A 3.25Gb/s, 13.2pJ/b, 0.64mm^2 Configurable Successive-Cancellation List Polar Decoder using Split-Tree Architecture in 40nm CMOS

Yaoyu Tao, Sung-Gun Cho, Zhengya Zhang
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor

Abstract

A 0.64mm^2 configurable successive-cancellation list polar decoder is designed in 40nm CMOS for 5G wireless applications. The decoding tree is split to 4 subtrees to be decoded by 4 sub-decoders in parallel to improve throughput and cut latency by 4×. To maximize utilization, 8 frames are interleaved and decoded simultaneously to increase throughput by another 8× to 3.25Gb/s for code length up to 1024b. Dynamic clock gating reduces the peak power dissipation to 42.8mW at 0.9V, or 13.2pJ/b.

Introduction

Polar codes have recently been adopted for the 5G enhanced Mobile Broadband (eMBB) control channels. Using successive cancellation (SC) list decoding, polar codes offer the state-of-the-art error-correcting performance. However, sequential SC decoding incurs a low throughput, and list decoding requires costly tracking of a list of decisions. The latest 28nm SC list decoder demonstrated only 614Mb/s at 209pJ/b [1], presenting a challenge for 5G adoption. In this work, we divide a polar code's decoding tree to subtrees using a split-tree algorithm [2]. The subtrees are decoded in parallel by sub-decoders with decision reconciliation in every stage. The split-tree architecture improves both throughput and latency. We apply frame interleaving to further increase throughput, and dynamic clock gating to reduce energy.

SC List Decoding

An N-bit polar code is constructed by a systematic encoder, shown in Fig. 1(a). Decoded using SC, polar codes exhibit a polarization phenomenon: certain bits become highly reliable and the others become highly unreliable. Information is conveyed over the reliable bits and the unreliable bits are frozen to 0. SC decoding follows a sequential order: in each step, the log-likelihood ratio (LLR) is computed for a bit following the polar code trellis (Fig. 1(b)), and a hard decision is made.

SC decoding can be represented by the depth-traversal of an N-level binary tree. An SC decoder descends one level at a time, and the most likely path is selected. SC list decoding improves error-correcting performance by keeping a list of L ($L > 1$) most likely paths at each level to avoid premature decisions, as shown in Fig. 1(c). Moving from one level to the next, the L paths from the current level branch to $2L$ paths. The path metrics (PM) are computed, sorted and the top L paths are kept. A SC list decoder outputs L candidate paths upon completion. A cyclic redundancy check (CRC) can be used to assist the path selection. SC list decoding is slower and more complex than SC decoding as it requires sorting and tracking L paths.

Split-Tree SC List Decoder Architecture

To improve both latency and throughput, we create a new split-tree architecture based on the recent algorithm innovation [2]. Conceptually, the algorithm splits an N-level decoding tree to M subtrees of N/M levels, shown in Fig. 1(d), which is equivalent to splitting an N-bit code to M N/M-bit subcodes linked by a constraint matrix. The new split-tree decoder consists of M sub-decoders that operate on the subcodes in parallel, shortening latency and increasing throughput proportionally to M.

In practice, the split-tree decoder architecture requires an extra step to reconcile sub-decoders' local decisions at each level to meet the constraint between subcodes. To minimize the impact on performance and cost, we store the constraint in a lookup table as valid combinations of local decisions. The valid combinations are pre-computed based on a polar code's frozen bit selection. The size of the lookup table and the reconciliation overhead are quadratically dependent on M, limiting the practical splitting factor to $M = 4$.

A split-tree SC list decoder is shown in Fig. 2 for $L = 2$, $M = 4$, configurable code length up to $N = 1024$, and variable code rates. The decoder consists of 4 256b SC list sub-decoders. A sub-decoder traverses its binary subtree from the root. At each level, it operates on the subcode trellis to compute the LLR of a bit, and accumulates PM. The local decisions from the 4 sub-decoders are then reconciled in 3 steps: 1) valid combinations of the 4 bits are enumerated, and PMs of valid global paths are computed; 2) the valid global paths are sorted to find the top $L = 2$ candidate paths; 3) the top candidate paths are disassembled and distributed to the 4 sub-decoders. The 3 steps are done by the path metric calculator (PMC), sorter, and data structure updater (DSU), respectively, as shown in Fig. 2. The PMC employs a piecewise log approximation to simplify PM accumulation. The sorter utilizes a 4-stage binary sorter to pick the top global path, followed by parallel comparisons to locate the second top global path. The optimized 3-step reconciliation, including PMC, sorter and DSU, occupy only 0.021mm^2 in 40nm CMOS.

