GPU-BASED IMPLEMENTATION OF BELIEF PROPAGATION DECODING FOR POLAR CODES

Zhanxian Liu[1], Rongke Liu[1], Zhiyuan Yan[2], Ling Zhao[1]
[1]School of Electrical and Information Engineering, Beihang University, Beijing, China
[2]Department of Electrical and Computer Engineering, Lehigh University, Pennsylvania, USA
zhy6@lehigh.edu, {liuzhanxian, rongke_liu, lingzhao}@buaa.edu.cn

Abstract

We propose an optimized software BP decoder for polar codes on graphics processing units (GPUs). A full-parallel decoding architecture for codes with length n ≤ 2048 is presented to simultaneously update n/2 processing elements (PEs) within each stage and achieve high on-chip memory utilization by using 8-bit quantization. And, for codes with length n > 2048, a partial-parallel decoding architecture is proposed to partly update PEs of each stage in parallel and coalesced global memory accesses are performed. Experimental results show that, with incorporation of the G-matrix based early termination criterion, more than 1 Gbps throughput for codes n ≤ 1024 can be achieved on NVIDIA TITAN Xp at 5 dB while the decoding latency is less than 1 ms. Compared with the state-of-the-art works, the proposed decoder achieves throughput speedups from 2.59x to 131x and provides good tradeoff between error performance and throughput.

Introduction

Belief Propagation (BP) decoding provides soft outputs. And, the BP decoding algorithm possesses intrinsic parallelism and can be efficiently implemented on the targets with highly parallel resources.

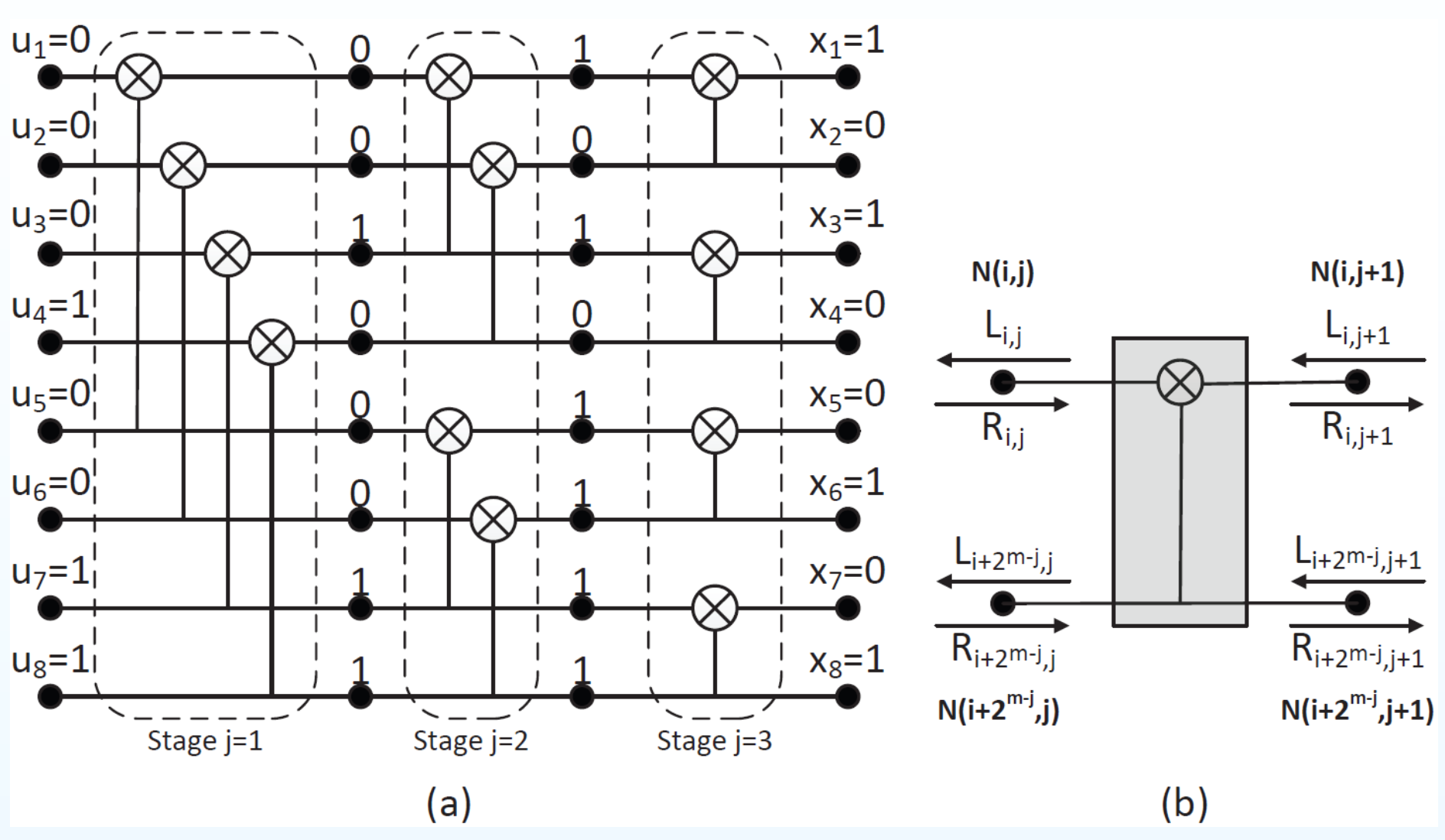
Contrary to the hardware BP decoders, this paper presents a software solution to meet the flexible and scalable requirements of the new generation communication systems such as Software Defined Radio and Virtualized Communication Systems.

