Architectural Comparison Model for Area-Efficient PMAP Turbo-Decoders

Original
Architectural Comparison Model for Area-Efficient PMAP Turbo-Decoders / Favero, S., Martina, M., Masera, G.. - In: IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. II, EXPRESS BRIEFS. - ISSN 1549-7747. - ELETTRONICO. - 70:1(2023), pp. 131-135. [10.1109/TCSII.2022.3207471]

Availability:
This version is available at: 11583/2971310 since: 2022-09-15T07:24:29Z

Publisher:
IEEE

Published
DOI:10.1109/TCSII.2022.3207471

(Article begins on next page)

Architectural Comparison Model for Area-Efficient PMAP Turbo-Decoders

Simone Favero, Maurizio Martina, *Senior Member, IEEE*, and Guido Masera, *Senior Member, IEEE*

***Abstract*—In this paper, a methodology to compare high-throughput turbo decoder architectures, is proposed. The model is based on the area-efficiency estimation of different architectures and design choices. Moreover, it is specifically oriented to the exploration of Parallel-MAP (PMAP) architectures, combined with both the Max-Log-MAP algorithm and the recently proposed Local-SOVA. The main objective is the search for optimal radix-orders, capable to maximize the area-efficiency of the decoder. In this scenario, it is proved that i) radix-orders higher than 4 are expected to drastically reduce the area-efficiency; ii) the optimal choice between radix-2 and radix-4 architectures strongly depends on the area distribution between logic and memory.**

***Index Terms*—Turbo-codes, Turbo-decoder, High-throughput, area-efficiency, Parallel-MAP, Max-Log-MAP, Local-SOVA.**

I. Introduction

DURING the last decades, the throughput demand in digital communication systems has been rapidly increasing. A projection of this trend in the following years is potentially leading to data-rates in the order of hundreds of Gbps, up to 1 Tbps [1]. Turbo Codes [2] represent a suitable solution for error correction, because of their capability to work close to the Shannon's limit. Since technology improvements are not expected to fully cover the next-generation requirements [3], the introduction of several parallelization degrees emerged as a viable solution to achieve high throughput turbo-decoders. For instance, multiple trellis sections can be simultaneously processed, by increasing the radix-order. Furthermore, one information frame can be split among different Soft-In-Soft-Out (SISO) processors and recent architectures handle multiple frames in parallel [4]. The extensive adoption of parallel architectures remarks the demand for area-efficient solutions, capable of maximizing the achievable throughput given a certain area budget.

This paper proposes a methodology to evaluate alternative design choices without going through the complete development of each architecture. A second purpose of this paper is to implement a detailed model to compare PMAP-Decoders [5], fixing a set of assumptions and tuning some parameters during the analysis.

The adoption of high radix-orders in Max-Log-MAP based decoders is often motivated with throughput benefits [6]–[8]; however, the effects of different radix-orders on complexity and area efficiency are rarely analyzed. A study on parallel XMAP architectures demonstrated how radix orders higher than 4 are inefficient, since the complexity overhead dominates the throughput benefits [4]. Therefore, an additional objective of this work is to verify if the same outcome can be extended to PMAP-based decoders. The presented results are also extended to architectures implementing the recently introduced Local-SOVA algorithm [9].

The paper is organized as follows. Section II details the methodology to derive the comparison model, where the fixed and variable parameters in the analysis are specified. Section III applies the model to explore both high radix-orders and Local-SOVA algorithm, in the context of PMAP-Decoders. Section IV concludes the paper.

II. Comparison Model for Area-Efficiency

A generic PMAP-Decoder block scheme is depicted in the left part of Fig. 1, where Π and Π^{-1} represent the permutation and inverse permutation, defined by the interleaver. It includes multiple SISO-processors, operating on different sections of the original information frame. The SISO-processor's internal organization is also reported in the Figure, assuming the Max-Log-MAP algorithm. The developed comparison model focuses on area efficiency estimation, $A_{eff} = Th/(A_L+A_M)$, where Th represents the throughput, A_L and A_M the area due to logic and memory, respectively.