The 4 sub-decoders represent the majority of the decoder. The core of a 256b SC list sub-decoder performs SC decoding to compute the LLR of a bit by recursively passing through the 8-stage code trellis. As shown in Fig. 3(a), stage n of the trellis is mapped to 2^n PEs (each PE consists of an F (minimum) and a G (add) function as in Fig. 3(b)), with the necessary back substitution of decoded bits. A 256b sub-decoder requires 256 PEs organized in 8 stages. To efficiently support shorter code lengths, some trellis stages can be bypassed and clock-gated. The sub-decoder is pipelined for high throughput.

Frame Interleaving and Efficiency Enhancement

As in conventional SC list decoding, we note that the sub-decoder decodes only 1 bit in 1 to 8 clock cycles, with an average PE utilization of 1.57%. The naïve approach is to fold the 8-stage code trellis to 1 stage of 128 PEs, resulting in a higher, 3.14% PE utilization. However, the varying wiring patterns between stages introduce a 24% mux and control overhead, resulting in a longer clock period.

A better approach towards a higher efficiency is based on the observation that a pipelined SC decoder can accommodate multiple frames in the pipeline through interleaving. With frame interleaving, throughput is increased, but resource contention may occur as shown in Fig. 3(c). Resolving the contention requires multiple copies of decoder stages, as well as PMCs, sorters, controllers and dispatchers. The overhead is reflected in the increased area (Fig. 4). If we use area efficiency as the metric, i.e., throughput/area, an 8-frame-interleaved architecture is the optimal, as it increases the throughput by 7.8× and the area efficiency by 2.56× over the baseline as shown in Fig. 4. Dynamic clock gating is employed to reduce the power dissipation of the under-utilized hardware (Fig. 5).

Chip Implementation and Measurement Results

A test chip for the split-4, 8-frame-interleaved, configurable polar decoder for code length up to 1024b and variable code rates was fabricated in a 40nm CMOS process. The decoder core occupies 0.64mm^2. An on-chip CPU supports chip testing, I/O, and optional post-processing. The chip is verified to be fully functional. Key optimization steps are shown in Fig. 6 and the error-correcting performance is shown in Fig. 8.

At room temperature and a 0.9V nominal supply voltage, the chip is measured to achieve a 3.25Gb/s throughput at 430MHz, consuming 13.2pJ/b. The throughput, energy efficiency and area efficiency of this 40nm chip are 5.3×, 15.9× and 8.0× better, respectively, than the latest polar SC list decoder chip in a more advanced 28nm process [1] as shown in Fig. 7. Scaling the supply voltage to 450mV reduces the energy further to 8.21pJ/b. Compared to the belief propagation (BP) polar decoder [3], this decoder achieves 1.25dB coding gain at 10^{-5} FER (based on a rate-0.5 1024b code). The results also compare favorably to the latest 28nm LDPC decoder [4]. The chip microphoto is shown in Fig. 9.

Acknowledgements

This work was supported by Intel and NSF CCF-1054270. We thank Dr. F. Sheikh and collaborators at Intel Labs for advice.

References

[1] P. Giard, et al., *JETCAS*, 2017
[2] B. Li, et al., *arXiv*, 2013.
[3] Y.-T. Chen, et al., *VLSIC,* 2017.
[4] M. Weiner, et al., *ISSCC,* 2014.

