A Fully Configurable Unified FEC Decoder for LDPC, Polar, Turbo, and Convolutional Codes with Row-First Collision-Free Compression

Yufan Yue[1], Seungkyu Choi[1], Tutu Ajayi[1], Xiangdong Wei[1], Ronald Dreslinski[1], David Blaauw[1], and Hun Seok Kim[1]

[1]University of Michigan

April 09, 2024

A Fully Configurable Unified FEC Decoder for LDPC, Polar, Turbo, and Convolutional Codes with Row-First Collision-Free Compression

Yufan Yue, Seungkyu Choi, Tutu Ajayi, Xiangdong Wei, Ronald Dreslinski, David Blaauw, Hun Seok Kim
University of Michigan, Ann Arbor, MI

***Abstract*—This paper presents the first chip implementation of a quad-mode decoder for LDPC, Polar, Turbo, and convolutional codes. It offers 9 fully configurable parameters accommodating arbitrary parity-check matrices up to the size of 8192×16384. It supports a broad range of wireless communication standards such as LTE, Wifi, WiMAX, WiGig, ITU, 5gNR, SDA-OCT, and proprietary codes. Through a novel Row-First Collision Free compression algorithm, LDPC memory usage is dramatically reduced by 89.4%. Furthermore, hardware-sharing techniques optimize memory requirements by an additional 36.9%. Operating at 93.08MHz in LDPC mode, the chip achieves a throughput of 1.62Gb/s/iteration at 239mW, with a normalized energy efficiency of 17.95fJ/bit/check/iteration. Its flexibility far exceeds state-of-the-art configurable and single-mode designs, positioning it as a front-runner in multi-mode decoding chips.**

I. Introduction

PROMINENT wireless communication standards increasingly utilize a wide range of forward error-correction (FEC) codes, including LDPC [1]–[5], [7], Polar [2], Turbo [4] and Convolutional Codes (CC). The rapid evolution of future standards necessitates flexible FEC decoders with multi-parameter and mode reconfigurability. Fast-changing non-idealistic channel conditions demand a wide range of coding gains and therefore call for a large variety of FEC codes. However, prior multi-mode designs [2], [4], [5], [7] can only support a small (≤2) set of parameters within 1 or 2 communication standards. They neither address all four codes, nor support a fully configurable LDPC parity check matrix (PCM). In particular, they lack CC support which is essential for 3G UMTS terrestrial communication and SDA-OCT inter-satellite communication. Significant challenges arise to directly map large LDPC PCMs in 5gNR or SDA-OCT onto prior architectures, as they do not harness sparsity effectively. Persistent issues related to memory collisions and the intricacies of interconnects have hindered the efficient realization of such multi-mode chips.

Addressing the above critical technology needs and technical challenges, we propose a novel FEC decoder that supports LDPC, Polar, Turbo, and CC under a unified architecture[1]. It is based on the critical observation that LDPC, Polar, Turbo, and CC decoding can be represented with a similar data-flow graph in Fig. 1. The unification of computational logics reduces the number of adders by 21.5%. In addition, we propose a new Row-First Collision-Free (RFCF) compression algorithm to eliminate memory collisions while optimizing memory utilization, culminating in an 89.4% memory savings for large PCMs. Our unified datapath efficiently resolves intricate routing challenges by introducing a flexible interconnect datapath that enables all routing patterns in quad-mode decoding. With multiple fully configurable parameters, it demonstrates unprecedented configurability for quad-mode support. In addition, these techniques exhibit normalized energy efficiency lower than prior reconfigurable designs and comparable to some state-of-the-art single-mode designs.