The main contribution of this work is as follows: (1) Two effective mapping strategies based on code lengths are proposed to not only reduce the decoding latency but also attain high-level resource utilization; (2) Full shared and global memory efficiency is achieved by adopting 8-bit quantization. In addition, the asynchronous data transfer technique is utilized to solve the unbalanced workloads among the GPU's Streaming Multiprocessors (SMs) and hide the data transfer latency.

Compared with [2], at the same BER 10^{-5}, the throughput speedup reaches up to 70.9 times but at the cost of 0.9 dB coding gain loss, which indicates that the proposed decoder provides effective tradeoff between throughput and error performance.

BP decoding with G-matrix based early termination criterion

1. Encoding factor graph for an (8,4) polar code (a) and general PE (b)



2. Message updating within each PE

$$L_{i,j}^{(t)} = \alpha \cdot f(L_{i,j+1}^{(t)}, R_{i+2^{m-j},j}^{(t-1)} + L_{i+2^{m-j},j+1}^{(t)})$$
$$L_{i+2^{m-j},j}^{(t)} = \alpha \cdot f(L_{i,j+1}^{(t)}, R_{i,j}^{(t-1)}) + L_{i+2^{m-j},j+1}^{(t)}$$
$$R_{i,j+1}^{(t)} = \alpha \cdot f(R_{i,j}^{(t)}, R_{i+2^{m-j},j}^{(t)} + L_{i+2^{m-j},j+1}^{(t)})$$
$$R_{i+2^{m-j},j+1}^{(t)} = \alpha \cdot f(R_{i,j}^{(t)}, L_{i,j+1}^{(t)}) + R_{i+2^{m-j},j}^{(t)}$$

