for the proposed architecture show that the three-stage architecture presented in [13], denoted as $\mathcal{T}$, requires $A_{\mathcal{T}} = 3513$ equivalent gates for a minimum delay $D_{\mathcal{T}} = 2.45$ ns. For the same parameters ($n = 16$, $p = 11$), the architecture proposed in this paper, requires $A_{\mathcal{C}} = 3976$ equivalent gates for a minimum delay of $D_{\mathcal{C}} = 1.95$ ns. Considering the same minimum delay as the three-stage architecture ($D_{\mathcal{T}} = 2.45$ ns), the area of the proposed architecture reduces to $A_{\mathcal{C}}@D_{\mathcal{T}} = 3027$ equivalent gates, which is 14% less than $A_{\mathcal{T}}$.

Regarding the area required to implement a MAP decoder based on the proposed n-input max* architecture, further novel results are shown for the UMTS-LTE [15] and CCSDS [3] (see Table III) as well as the Wi-MAX [16] (see Table IV) turbo code decoders. For binary codes (UMTS-LTE and CCSDS) the proposed n-input max* architecture can be exploited only in the computation of the *a posteriori* information. Thus, columns labeled as APO in Tables III and IV refer to the area of the hardware block required to compute the *a posteriori* information. Indeed, the computation of the *a posteriori* information requires us to: 1) add forward state metrics, backward state metrics, and branch metrics; 2) compute two and four max* operations for binary and double-binary codes, respectively, where each max* operation has a number of input values equal to the number of trellis states of the component convolutional code; and 3) perform one and three subtractions to compute the *a posteriori* information for binary and double-binary codes, respectively. Therefore, columns labeled as APO in Tables III and IV list a realistic result of the complexity reduction achievable in a MAP decoder with the proposed n-input max* architecture. On the other hand, for double-binary codes, the proposed n-input max* architecture can be exploited to reduce the complexity of the computation of forward (α) and backward (β) state metrics as well. The computation of state metrics has been accomplished by following the wrapping technique proposed in [22] for Viterbi decoding, and which has been employed subsequently for turbo decoding [23], [24]. The computation of *a posteriori* information has been implemented by extending the data width to prevent the overflow and performing saturation on the extrinsic information. The implementation results of Tables III and IV show that the proposed n-input max* architecture features the lowest area among the Log-MAP implementations in all the considered cases, even when compared with Log-MAP delta proposed recently in [14]. Results shown in Table III also highlight that with the proposed n-input max* architecture the amount of saved area increases as long as n increases. Indeed, more area is saved in the CCSDS case ($n = 16$) than in the UMTS-LTE case ($n = 8$). Moreover, by comparing the results in Table IV with the equivalent ones in Table III it is observed that for eight-state codes (UMTS-LTE versus Wi-MAX turbo codes) the proposed n-input max* architecture is more effective in reducing the area of double-binary codes (Wi-MAX) than the area of binary ones (UMTS-LTE). This figure of merit relies on the fact that in binary codes the proposed n-input max* architecture is used twice in the computation of the *a posteriori* information, whereas in double-binary codes it is used four times. Moreover, in double-binary codes the proposed n-input max* architecture can be exploited in the computation of state metrics as well.

V. CONCLUSION

A novel approximation for the n-input max* operation has been proposed and its efficient hardware architecture was explicitly shown. Both simulation and experimental hardware results confirmed that the proposed approximation requires from 21% to 38% smaller area than the constant Log-MAP algorithm with the same delay, while maintaining the near optimal Log-MAP turbo code performance, especially when scaling was applied in the extrinsic information. Furthermore, the proposed n-input max* architecture required 14%

lower complexity than the three-stage one recently presented in [13] with the same delay. Hardware complexity results were also provided for some practical implementations of MAP decoders based on UMTS-LTE [15], CCSDS [3], and Wi-MAX [16] turbo codes showing that the proposed n-input max* architecture features the lowest area among Log-MAP implementations.

REFERENCES

[1] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Commun.*, Jun. 1995, pp. 1009–1013.

[2] S. Jeon and B. V. K. V. Kumar, "Binary SOVA and nonbinary LDPC codes for turbo equalization in magnetic recording channels," *IEEE Trans. Magn.*, vol. 46, no. 6, pp. 2248–2251, Jun. 2010.

[3] (2002). *Consultative Committee for Space Data Systems (CCSDS): Telemetry Channel Coding* [Online]. Available: http://public.ccsds.org/publications/archive/101x0b6s.pdf

[4] W. J. Gross and P. G. Gulak, "Simplified MAP algorithm suitable for implementation of turbo decoders," *IET Electron. Lett.*, vol. 34, no. 16, pp. 1577–1578, Aug. 1998.

[5] J. Cheng and T. Ottosson, "Linearly approximated log-MAP algorithms for turbo decoding," in *Proc. IEEE Veh. Technol. Conf.*, May 2000, pp. 2252–2256.