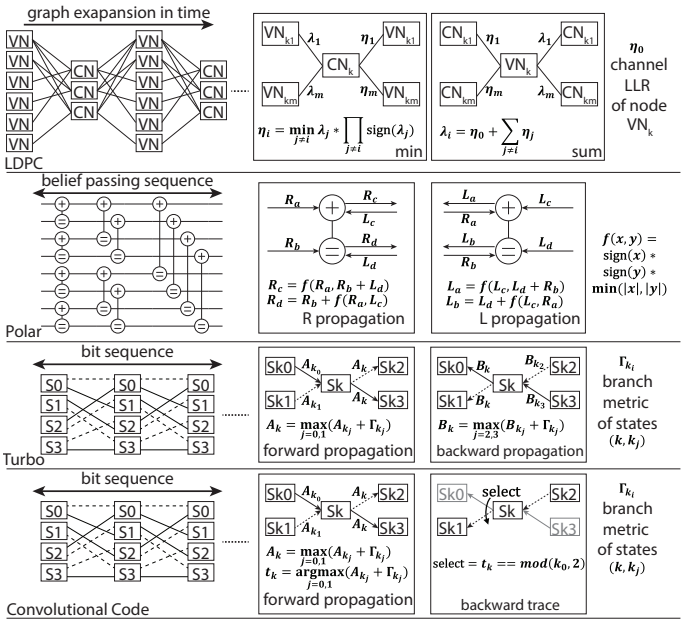

Fig. 1: Algorithm visualizations for quad-mode decoding

To design a unified decoder, we selected a proper set of algorithms that allows hardware resource sharing without compromising the decoding performance. As shown in Fig. 1, all LDPC, Polar, Turbo and CC decoding algorithms employ similar iterative *message-passing* structures using a dataflow graph that connects parallel computing units and memory units. The message-passing patterns are deterministic and can be programmed offline, therefore a flexible message-passing datapath can realize all aforementioned decoding algorithms. This allows a unified architecture that shares hardware resources for all codes without significantly degrading energy efficiency.

II. RFCF Compression Algorithm

Direct scaling of prior configurable designs incurs a quadratic growth with codelength of both memory bandwidth and storage requirements, rendering the previous approaches practically infeasible for large LDPC matrices such as 5gNR (2944×4352) and SDA-OCT (8192×16384). To tackle this challenge, our new RFCF compression fully harnesses the sparsity of PCMs to support arbitrary configurations without paying the quadratic complexity growth penalty for large PCMs. This offline algorithm maps non-empty proto-matrix entries only and skips zeros, resulting in a compressed representation of the PCM. As shown in Fig. 2 top-left, the initial mapping follows a row-first sequence across memory banks. Targeting collision-free bank accesses, any collision in the same proto-matrix column during the sum-half iteration is resolved by a swapping procedure (Fig. 2 top) which guarantees collision-free operations. After finding

[1]This work was in part published in ISLPED 2022 [6].

the desired mapping offline, the chip is programmed to use the identified deterministic, collision-free access patterns. Rigorous simulations involving 5gNR, Wifi, SDA-OCT, WiMAX, WiGig, WRAN, ITU, and custom PCMs confirm the absence of memory collisions with 33 memory banks. Furthermore, empirical simulations on PCMs show that collision-free mapping can always be found when the sum of proto-matrix row degree and column degree is less than the number of memory banks. This RFCF compression supports all arbitrary sparse PCMs and even some dense ones. This novel RFCF algorithm, combined with collision-free memory mappings for Polar and Turbo codes (Fig. 2), is unified seamlessly onto the shared RFCF Compression Datapath (Fig. 3).

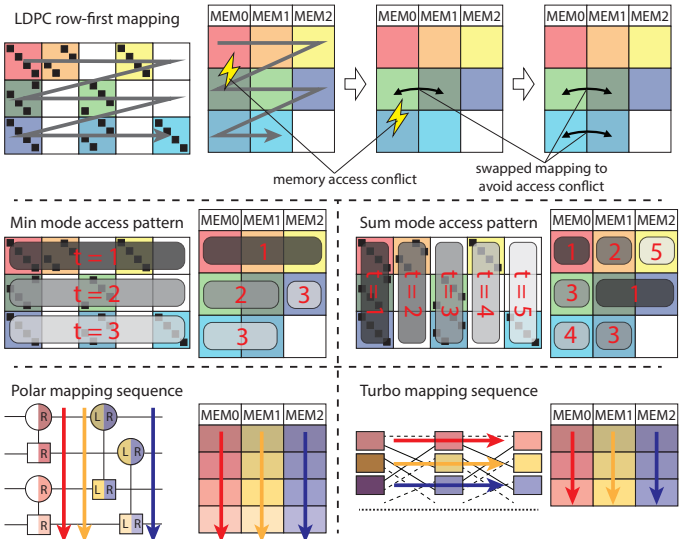

Fig. 2: RFCF compression visualization; memory mapping and access sequence for LDPC/Polar/Turbo mode