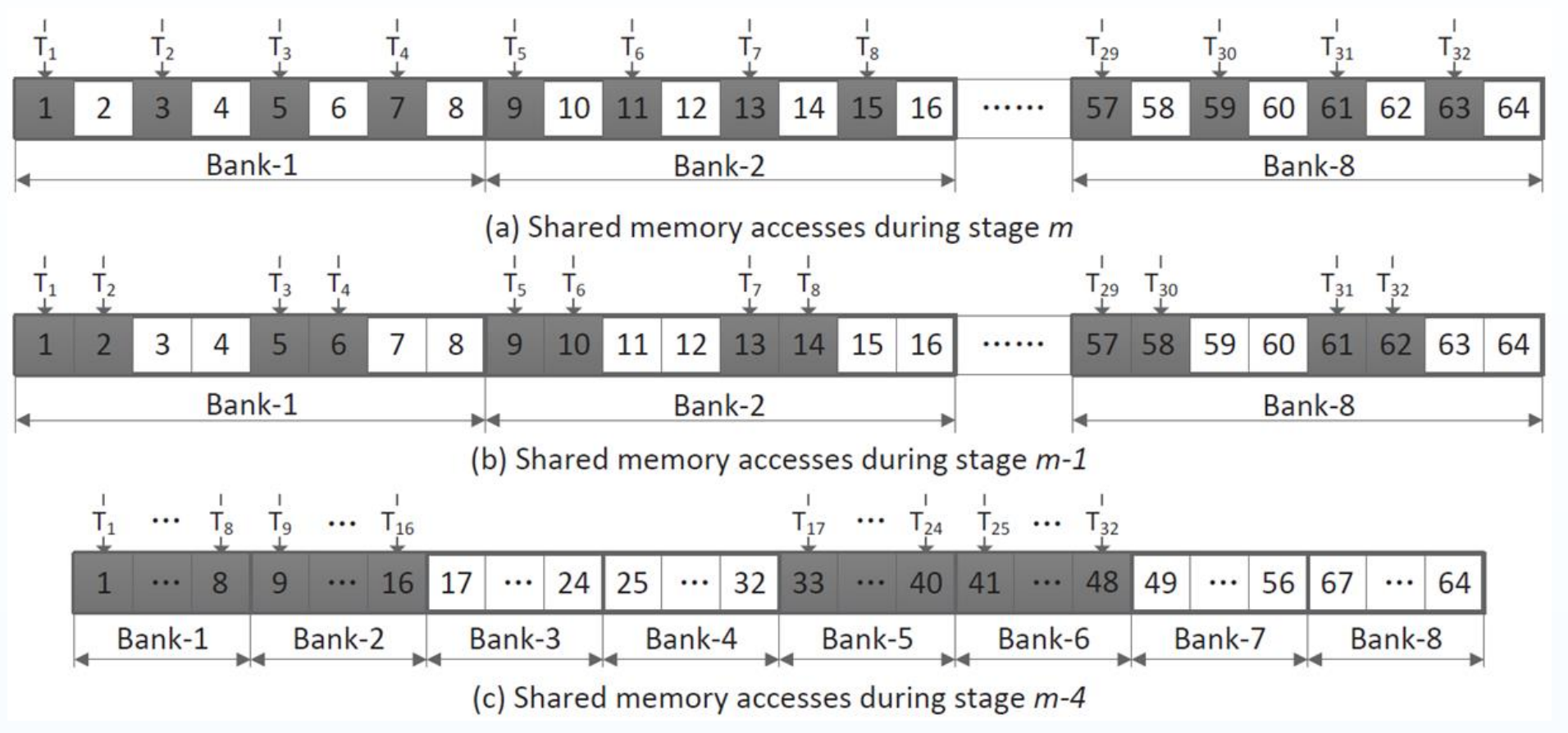
After each iteration, if $\hat{\boldsymbol{u}}\mathbf{G}=\hat{\boldsymbol{x}}$ or t reaches up to the maximum iteration number (I_{max}), the decoder will output $\hat{\boldsymbol{u}}$ as the decoding result.