First, the architectural space to be explored is identified and both logic and memory areas are estimated in terms of *Gate Equivalent* (GE). The logic part is analyzed by identifying the set of operators necessary for the execution of the algorithm. Then, area estimations are obtained through a synthesis campaign. The main memory contributions are identified by studying the storage requirements, the implementation models and the required access policies. Depending on the memory model, area can be studied either through a synthesis campaign or through a set of available library memory models. Lastly, the throughput is estimated by means of a specific mathematical model, depending on the architecture parameters. Concerning the working clock frequency, critical paths are identified, and the frequency degradation caused by different architectural choices is highlighted. The model's capability to select optimal designs is tested against potential errors, affecting the estimations of throughput, logic area and memory area.

A. Architectural Space

In order to limit the size of the explored design space, several choices are initially fixed. These *fixed parameters* in the analysis are:

- Convolutional codes declared in the LTE and UMTS standards;
- Max-Log-MAP algorithm;

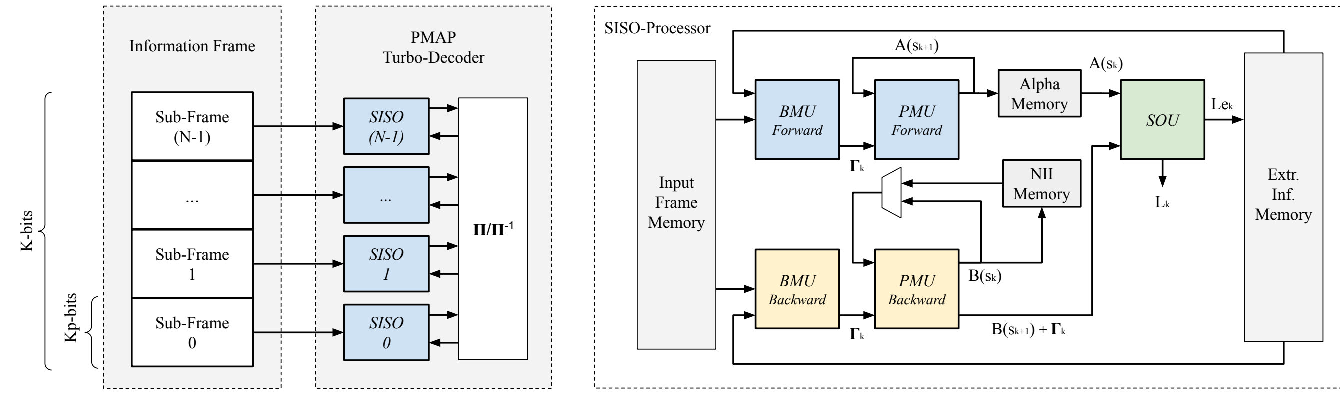

Fig. 1. PMAP-Decoder and SISO-processor block-schemes.

- Extrinsic Scaling Factor [10];
- Forward-Backward scheduling [11];
- Next-Iteration-Initialization [12].

It is is worth pointing out that the proposed model is not limited to the listed assumptions, which have been selected with the unique purpose of presenting one possible analysis. Indeed, the model can be easily extended, including, for instance, different scheduling policies. The following elements have instead been assumed as *variable parameters*, studied and regulated during the analysis:

- Frame-Size (K)
- Sub-Frame-Size (Kp)
- Window-Size (WS)
- Radix-order

More specifically, radix-2, 4, 8 and 16 have been analyzed. It has been observed that the considered variation ranges cause a maximum SNR loss of 0.3 dB at BER equal to 10^{-7} on an AWGN channel, where the main source of BER degradation is the adoption of different window-sizes.

B. Logic Area

The right part of Fig. 1 highlights the inside architecture of a SISO-processor, which includes three fundamental logic units: the *Branch Metric Unit* (BMU), the *Path Metric Unit* (PMU) and the *Soft-Output Unit* (SOU). Their role is respectively to compute the branch metrics Γ_k, the Alpha/Beta metrics $A(s_k), B(s_k)$ and the soft-output information L_k, as described in [2]. The mentioned logic units have been specifically optimized, in order to limit complexity and critical path. In the BMU, the number of required adders has been indeed minimized. The PMU critical path has been optimized by employing fast Compare-and-Select units and exploiting, when possible, parallel paths elimination [6]. Lastly, the minimum complexity SOU architecture, originally proposed in [13] for a radix-16 design, has been considered. Its guiding principle consists in minimizing the number of Compare-and-Select units, by avoiding redundant comparisons, and grouping together paths that share the same sequences of systematic bits. In this work, the same concept has been extended and adapted to radix-4 and radix-8 designs.