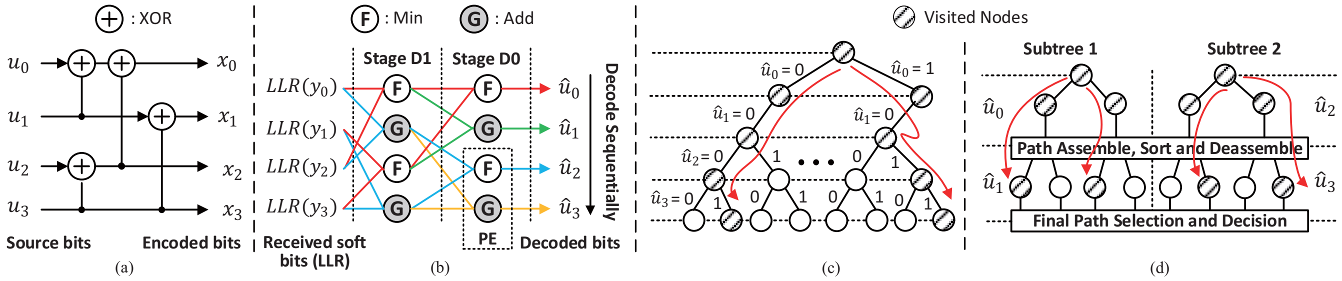
⊕ : XOR
u_0 u_1 u_2 u_3
x_0 x_1 x_2 x_3
Source bits
Encoded bits
(a)
F : Min
G : Add
Stage D1
Stage D0
$LLR(y_0)$ $LLR(y_1)$ $LLR(y_2)$ $LLR(y_3)$
$\hat{u}_0$ $\hat{u}_1$ $\hat{u}_2$ $\hat{u}_3$
Decode Sequentially
PE
Received soft bits (LLR)
Decoded bits
(b)
Visited Nodes
$\hat{u}_0 = 0$ $\hat{u}_0 = 1$ $\hat{u}_1 = 0$ $\hat{u}_1 = 0$ $\hat{u}_2 = 0$ $\hat{u}_3 = 0$
(c)
Subtree 1
Subtree 2
$\hat{u}_0$ $\hat{u}_1$ $\hat{u}_2$ $\hat{u}_3$
Path Assemble, Sort and Deassemble
Final Path Selection and Decision
(d)

Fig. 1. (a) Polar encoder; (b) polar code trellis (each color corresponds to a path to compute LLR of a bit); (c) 4-bit SC list decoding tree (*L*=2); (d) split-2, 4-bit SC list decoding tree (*L*=2).

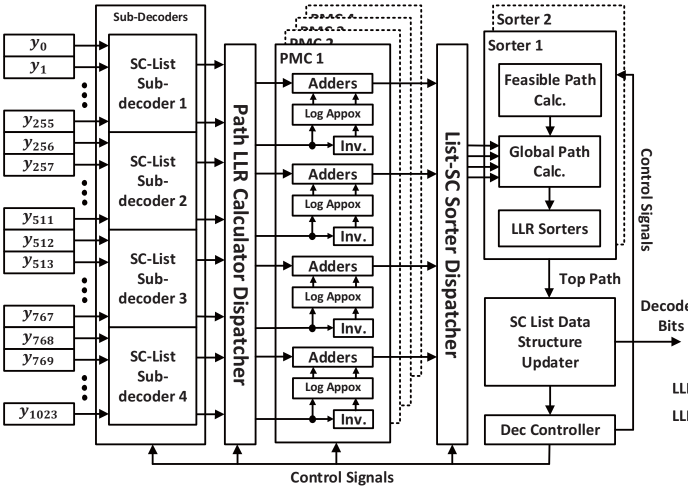

Fig. 2. Architecture of split-4 SC list decoder to support up to 1024b polar code and variable code rate (dotted blocks show the extra copies of PMC and sorter needed for 8-frame interleaving).

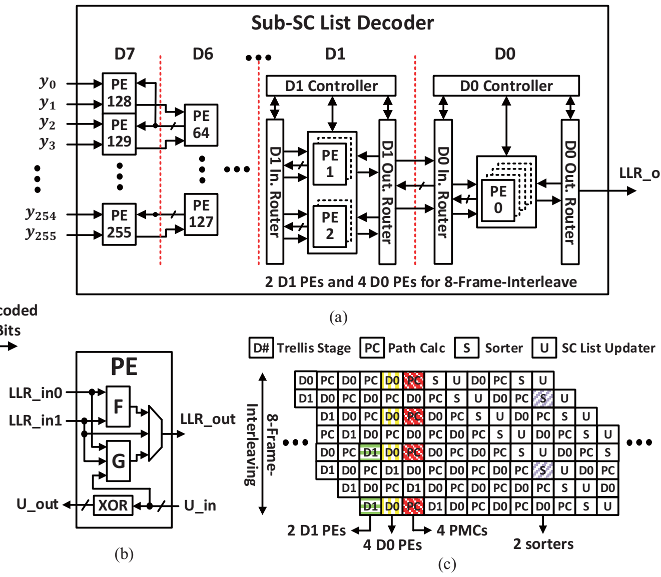

Fig. 3. (a) Design of a SC list sub-decoder to support 8-frame interleaving; (b) PE design; (c) pipeline chart for split-4 8-frame-interleaved decoding.