Optimized GPU-Based BP decoder

1. Full-parallel BP (FP-BP) decoding architecture for codes n ≤ 2048

1) On-chip shared memory allocation and access

- Storing all $L_{i,j}^{(t)}$ and $R_{i,j}^{(t)}$ into shared memory
- 8-bit quantification
- Free bank conflicts (8-byte bandwidth granularity)



(a) Shared memory accesses during stage *m*

(b) Shared memory accesses during stage *m-1*

(c) Shared memory accesses during stage *m-4*

2) Mapping strategy of the threads

- Assigning one thread block with n/2 threads to one polar codeword and mapping the n/2 threads to n/2 PEs within the same stage
- Active blocks per SM: $N_{asm}= \left\lceil\frac{M_{sm}\times1024}{2n(m+1)}\right\rceil$, M_{sm}: shared memory per SM
- Registers per thread: $\left\lceil\frac{M_{rg}\times1024\times2}{N_{asm}\times n}\right\rceil$, M_{rg}: registers per SM (64KB in general)

2. Partial-parallel BP (PP-BP) decoding architecture for codes n > 2048

1) Global memory accesses

- Storing all $L_{i,j}^{(t)}$ and $R_{i,j}^{(t)}$ into off-chip global memory
- Coalesced memory access is achieved since the cache line size (128 bytes) is twice the area of data addresses (64 bytes) accessed by the 32 threads in a warp

2) Mapping strategy of the threads

- One thread block is assigned to one codeword but the number of threads per block is set to 1024
- Only 1024 PEs belonging to one stage are simultaneously processed and all Pes from the same stage are partially updated
- Maximum registers per thread are set to 32

3. Asynchronous data transfer

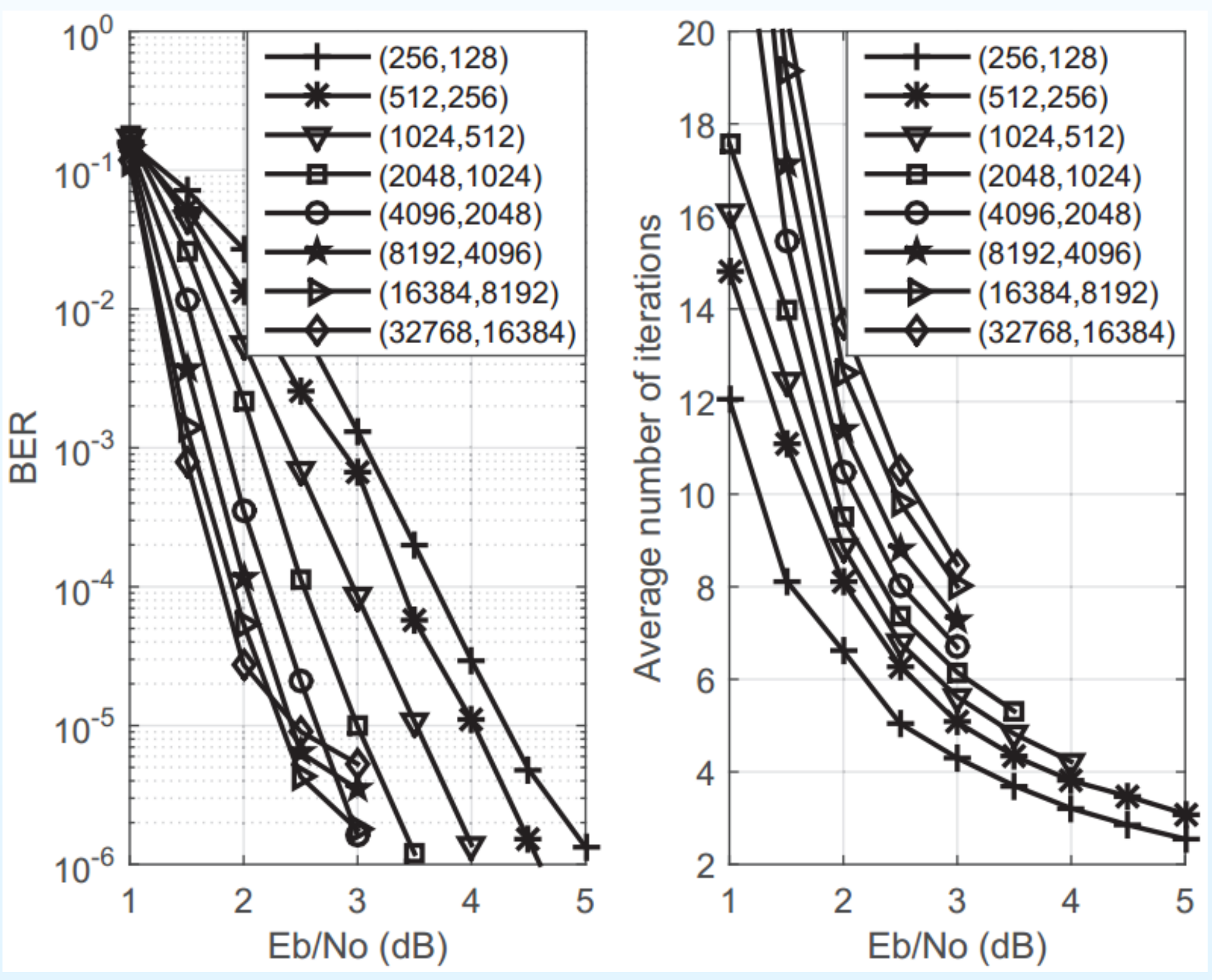
- Adopting the asynchronous data transfer mode to not only balance the SMs workloads but also hide the data transfer latency between host and device