TABLE I
OPERATORS COUNT AND AREA ESTIMATION IN A SISO-PROCESSOR.

Radix	2	2	2	4	4	4
Unit	BMU	PMU	SOU	BMU	PMU	SOU
ADD/SUB	2/-	16/-	16/1	13/-	13/-	32/2
CS2/CS4	-/-	8/-	14/-	-/-	-/8	32/-

Radix	8	8	8	16	16	16
Unit	BMU	PMU	SOU	BMU	PMU	SOU
ADD/SUB	60/-	64/-	64/3	157/64	128/32	128/4
CS2/CS4	-/-	32/8	74/-	-/-	-/8	144/-

Logic	**Unit**	**ADD**	**SUB**	**CS2**	**CS4**
RCA-based	[um^2]	58.68	66.96	67.32	332.20
RCA-based	[GE]	41	47	47	224
CLA-based	[um^2]	159.74	182.28	183.26	904.32
CLA-based	[GE]	112	128	128	610

Then, four fundamental classes of operators are identified: adders (ADD), subtractors (SUB), 2-inputs compare-and-select (CS2), 4-inputs compare and select (CS4). Despite operands are expressed on a limited parallelism [12], Ripple-Carry-Adders (RCA) based logic and logarithmic Carry-Look-Ahead-Adders (CLA) based logic have been compared. Effects on delay and area are applied by following the models presented in [14]. Table I summarizes the operator count in a SISO-processor, as well as the area estimation results derived from a synthesis campaign on the 65 nm technology, assuming a parallelism of 8-bits. A proportionality between area and data parallelism has been proved for all the considered operators, therefore the reported area results can be easily converted to different bit-widths. Maximum relative variations around 10% are found when comparing the GE data with the other tested technological nodes, specifically 90 nm and 45 nm.

C. Memory Area

As depicted in the right part of Fig. 1, four memory components are identified in a PMAP-Decoder (gray blocks): the Input-Frame Memory, the Extrinsic-Information Memory, the Alpha Memory and the Next-Iteration-Initialization (NII) Memory. The first two memory components are globally shared among all the SISO-processors. On the contrary, the last two components are locally implemented in each SISO-processor. The storage requirements have been studied as a function of the considered variable parameters, producing the results reported in Table II, where $N = K/Kp$ represents the total number of SISO-processors, v is the memory length of the code and n is the inverse of the code rate. In the presented analysis, $v = 3$ and $n = 3$. The w parameters indicate the bit-widths for channel (ch), extrinsic (ext) and state metric (sm) quantities.

TABLE II
MEMORY STORAGE REQUIREMENTS.

Memory	N. Instances	Words	Bits/Word
Input Frame Mem.	1	K	$n \cdot w_{ch}$
Extr. Inf. Mem.	1	K	w_{ext}
Alpha Mem.	N	$WS/\log_2(radix)$	$2^v \cdot w_{sm}$
NII Mem.	N	Kp/WS	$2^v \cdot w_{sm}$

The introduced parallelization degrees generate simultaneous accesses to the memories. Therefore, avoiding access conflicts is of primary importance. The first step in this direction is to identify the maximum number of simultaneous accesses to each memory. In this scenario, increasing the radix-order produces a larger number of parallel accesses to global memories. Assuming the use of single-port memories, the reference technique to avoid conflicts is *memory partitioning*. The amount of required partitions has been derived from the parallel accesses analysis. Moreover, specific access policies have been identified to properly manage the access to the partitions. It is remarkable how the highest complexity access policy belongs to the Extrinsic-Information Memory, since conflicts are avoided during both natural and interleaved half-iterations. In this paper, a specific memory mapping technique is considered [15], extensible to any permutation law. A crossbar controls the interface between the processing units (SISO-processors) and the memory partitions. The addresses to drive the crossbar are selected by ensuring conformity to the permutation law and the absence of conflicts. In this work, additional memory banks are allocated, with the purpose of storing the permutation addresses. The crossbar complexity has been modeled by following the approach explained in Section II-B.