III. Hardware Architecture

Fig. 3 illustrates the top-level architecture. The design consists of four circuit groups: RFCF Compression Datapath, Flexible Interconnect Datapath, Flexible Computation Datapath, and IO Interface. The data follows a left-to-right (downstream) trajectory from Metric Memory to Reduction Tree, then right-to-left (upstream) to finish an iteration. Subsequent subsections will delve into the details of these circuit groups.

A. RFCF Compression Datapath

The RFCF Compression Datapath encompasses memory units and logic elements that facilitate the (de)compression of LDPC PCMs. It efficiently shares its operational logic with Turbo and Polar modes. Strategically, we have implemented 33 banks of 2.5KB Metric Memories (MM). These MMs store the compressed versions of belief propagation messages in LDPC or Polar, and path metrics in Turbo mode. Concurrently, the 33 banks of 60B Shift Memories (SM) store compressed shift values in the LDPC proto-matrices. Memory access patterns for PCM found by the RFCF algorithm are recorded in the 400B Count Mem and the 3.94KB Schedule Mem. To meet the demanding throughput requirements, MM operates with a bandwidth of 2112 messages/cycle for 4-bit LDPC messages. Additionally, a deterministic crossbar at the MM boundary facilitates the alteration of data flows between LDPC, Turbo, and Polar modes.

B. Flexible Interconnect Datapath

Overcoming the intricate data movement patterns inherent in arbitrarily configurable PCMs is crucial for designing efficient and flexible decoders. Our proposed design introduces an efficient Flexible Interconnect Datapath, addressing diverse data movement requirements. Comprising Static Layer 1/2 (SL1/2), Fully Connected Layer (FC), and Cyclic Shift Layer (CS), this architecture is shared between LDPC, Turbo, and Polar decoding. The Static Layers pipeline and reorder bits in messages, mitigating critical paths along millimeter-long wires in the large chip (23.76 mm^2). Given the parallelism and hardware complexity of FC and CS blocks, we adopt a hierarchical design to circumvent EDA tool limitations. In the FC layer, each direction features 32 groups of fully connected muxes, establishing full connectivity between MM banks and Processing Element (PE) units. Therefore satisfying all needs for arbitrary LDPC PCMs, as well as Polar and Turbo connection patterns (Fig. 4). The CS layer executes cyclic shifting on lifted PCMs based on SM-stored values, incorporating 36 instances of log shifters per direction, capable of arbitrarily configurable cyclic shifting within a length of 64 (Fig. 4). This CS layer also serves as a pipeline stage to boost clock frequency for Turbo/Polar codes.

C. Flexible Computation Datapath

As shown in Fig. 1, the four modes share significant algorithmic similarities. This critical observation enables efficient hardware sharing in computation datapaths. The Flexible Computation Datapath consists of a PE array with 36 PEs, and a Reduction Tree (RT) with 74 RT units. Detailed architecture and configurations of PE and RT units are shown in Fig. 4. Each PE house 64 flexible adders configured to perform either compare-and-select (CAS), subtraction (SUB), addition (ADD), or unsigned-min (UMN). The PEs can forward metrics in Turbo and CC modes via the deterministic crossbar to minimize memory access. Each PE is equipped with a 512B memory for temporary storage required by LDPC and CC decoding. Datapaths inside each PE are configured deterministically for each decoding mode. The RT units, instantiated in a 6-layer tree-like structure, compute minima, maxima, or sums. Each RT unit incorporates two flexible units to execute UMN, ADD, or MAX. They are configured according to Fig. 4 in different decoding modes.