The experimental results

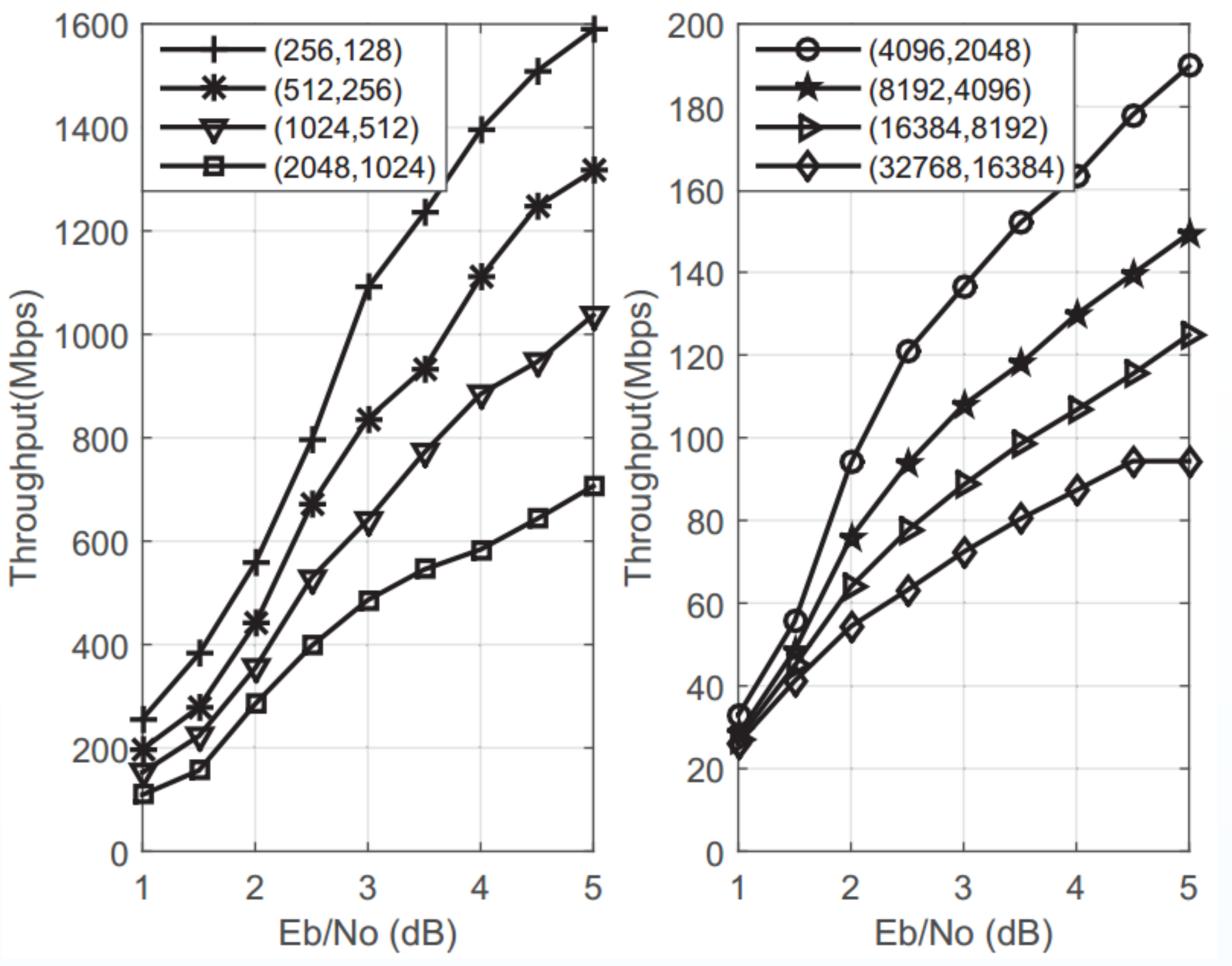
1. Platform setup

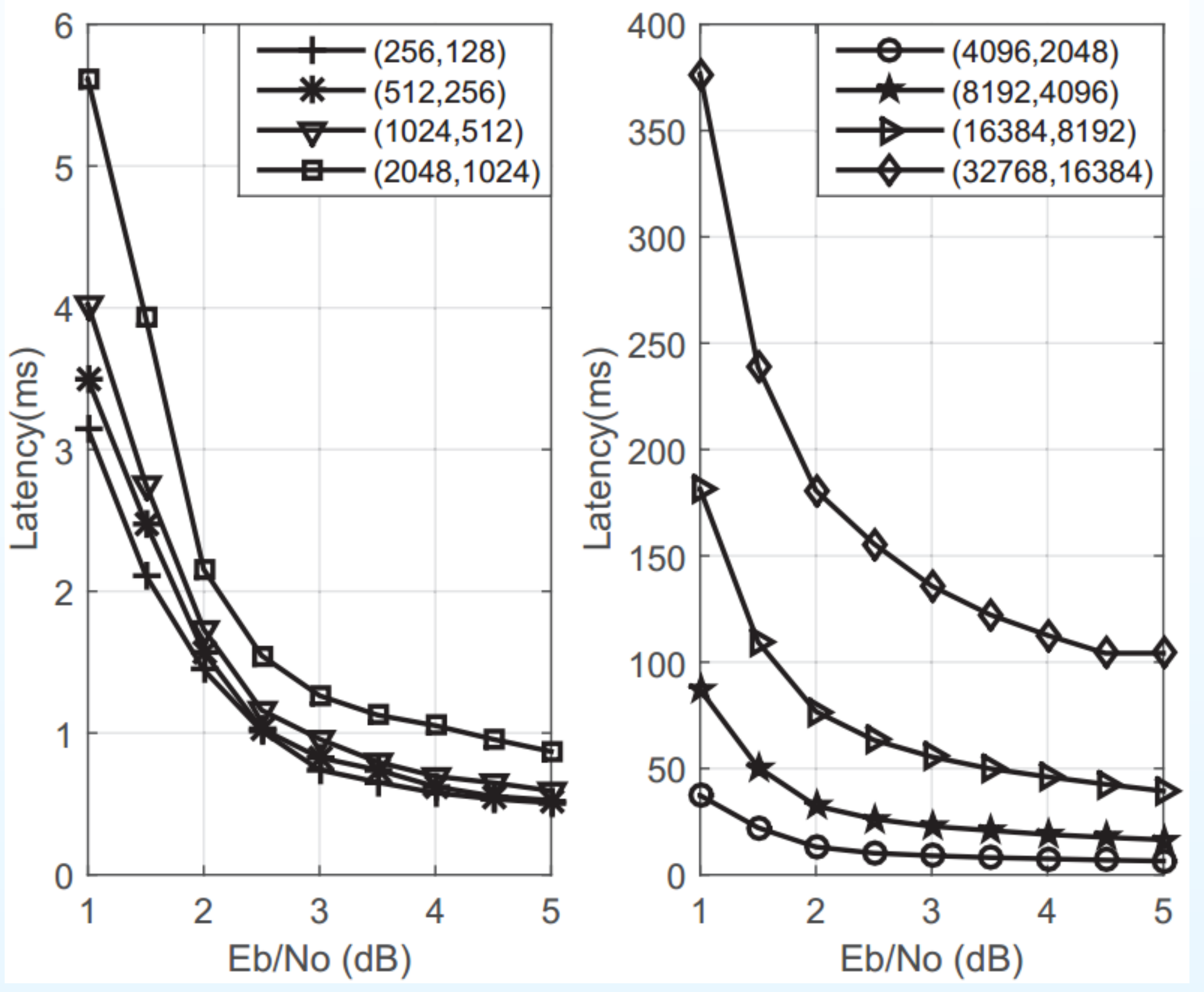
- Intel i7-8700K running at 3.7GHz and NVIDIA GTX TITAN Xp (Pascal architecture, 1405MHz, 30 SMs, 3849 cores, 12GB global memory)
- CUDA Toolkit 9.2 and Visual Studio 2013
- Windows 7 x64 system