The memory complexity is better evaluated in terms of occupied area. However, to have a single complexity metric for both logic and memory, the GE/bit parameter has been introduced to indicate the number of GEs per stored information bit. Due to the presence of peripheral circuits in the memory architecture, the GE/bit parameter is expected to depend on the storage size. Therefore, a GE/bit variation model has been derived for both library and synthesized memories on the 65 nm technology. The latter are implemented when the requested number of words is below 32. As a result, library memories feature values lower than 3 GE/bit. Synthesized memories present instead values larger than 9 GE/bit.

D. Throughput and Latency

The generic throughput model for a PMAP architecture is $Th = K \cdot f/\Delta$, where $\Delta = (Kp + WS) \cdot n_{HI}/\log_2(radix)$ is the decoding latency, expressed in number of clock cycles. n_{HI} is the number of half-iterations and it has been fixed to 12; The working clock frequency f is set considering that the critical path is bounded by the PMU, due to the included feedback loop. The critical path increases with the radix-order, given the higher logic complexity. The cascades of logic operators on the critical paths have been identified, by estimating the delays through a synthesis campaign. The results are presented in Table III, where the last three columns report the expected increment factors of the critical path, computed considering the radix-2 architecture as a reference. In this regard, it is remarkable how critical paths are similarly affected by the radix order, even on different technologies.

TABLE III
CRITICAL PATH DATA.

	Technology					
	65 nm	90 nm	45 nm	65 nm	90 nm	45 nm
Radix	**Critical Path [ns]**			**Radix-2 CP Incr. Factor**		
2	1.46	0.91	1.37	1.00	1.00	1.00
4	1.99	1.26	1.80	1.36	1.38	1.31
8, 16	3.37	2.13	3.09	2.31	2.34	2.25

E. Power

A high-level dynamic power estimation is available given the definition of consumed power per unit of area and unit of frequency. This quantity, measured in [W/mm^2/MHz], presents similar values in different turbo-decoder designs mapped on the same technology [16]–[18]. Concerning the 65 nm node, values in the range 550-950 uW/mm^2/MHz have been found, with average 800 uW/mm^2/MHz. This parameter is customizable inside the model, to be tunable for different technologies and use cases. While its accuracy is limited, this power estimation technique provides a high-level estimate of the consequences of frequency and area variations in the design space explored.

III. EFFICIENT PMAP-DECODERS DESIGN

A. Max-Log-MAP Algorithm

The first step in searching for high-efficiency PMAP architectures is to identify the radix-orders that maximize the area efficiency. Table IV reports minimum and maximum area efficiency increment factors, extracted from the model's analysis and referred to the radix-2 architecture. The WS parameter has been fixed, while imposing a variation range on K and Kp. More specifically, K ranges from 2048 to 6144, while Kp from 128 to 512. According to Table IV, radix-8 and radix-16 designs present area increment factors larger than throughput ones, leading to a reduction of the area efficiency when compared against the radix-2 solution. Therefore, radix-orders higher than 4 have been discarded.

Concerning RCA based logic, the choice of the most efficient radix-order strongly depends on the Window-Size, which affects the Alpha Memory storage requirements, as indicated in Table II. This memory is larger for the radix-2 case than for the radix-4 one. Therefore, the model tends to select a radix-2 implementation when the Alpha-Memory can be efficiently mapped on a library memory, while the

TABLE IV
AREA EFFICIENCY INCREMENT FACTORS (RADIX-2 RELATED).