[6] B. Classon, K. Blankenship, and V. Desai, "Channel coding for 4G systems with adaptive modulation and coding," *IEEE Wireless Commun. Mag.*, vol. 9, no. 2, pp. 8–13, Apr. 2002.

[7] S. Talakoub, L. Sabeti, B. Shahrrava, and M. Ahmadi, "An improved max-log-MAP algorithm for turbo decoding and turbo equalization," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 3, pp. 1058–1063, Jun. 2007.

[8] S. Papaharalabos, P. T. Mathiopoulos, G. Masera, and M. Martina, "On optimal and near-optimal turbo decoding using generalized max operator," *IEEE Commun. Lett.*, vol. 13, no. 7, pp. 522–524, Jul. 2009.

[9] Y. Tong, T. H. Yeap, and J. Y. Chouinard, "VHDL implementation of a turbo decoder with log-MAP-based iterative decoding," *IEEE Trans. Instrum. Meas.*, vol. 53, no. 4, pp. 1268–1278, Aug. 2004.

[10] X. Yin and J. Liu, "Design and implementation of an improved 3G turbo codes interleaver for 3GPP system," in *Proc. Int. Conf. Electron. Meas. Instrum.*, 2009, pp. 319–321.

[11] M. Martina, G. Masera, S. Papaharalabos, P. T. Mathiopoulos, and F. Gioulekas, "On practical implementation and generalizations of max operator for turbo and LDPC decoders," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 4, pp. 888–895, Apr. 2012.

[12] S. Papaharalabos, M. Sybis, P. Tyczka, and P. T. Mathiopoulos, "Modified log-MAP algorithm for simplified decoding of turbo and turbo TCM codes," in *Proc. IEEE Veh. Technol. Conf.*, Apr. 2009, pp. 1–5.

[13] K.-T. Shr, Y.-C. Chang, C.-Y. Lin, and Y.-H. Huang, "A 6.6 pJ/bit/iter radix-16 modified log-MAP decoder using two-stage ACS architecture," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Nov. 2011, pp. 313–316.

[14] S. Papaharalabos, P. T. Mathiopoulos, G. Masera, and M. Martina, "Novel non-recursive max operator with reduced implementation complexity for turbo decoding," *IET Commun.*, vol. 6, no. 7, pp. 702–707, Jul. 2012.

[15] *3GPP TS 36.212 v8.0.0: Multiplexing and Channel Coding*, Standard 2007-2009, 2009.

[16] *Air Interface for Fixed and Mobile Broadband Wireless Access Sytems: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands*, IEEE Standard P802.16e-2005, Feb. 2006.

[17] C. L. Wey, M. D. Shieh, and S. Y. Lin, "Algorithms of finding the first two minimum values and their hardware implementation," *IEEE Trans. Circuits Syst. I*, vol. 55, no. 11, pp. 3430–3437, Dec. 2008.

[18] L. G. Amarù, M. Martina, and G. Masera, "High speed architectures for finding the first two maximum/minimum values," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 12, pp. 2342–2346, Dec. 2012.

[19] C. Douillard and C. Berrou, "Turbo codes with rate-m/(m+1) constituent convolutional codes," *IEEE Trans. Commun.*, vol. 53, no. 10, pp. 1630–1638, Oct. 2005.

[20] J. Vogt and A. Finger, "Improving the max-log-MAP turbo decoder," *IEE Electron. Lett.*, vol. 36, no. 23, pp. 1937–1939, Nov. 2000.

[21] H. Claussen, H. R. Karimi, and B. Mulgrew, "Improved max-log-MAP turbo decoding by maximization of mutual information transfer," *EURASIP J. Appl. Signal Process.*, vol. 6, pp. 820–827, May 2005.

[22] A. P. Hekstra, "An alternative to metric rescaling in Viterbi decoders," *IEEE Trans. Commun.*, vol. 37, no. 11, pp. 1220–1222, Nov. 1989.

[23] G. Montorsi and S. Benedetto, "Design of fixed-point iterative decoders for concatenated codes with interleavers," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 5, pp. 871–882, May 2001.

[24] E. Boutillon, W. J. Gross, and P. G. Gulak, "VLSI architectures for the MAP algorithm," *IEEE Trans. Commun.*, vol. 51, no. 2, pp. 175–185, Feb. 2003.

Maurizio Martina (S'98–M'04) received the M.Sc. and Ph.D. degrees in electrical engineering from Politecnico di Torino, Torino, Italy, in 2000 and 2004, respectively.

He is currently an Assistant Professor with the VLSI Laboratory, Electronics and Telecommunications Department, Politecnico di Torino. His current research interests include VLSI design and implementation of architectures for digital signal processing and communications.

Stylianos Papaharalabos (M'04) received the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2001 and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Surrey, Guildford, U.K., in 2002 and 2006, respectively.