D. IO Interface

Decoding completes after a programmed number of iterations, and the RT/PE generates soft output results. These soft decision results are converted to hard bits. These are then punctured based on the programmed parameter and re-ordered to produce the final output bit stream in the correct order.

E. Quad-mode operation

For LDPC, received soft decisions (LLRs) are initialized in In/Extrinsic Mem (IEM), while all belief propagation messages (η and λ in Fig. 1) are stored in MM. In the min/sum half-iterations, the messages of an entire PCM proto-matrix row/column are read from MM, and pass through the Flexible Interconnect Datapath. The message then goes to PE and RT. The results of RT broadcast back to PE. Combined with messages stored in the temporary memory, the RT generates

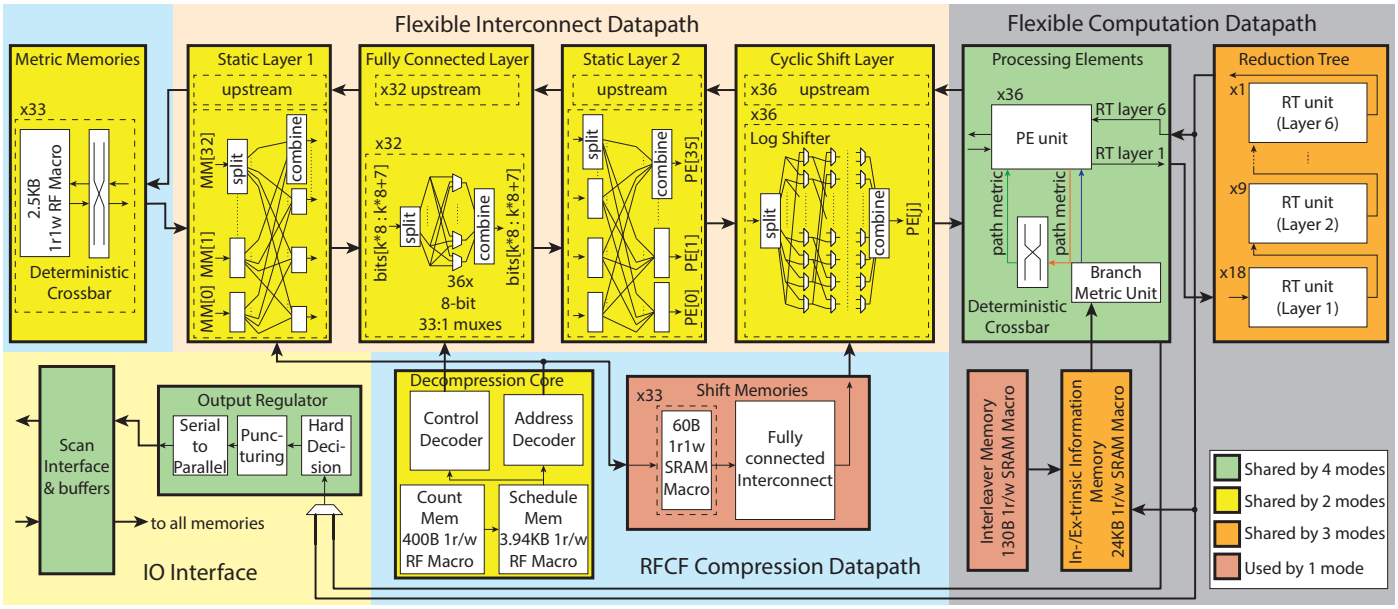

Fig. 3: Proposed top-level, detailed architecture, and hardware sharing status

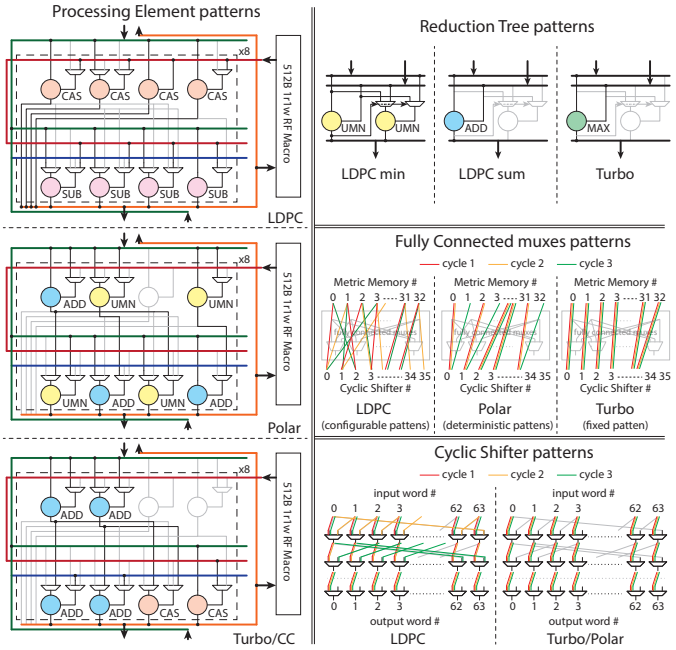

Fig. 4: Different configurations of PE/RT/FC/CS units

updated belief messages. They go through the Flexible Interconnect Datapath, and are ultimately stored in MM, completing a half-iteration.

In Polar mode, MM holds belief propagation messages L_x and R_x (Fig. 1). The messages pass through Flexible Interconnect Datapath. Then PEs update these metrics and write them back to MM for the next column.

In Turbo mode, received LLRs are initially stored in IEM. These values are accessed based on addresses stored in the interleaver memory. The LLRs are used in the calculation of branch metrics Γ_k. Using Γ_k, PEs calculate forward path metrics A_k (Fig. 1) and forward them to other PEs via the configured crossbar. Copies of A_k are routed through the Flexible Interconnect Datapath and are written to MM. During backward propagation, backward path metrics B_k and branch metrics Γ_k are computed using IEM entries. Combined with A_k from MM, these results go through additional processing in RT and are written back to IEM to update the extrinsic information.