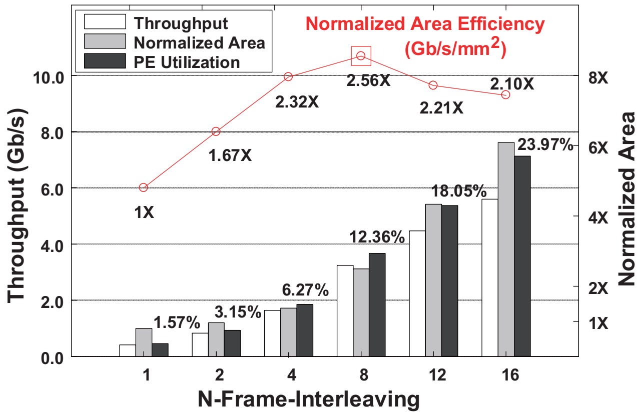

Fig. 4. Frame-interleaving architecture design exploration.

	This work		Giard JETCAS'17 [1]			Chen VLSI'17 [3]
Code Length	Variable length up to 1024		1024			1024
Code Rate	Variable rate		Variable rate			1/2
Algorithm	Split-tree SC List		SC List			BP 7.38 Iter.
List Size	2		4			N/A
SNR (dB) @ FER 1e-5	3.55 (8b-CRC)		3.20 (8b-CRC)			4.80
Quantization	6b (Q5.1)		12b (Q6.6)			5b
Process	40nm CMOS		28nm FD-SOI			40nm CMOS
Decoder Area (mm²)	0.637		0.44			0.704
Supply (V)	0.9	0.45	1.3	0.9	0.5	0.9
Frequency (MHz)	430	12.5	721	308	20	500
Throughput (Gb/s)	3.25	0.095	0.61	0.26	0.017	7.61
Power (mW)	42.80	0.78	128.3	23.3	0.6	422.7
Energy Effciency (pJ/b)	13.17	8.21	209	88.93	35.29	55.80
Area Effciency (Gb/s/mm²)	5.10	0.15	0.64	0.29	0.019	10.80

* Frequency, throughput and power comparisons are based on 1024b rate-0.5 polar code.

Fig. 7. Chip summary and comparison.

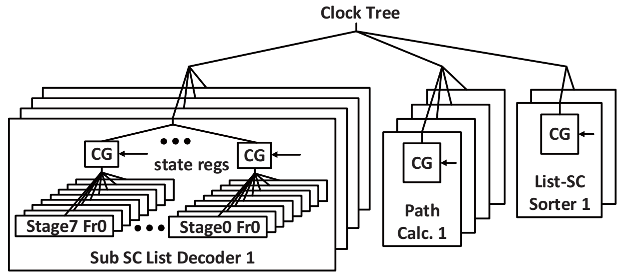

Fig. 5. Clock gating design of the split-4 8-frame-interleaved SC list decoder chip.

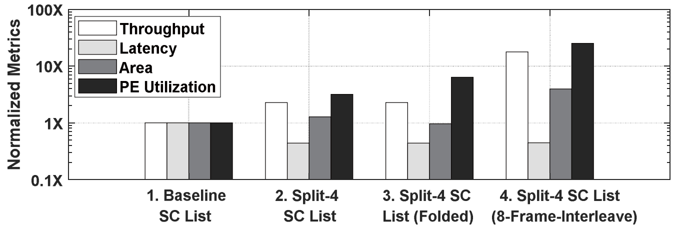

Fig. 6. Steps of chip design optimization.

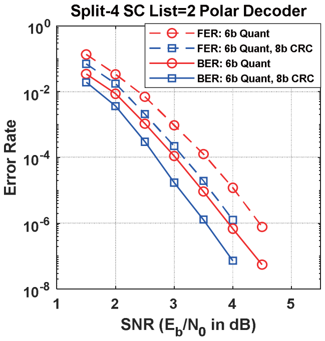

Fig. 8. Frame error rate (FER) and bit error rate (BER) performance of the chip design.

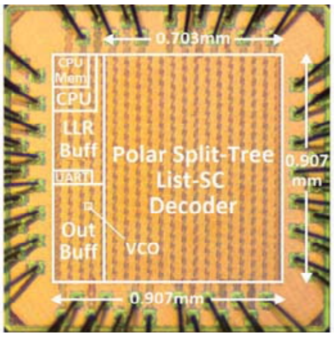

Fig. 9. Chip microphotograph.