WS	32		64	
	RCA	CLA	RCA	CLA
Rad.	Min – Max	Min – Max	Min – Max	Min – Max
2	1.00 – 1.00	1.00 – 1.00	1.00 – 1.00	1.00 – 1.00
4	1.09 – 1.36	0.88 – 1.08	0.71 – 0.91	0.64 – 0.78
8	0.56 – 0.93	0.41 – 0.59	0.40 – 0.67	0.32 – 0.45
16	0.45 – 0.78	0.30 – 0.43	0.34 – 0.58	0.24 – 0.34

radix-4 version requires a flip-flop based synthesis. On the other hand, if both radix-2 and radix-4 designs allow the same Alpha-Memory technological implementation, the radix-4 architecture dominates the area efficiency. If CLA logic is selected, the logic area impact becomes larger. Therefore, the radix-2 architecture is always preferable in terms of better area efficiency. A comparison among different achievable area efficiencies, considering $K = 6144$ and $Kp = 384$, is depicted in Fig. 2. As visible, CLA logic adds a significant boost on the area efficiency. However, consequences on dynamic power consumption must be considered as well. The nominal power consumption tends to decrease by selecting the most efficient radix orders, between radix-2 and radix-4. As expected, CLA logic has a strong impact on the nominal power. Concerning power density, the latter tends to decrease with the radix-order, since lower frequencies and larger areas are involved. A degradation on the power density is found when switching from RCA logic to CLA logic: the frequency increment has a larger impact if compared to the logic complexity overhead. Finally, energy efficiency is expected to decrease as well with the radix-order, following the same considerations mentioned above. No important differences in energy efficiency are instead expected from switching between CLA logic and RCA logic.

The proposed model is able to explicit interesting details about logic and memory areas. For instance, considering $K = 6144$, $Kp = 256$, $WS = 32$ and a radix-4 approach, 19% of the total area is expected to be covered by logic, while 81% by memories, which confirms the fundamental impact of memories in the architecture. Furthermore, it can be appreciated how the PMU is the dominant logic element, while the Alpha Memory is the largest memory contribution.

Given the large impact of the Alpha Memory on the occupied area, re-computation represents an efficient solution to reduce complexity [19]. A subset of the Alpha metrics can indeed be re-computed during backward propagation, with a limited logic overhead and without affecting the BER. Practical results, involving the Max-Log-MAP algorithm, feature reduction factors on the Alpha Memory up to 80% [19]. The developed model includes the capability to consider the additional propagation logic and a percentage reduction on the Alpha Memory area, aiming to investigate the effect of re-computation. Reductions higher than 50% have a critical impact on the efficient radix distribution, which is progressively shifted toward radix-2 solutions.

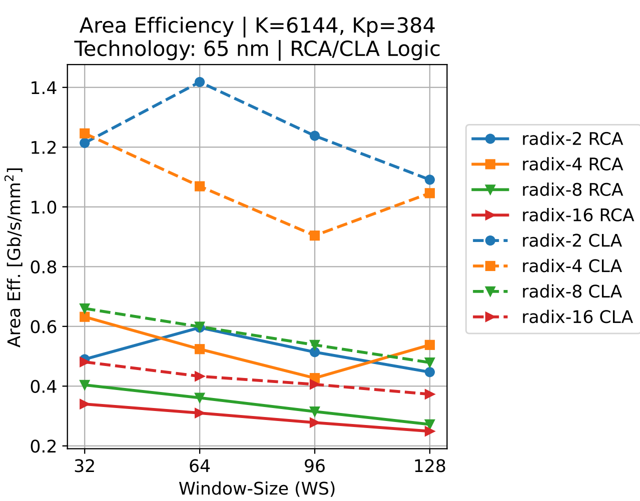

Fig. 2. Area efficiency variation as function of WS, for different radix-orders.

B. Model's Assessment

The developed model is able to verify the consistency of the selected radix-orders, by superimposing relative errors to estimated quantities and checking if the same outcomes are produced. Several tests in different scenarios have been performed, forcing maximum errors up to 10% on throughput, 20% on logic area and 20% on memory area. Based on the achieved results, the selected optimal radix-orders are not affected by the mentioned relative errors. Moreover, a synthesis campaign has been performed on several PMAP architectures, aiming to investigate potential errors affecting the parameters estimations. The synthesized architectures include datapath and memories, neglecting the control logic. The results in table V show how areas are typically underestimated by the model, with errors below 10%. As a matter of fact, the model does not consider part of the elements included in a complete architecture. However, *relative improvements in area efficiency*, as obtained by selecting specific designs, present errors below 4%, which proves the reliability of the methodology.