In CC mode, PEs retrieve data from IEM to perform forward propagation, calculate path metrics A_k, and store its path selection t_k in the temporary memories. The path selections are sent to other PEs via the interconnect in backward propagation to find the most probable path of states to complete decoding.

IV. Analysis, Measurements and Results

The RFCF compression algorithm and configurable hardware demonstrate extensive flexibility, as detailed in Table 1.

TABLE I: Configurability comparisons with flexible designs

Parameters		Proposed	VLSI'10 [4]	SSCL'22 [2]	ISSCC'14 [7]	ISSCC'08 [5]
Code length	LDPC	**Arbitrary max 16384**	**Arbitrary max 2304**	Flexible max 6400	672	64800, 16200
	Turbo	**Arbitrary max 240**	**Arbitrary max 6144**	/	/	/
	Polar	**Arbitrary max 256**	/	**Arbitrary max 1024**	/	/
	CC	948	/	/	/	/
Code rate	LDPC	**Arbitrary**	1/2, 2/3 3/4, 5/6	10/52 to 10/14	1/2, 5/8, 3/4, 13/16	1/4 to 9/10
	Turbo	1/3	1/3	/	/	/
	Polar	**Arbitrary**	/	**Arbitrary**	/	/
	CC	1/6	/	/	/	/
LDPC lifting size (Z)		64	Flexible max 96	Flexible max 128	28	360
LDPC proto-matrix		**Arbitrary max 128x256**	Flexible max 12x24	Fixed	12x24	Flexible max 45x90
Turbo polynomials		**Arbitrary**	Fixed	/	/	/
Polar frozen location		**Arbitrary**	/	Flexible	/	/
CC polynomials		**Arbitrary**	/	/	/	/

Fixed: single value, exact value not mentioned in paper

This design represents a significant advancement as it is the first design for arbitrary sparse PCMs. Previous multi-mode designs [2], [4], [5], [7] were restricted to one or two modes with few configurable parameters and had significantly limited ranges of parameters. Overcoming technical challenges through innovative mapping, our design supports quad modes with 9 fully configurable parameters. It excels in configurability for longer LDPC codelengths up to 16384 bits and accommodates PCM proto-matrix sizes up to 128×256 with Z = 64 (PCM size ≤ 8192×16384). Furthermore, it allows for flexible