He has been a Post-Doctoral Research Fellow with the Institute for Astronomy Astrophysics Space Applications and Remote Sensing (IAASARS), National Observatory of Athens, Penteli, Greece, since 2006. He has participated in a number of European Information and Communication Technologies and European Space Agency research and development projects dealing with the physical layer of wireless telecommunication systems. He is the author or co-author of more than 38 journal and conference papers, including two book chapters. His current research interests include advanced channel coding schemes and iterative decoding techniques, joint source and channel coding, coded modulation, channel modeling, estimation algorithms, MIMO systems, and software/cognitive radios.

Dr. Papaharalabos is a member of the Technical Chamber of Greece. He has acted as a Technical Program Committee Member for several international conferences. For the International Conference on Advances in Satellite and Space Communications (SPACOMM) from 2010 to 2014, he has been an Advisory Chair Committee Member. He has been an Exemplary Reviewer for the IEEE Communications Letters in 2011 and an Editorial Board Member of the *IARIA International Journal on Advances in Telecommunications*.

P. Takis Mathiopoulos (S'79–M'84–SM'95) received the Ph.D. degree in digital communications from the University of Ottawa, Ottawa, ON, Canada, in 1989.

He was with Raytheon Canada Ltd., Waterloo, ON, Canada, from 1982 to 1986, on air navigational and satellite communications. In 1988, he joined the Department of Electrical and Computer Engineering, University of British Columbia (UBC), Vancouver, BC, Canada, where he was a Faculty Member in 2003. He is currently the Director of Research with the Institute for Astronomy Astrophysics Space Applications and Remote Sensing (IAASARS), National Observatory of Athens, Athens, Greece, where he established the Wireless Communications Research Group. As an ISARS' Director from 2000 to 2004, he has led the Institute to a significant expansion research and development growth, and international scientific recognition. For these achievements, ISARS has been selected as a national Centre of Excellence for the years 2005 to 2008. Since 2003, he has been teaches part-time with the Department of Informatics and Telecommunications, University of Athens, Athens, where he is an elected Professor of digital communications. Since 2008 he has been a Guest Professor with Southwest Jiaotong University, Chengdu, China. He has been conducting research mainly on the physical layer of digital communication systems for terrestrial and satellite applications, including digital communications over fading and interference environments. He co-authored a paper in GLOBECOM in 1989 establishing for the first time in the open technical literature the link between MLSE and multiple (or multisymbol) differential detection for the AWGN and fading channels. His current research interests include channel characterization and measurements, modulation and coding techniques, SIMO/MIMO, UWB, OFDM, software/cognitive radios, and green communications. He has research activities in the fields of remote sensing, LiDAR systems and photogrammetry. He has co-authored some 100 journal papers, mainly published in various IEEE and IET journals, four book chapters, and more than 120 conference papers. He has been PI for more than 40 research grants and has supervised the thesis of 11 Ph.D. and 23 Master students.

Dr. Mathiopoulos has been serving on the editorial board of several archival journals, including the *IET Communications* and the IEEE TRANSACTIONS ON COMMUNICATIONS from 1993 to 2005. He has regularly acted as a consultant for various governmental and private organizations. Since 1993, he has been served on a regular basis as a Scientific Advisor and a Technical Expert for the European Commission (EC). Since 2001 he has been served as the Greek representative to high level committees in the EC and the European Space Agency (ESA). He has been a member of the TPC of more than 50 international conferences, as well as TPC Vice Chair for the 2006-S IEEE VTC and 2008-F IEEE VTC as well as Co-Chair of FITCE in 2011. He has delivered numerous invited presentations, including plenary lectures, and has taught many short courses all over the world. He was an ASI Fellow, a Killam Research Fellow, and a co-recipient of two Best Paper Awards (2nd International Symposium on Communication, Control, and Signal Processing in 2008 and 3rd International Conference on Advances in Satellite and Space Communications in 2011).

Guido Masera (SM'07) received the Dr.Eng. degree (*summa cum laude*) and the Ph.D. degree in electrical engineering from Politecnico di Torino, Torino, Italy, in 1986 and 1992, respectively.

He was with Centro Studi e Laboratori in Telecomunicazioni, Torino, from 1986 to 1988, as a Researcher, involved in the standardization activities for the GSM system. Since 1992, he has been an Assistant Professor and then Associate Professor with the Electronic Department, where he is a member of the VLSI-Laboratory Group. In the frame of National and European research projects, he was a co-designer of several ASIC and FPGA implementations in the fields of artificial intelligence, computer networks, digital signal processing, and transmission and coding. He has co-authored more than 160 journal and conference papers in the areas of ASIC-SoC development, architectural synthesis, and VLSI circuit modeling and optimization. His current research interests include several aspects of the design of digital integrated circuits and systems, with special emphasis on high-performance architecture development (especially for wireless communications and multimedia applications) and on-chip interconnect modeling and optimization.