2. BER performance and average number of iterations over different Eb/No values ($I_{max}=40$)



3. Decoding throughputs and latency over different Eb/No values ($I_{max}=40$)





For codes with length n ≤ 1024, the FP-BP decoder achieves more than 1 Gbps throughput and less than 1ms latency at 5 dB.

3. Throughput comparison with related works

N	$Eb/N0$ (dB)	Related works		Ours	Speed up
		Ref.	TNDC	TNDC	
256	4.0	[1]	1.837	38.0	20.7
512			0.911	31.2	34.2
1024			0.371	23.8	64.3
2048			0.129	16.9	131
4096	2.0	[2]	0.23	2.43	10.7
	3.0		1.045	3.44	3.29
	4.0		1.545	4.010	2.59
	5.0		1.545	4.415	2.86

Conclusion

This paper presents an FP-BP decoding architecture for codes n ≤ 2048 and a PP-BP decoding architecture for codes n > 2048 with efficient memory allocation and mapping strategies. To achieve high resource utilization, 8-bit quantization and the asynchronous data transfer mode are adopted. Compared with the related works, much higher throughput can be achieved, especially for short codes. For codes with length n ≤ 1024, the proposed TITAN Xp decoder achieves above 1 Gbps throughput and less than 1 ms latency by using the G-matrix based ETC. The presented GPU-based decoder can be used as a flexible channel decoding module in the new generation communication systems.

[1] B. K. Reddy L. and N. Chandrachoodan, "A GPU implementation of belief propagation decoder for polar codes," in Proc. Asilomar Conf. on Signals, Systems, and Computers, Nov. 2012, pp. 1272-1276.

[2] S. Cammerer, B. Leible, M. Stahl, J. Hoydis and S. T. Brink, "Combining belief propagation and successive cancellation list decoding of polar codes on a GPU platform," in Proc. IEEE Int. Conf. Acoust. Speech, Signal Process. (ICASSP), pp. 3664-3668, Mar. 2017.

Acknowledgements

The work was supported by the National Natural Science Foundation of China (61871009) and the US National Science Foundation under Grant ECCS-1509674.