TABLE II: Performance comparisons with state-of-the-art designs

	Proposed				VLSI'10 [4]		SSCL'22 [2][a]		VLSI'23 [1]	ISSCC'15 [3]	ISSCC'08 [5]	ISSCC'14 [7]
Technology (nm)	**12**				65		40		28	65	65	28
Decoding Mode	**L**	**T**	**P**	**C**	L	T	L	P	L	L	L	L
Frequency (MHz)	**93.1**	**126**	**168**	**133**	320	320	180	150	813	517	174	260
Throughput (Mb/s/iter)	**1622**	**9**	**30.3**	**32.7**	640	140	2201	8140	34456	5120	135	45000
Power (mW)	**239**	**177**	**298**	**179**	675	570	63.3	169	2071[b]	103	350	180
Energy Effic. FoM (fJ/b/c/iter)	**17.95**				2746		23 ~ 1449 [c]		7.736	96.63	108.8	63.49
Max PCM size	**8192 × 16384**				1920 × 2304		5376 × 6656		6696 × 7440	208 × 416	32400 × 64800	346 × 672
Fully configurable parameters	**9**				2		2		0	0	0	0
Coding gain range (dB)	**1.9 ~ 30.3**				6.7 ~ 10.8		8.3 ~ 16.0		5.7	10.8	5.7 ~ 14.6	7.0 ~ 10.8
Supported Standards	**5gNR Wifi SDA-OCT WiMAX WiGig WRAN ITU LTE custom**				Wifi WiMAX LTE		5gNR		custom	custom	DVB-S2	WiGig

a: Unknown iteration; b: Based on data at 1.15V; c: Unknown LDPC lifting size;

Turbo/CC polynomials, constraint numbers, and full coverage of LDPC/Polar code rates and Polar frozen bit locations.

Fig. 3 specifies how each hardware component is shared among different codes, and Fig. 5 quantifies the gain (memory savings) from hardware sharing. Most power-intensive and area-consuming blocks are shared among 3 or 4 decoding modes, while smaller blocks are used in 1 or 2 decoding modes. Our design achieves notable savings in memory and logic, with RFCF compression leading to an impressive 89.4% reduction in memory footprint, consuming only an additional 1.3% of the total power. Furthermore, the consolidation of hardware results in an additional 36.9% memory saving. The incorporation of flexible computation units reduces full-adder count by 20.9% for PE and 30.2% for RT compared to separate designs.

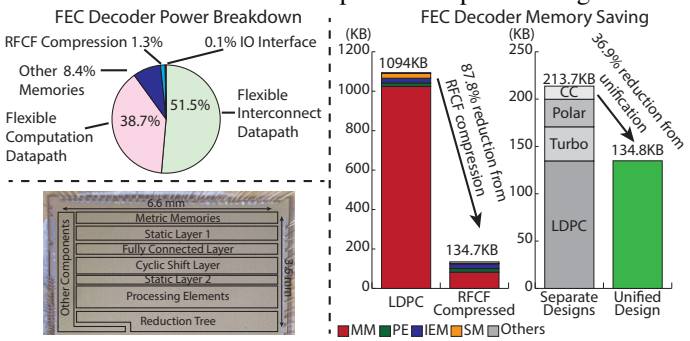

Fig. 5: Power breakdown, memory savings, and die photo

Figure 6 showcases the extensive LDPC configurability by offering a wide range of throughput-coding gain tradeoffs where the achievable coding gain is calculated using Gallager's coding gain bound. The flexibility to handle arbitrary PCM patterns empowers our decoder to adjust the coding gain to a channel-dependent optimal value for any valid PCM size. Our decoder surpasses previous designs by covering the entire range of coding gains achievable by a PCM matrix size of up to 8192×16384 (PCM proto-matrix size up to 128×256). Our decoder can be configured with an impressive 30.3dB coding gain with a large PCM matrix for challenging channel conditions. When the required coding gain is 10.8dB, it can deliver a throughput of 1.6Gb/s/iter.