TABLE V
MODEL VS SYNTHESIS AREA ESTIMATIONS.

K = 6144, Kp = 256						
WS	Rad.	Mod. Area		Syn. Area		Error
		[mm^2]	GE	[mm^2]	[GE]	[%]
32	2	2.05	1.42M	2.20	1.53M	7.0
32	4	2.37	1.64M	2.60	1.81M	9.0
64	2	1.57	1.09M	1.74	1.21M	9.8
64	4	2.74	1.90M	2.96	2.06M	7.5

C. Local-SOVA algorithm

The Local-SOVA is a recently introduced algorithm, potentially achieving high-throughput in hardware implementations [9]. The Max-Log-MAP and the Local-SOVA perform the branch metrics computation and the state metrics propagation in the same way, but the soft-output information is computed differently. Indeed, the Local-SOVA combines paths by introducing a sequence of *merge* operations. Since the *merge* operator presents both the commutative and associative properties, paths can be merged in a dichotomous fashion, following a custom order. As a main advantage, selecting specific merging patterns against other ones reduces the computational complexity. The concepts of paths and merging can be extended to radix-orders higher than 2 [9]. Also in this case, properly organizing the merging order reduces the complexity.

A further complexity reduction is available, with an acceptable degradation on the error correction performances [9]. Indeed, the reliability update rules, defined in the *merge* operator, can be simplified. By gradually updating the *merge* operations, a trade-off is generated between complexity and BER. A complexity study reported in [9] for a radix-8 design predicts a maximum 49% logic complexity reduction against the Max-Log-MAP algorithm, featuring a performance loss of 0.05 dB, at a BER equal to 10^{-6}. The estimations take into account just the modified PMU and SOU. Place&Route results presented in [20] confirmed the possibility to reach complexity reductions up to 46%, by introducing new design solutions. If the low complexity SOU, employed in this work, is considered, the complexity benefit is expected to be reduced below 27%.

Since the Local-SOVA is specifically focused on the soft-output computation, the generic PMAP architecture, presented in Fig. 1, is still valid, as well as the timing for the execution of the algorithm and the memory organization. The unique differences are in the backward PMU and the SOU architectures, which are modified, with the purpose of applying the merging sequence. In order to include the Local-SOVA in the analysis, it is enough to introduce different models for the two mentioned units, considering the logic operators' data provided in [9]. The rest of the model remains unchanged. The most efficient radix-orders have been mapped again, exploring the same architectural space presented in Section III-A. As a result, the same radix distribution, reported for the Max-Log-MAP algorithm, has been generated. In conclusion, the logic complexity reduction achieved by the Local-SOVA is not expected to change the dominance of the radix-2 and radix-4 designs, in terms of area efficiency. Indeed, the logic reduction effect over the total area is limited by the presence of the remaining logic and memories.

IV. Conclusions

In this work, a generic methodology for the high-level complexity evaluation of PMAP high-throughput turbo-decoders is presented. The proposed model and the related tool [21] have been conceived to explore implementation trade-offs in area efficient decoders. The model was used to explore PMAP-based architectures, implementing both the Max-Log-MAP and the Local-SOVA algorithms. The achieved results show that radix-orders higher than 4 are not suitable solutions, since they feature a reduction of the area efficiency. The best choice between radix-2 and radix-4 architectures strongly depends on their effect on the memory organization. Additional results provided by the presented methodology are available in [21].

References

[1] "B5G Wireless Tb/s FEC KPI Requirement and Technology Gap Analysis," 2018. [Online]. Available: https://epic-h2020.eu/downloads/EPIC-D1.2-B5G-Wireless-Tbs-FEC-KPI-Requirement-and-Technology-Gap-Analysis-PU-M22.pdf

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proceedings of ICC '93 - IEEE International Conference on Communications*, vol. 2, 1993, pp. 1064–1070 vol.2.

[3] C. Kestel, M. Herrmann, and N. When, "When channel coding hits the implementation wall," in *2018 IEEE 10th International Symposium on Turbo Codes Iterative Information Processing (ISTC)*, 2018, pp. 1–6.