The chip was fabricated with GF 12nm FinFET (Fig. 5), occupying an area of 23.76 mm^2. LDPC mode consumes 238.6mW at the throughput of 1.6224Gb/s/iter for an SDA-OCT mode with a PCM size of 8192x16384. In Polar, Turbo, and CC modes, it consumes 176.6mW at 9.00Mb/s/iter, 298.2mW at 30.26Mb/s/iter, and 178.6mW at 32.66Mb/s/iter, respectively. Energy breakdown (via simulations) shows that 51.5% is from interconnects and 38.7% is the PE/RT. The RFCF (de)compression overhead is only 1.3%. Note that

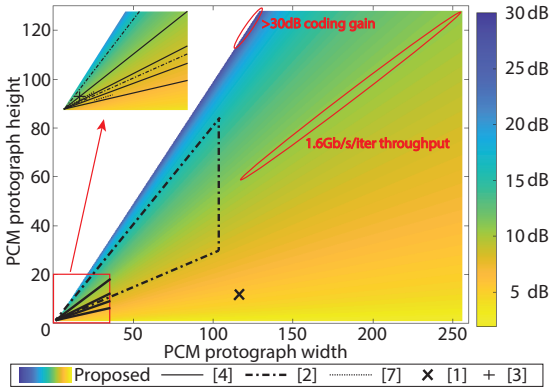

Fig. 6: Coding gain comparison over PCM proto-matrix size with state-of-the-art designs

decoding complexity grows quadraticaly with LDPC code-length (linearly with the size of the PCM) due to increased routing complexity (Fig. 5) and check node activity, a phenomenon also seen in previous work with a large PCM matrix [5]. Therefore, to facilitate meaningful/fair LDPC efficiency comparisons, we use as FoM the energy per bit per check per iteration (fJ/b/c/iter). Our quad-mode decoder achieves an FoM of 17.95fJ/b/c/iter at 0.8V which is superior than dual-mode designs [2], [4] and comparable with single-mode designs [1], [3], [5], [7] (Table 2). Our unconstrained configurability (Fig. 5) addresses an unprecedented range of standards (5gNR/Wifi/WiMAX/SDA-OCT/LTE/ITU etc.) with future-proofing.

V. Acknowledgments

This research was, in part, developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

[1] J. Choe and Y. Lee, "A 2.35 Gb/s/mm2 (7440, 6696) NB-LDPC Decoder over GF(32) using Memory-Reduced Column-Wise Trellis Min-Max Algorithm in 28nm CMOS Technology," VLSI 2023.

[2] B. -S. Su, C. -H. Lee and T. -D. Chiueh, "A 58.6/91.3 pJ/b Dual-Mode Belief-Propagation Decoder for LDPC and Polar Codes in the 5G Communications Standard," SSCL 2022.

[3] C. -H. Chen, W. Tang and Z. Zhang, "A 2.4mm2 130mW MMSE-nonbinary-LDPC iterative detector-decoder for 4×4 256-QAM MIMO in 65nm CMOS," ISSCC 2015.

[4] F. Naessens et al., "A 10.37 mm2 675mW reconfigurable LDPC and Turbo encoder and decoder for 802.11n, 802.16e and 3GPP-LTE," VLSI 2010.

[5] P. Urard et al., "A 360mW 105Mb/s DVB-S2 Compliant Codec based on 64800b LDPC and BCH Codes enabling Satellite-Transmission Portable Devices," ISSCC 2008.

[6] Yufan Yue et al., "A Unified Forward Error Correction Accelerator for Multi-Mode Turbo, LDPC, and Polar Decoding". ISLPED 2022.

[7] M. Weiner et al., "A scalable 1.5-to-6Gb/s 6.2-to-38.1mW LDPC decoder for 60GHz wireless networks in 28nm UTBB FDSOI," ISSCC 2014.