[4] S. Weithoffer, C. A. Nour, N. Wehn, C. Douillard, and C. Berrou, "25 Years of Turbo Codes: From Mb/s to beyond 100 Gb/s," in *2018 IEEE 10th International Symposium on Turbo Codes Iterative Information Processing (ISTC)*, 2018, pp. 1–6.

[5] M. Thul, F. Gilbert, T. Vogt, G. Kreiselmaier, and N. Wehn, "A scalable system architecture for high-throughput turbo-decoders," in *IEEE Workshop on Signal Processing Systems*, 2002, pp. 152–158.

[6] O. Sánchez, C. Jégo, M. Jézéquel, and Y. Saouter, "High speed low complexity radix-16 Max-Log-MAP SISO decoder," in *2012 19th IEEE International Conference on Electronics, Circuits, and Systems (ICECS 2012)*, 2012, pp. 400–403.

[7] P.-H. Chang, C.-Y. Lin, C.-H. Sun, Y.-C. Liao, and H.-C. Chang, "A 188-Length Full Code Rate 333Mbps 1.08mm^2 Radix-4 Hybrid-Trellis Turbo Decoder with Zero Patching for 3GPP LTE-A," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–4.

[8] C.-Y. Lin, C.-C. Wong, and H.-C. Chang, "An Area Efficient Radix-4 Reciprocal Dual Trellis Architecture for a High-Code-Rate Turbo Decoder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 1, pp. 65–69, 2015.

[9] V. H. S. Le, C. Abdel Nour, E. Boutillon, and C. Douillard, "Revisiting the Max-Log-Map Algorithm With SOVA Update Rules: New Simplifications for High-Radix SISO Decoders," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 1991–2004, 2020.

[10] J. Vogt and A. Finger, "Improving the max-Log-MAP turbo decoder," *Electronics Letters*, vol. 36, pp. 1937 – 1939, 12 2000.

[11] E. Boutillon, W. Gross, and P. Gulak, "VLSI architectures for the MAP algorithm," *IEEE Transactions on Communications*, vol. 51, no. 2, pp. 175–185, 2003.

[12] E. Boutillon, C. Douillard, and G. Montorsi, "Iterative Decoding of Concatenated Convolutional Codes: Implementation Issues," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1201–1227, 2007.

[13] O. D. Sanchez Gonzalez, "Towards higher speed decoding of convolutional turbocodes," Ph.D. dissertation, Télécom Bretagne, Université de Bretagne-Sud, 2013.

[14] B. Jovanovic and M. Jevtić, "Optimization of the binary adder architectures implemented in ASICs and FPGAs," vol. 195, pp. 295–308, 01 2013.

[15] A. Tarable and S. Benedetto, "Mapping interleaving laws to parallel turbo decoder architectures," *IEEE Communications Letters*, vol. 8, no. 3, pp. 162–164, 2004.

[16] X. Chen, Y. Chen, Y. Li, Y. Huang, and X. Zeng, "A 691 Mbps 1.392mm2 configurable radix-16 turbo decoder ASIC for 3GPP-LTE and WiMAX systems in 65nm CMOS," in *2013 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, 2013, pp. 157–160.

[17] S. Belfanti, C. Roth, M. Gautschi, C. Benkeser, and Q. Huang, "A 1Gbps LTE-advanced turbo-decoder ASIC in 65nm CMOS," in *2013 Symposium on VLSI Circuits*, 2013, pp. C284–C285.

[18] A. Li, L. Xiang, T. Chen, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "VLSI implementation of fully parallel LTE turbo decoders," *IEEE Access*, vol. 4, pp. 323–346, 2016.

[19] D. Spasov, M. Gushev, and S. Ristov, "Max-log-MAP decoding with reduced memory complexity," in *IEEE EUROCON 2015 - International Conference on Computer as a Tool (EUROCON)*, 2015, pp. 1–5.

[20] S. Weithoffer, R. Klaimi, C. A. Nour, N. Wehn, and C. Douillard, "Low-complexity Computational Units for the Local-SOVA Decoding Algorithm," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–6.

[21] S. Favero, "Area Efficiency Model for Turbo-Decoders," https://github.com/simomaiden/turbo_dec_aeff, 2021.