I J C T A, 9(24), 2016, pp. 291-298 ISSN: 0974-5572

High Speed Downlink Packet Access efficient turbo decoder architecture: 3GPP Advanced Turbo Decoder

Parvathy M.*, Ganesan R.*, and Tefera Meckenon****

ABSTRACT

The multi-application specific instruction set processor (ASIP) architecture is a promising candidate for flexible high-throughput turbo decoders. This in brief proposes a network-on-chip (NoC) structure for multi-ASIP turbo decoders. The process of turbo decoding are studied and the addressing patterns for turbo codes in long term evolution (LTE) and High Speed Downlink Packet Access (HSDPA) is analyzed. Based on this analysis, the two techniques such as subnetworking and calculation sequence have been proposed for reducing the complexity of the NoC. The implementation result shows that the proposed structure provides an improvement of 53% for HSDPA and 133% for LTE in throughput/area efficiency compared with state-of-the-art NoC solutions. Multi-application specific instruction processor is combined with new decoding algorithm to increase the throughput.

Keywords: 3GPP,Turbo codes, SISO

1. INTRODUCTION

TURBO Codes have been mainly used in modern mobile communication standards such as 3G and 4G due to their remarkable error correcting capability. Turbo code was coded by C. Berrou et al. in 1993.Turbo decoders for the upcoming standard is not easy due to the nature of iterative turbo decoding. So only a few works have been demonstrated for LTE-Advanced turbo decoders. Soft-in Soft-out (SISO) decoding is the most used method for implementing a increased throughput turbo decoder. Many SISO decoders are used in Turbo, to split the code into sub blocks and process them in parallel.

When the decoding throughput can be increased by the number of SISO decoders included, the SISO architecture affects severely due to increased hardware complexity because a SISO decoder is one of the main component in a turbo decoder. This drawback is the main problem while implementing a Advanced turbo decoder, since the turbo decoder is required to attain a high decoding throughput which is greater than 1Gbps. For example, recently a turbo decoder [3] was demonstrated for LTE-Advanced standard which has 64 SISO decoders but such a complex system is not used for mobile applications.

The turbo decoder has two decoders operating parallel with the input data bits as shown in Figure.1.We can design either a 4-state or 8- state turbo decoder with different rate. Interleaver and De-interleaver are used between the convolution decoders. Coding rates are calculated by puncturing algorithm. When turbo encoded data is applied to the turbo decoder through Base-Band transmission over AWGN channel, where Log-Map decoding structure provides performance nearer to Shannon's limit with less complexity with respect to MAX-Log-MAP. Turbo codes are generally used in different communication system such as wireless communication, CDMA etc.

* Noorul Islam Centre for Higher Education, Kumaracoil, Thuckalay, Tamil Nadu, India, *Emails: meetparu.parvathy@gmail.com, dr.ganesh.jass@gmail.com*

** Jimma Institute of Technology, Jimma University, Ethiopia, *Email: mtefee@gmail.com*

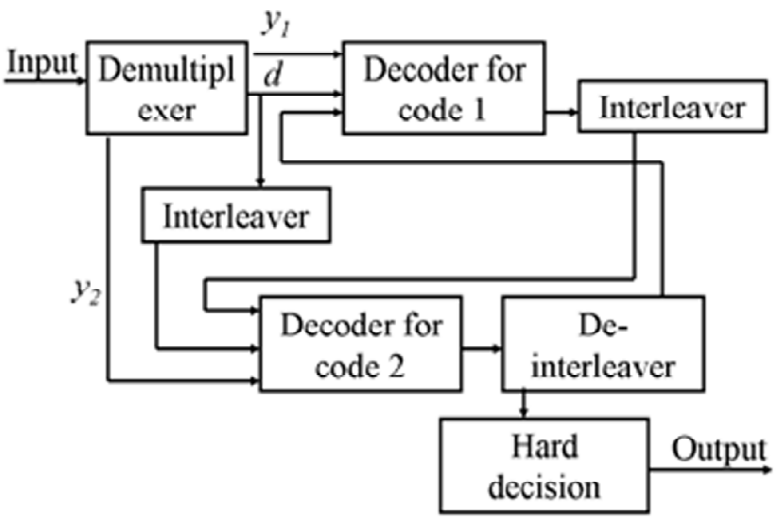


Figure 1: Turbo decoder

The conventional turbo decoder consist of two phases, in ordered phase and interleaved ones, alternatively by utilizing the extrinsic information obtained in the previous phase. To exchange the extrinsic information properly, each phase can start only when the previous phase is completed. Since each phase needs some setup time, the conventional decoder inevitably introduces undesired idle cycles, named as phase-switching latency (PSL), between the two phases. In order to relieve the extreme complexity of SISO-parallel decoders by eliminating the PSL, a new decoding algorithm called tail-overlapped decoding (TOD) is proposed in this paper.

Performance of extrinsic information memory (EIM) system is examined to support the TOD without incurring noticeable hardware overhead. By taking the advantage of the LTE-Advanced interleaving method [6], the proposed EIM system can be realized with single-port memories and exhibits much lower complexity than the conventional ones. Finally, a prototype LTE-Advanced turbo decoder is designed to realize the TOD in a configurable way. Either the conventional turbo de-coding mode or the TOD mode is selected according to the required throughput. In the TOD mode, the maximum decoding throughput is enhanced by 25% without degrading the error-correcting performance noticeably. As enhancing the maximum throughput it enables us to reduce the number of parallel SISO decoders. The proposed TOD is effective in mitigating the hardware complexity.

2. TAIL-OVERLAPPED DECODING METHOD

The MAP decoding is also known as BCJR [3] algorithm is not either practical algorithm for implementation in real time systems. Maximum a Posterior algorithm is computationally complex and sensitive to SNR mismatch and error in estimation of the noise variance [4]. This algorithm requires non-linear functions for computation of the probabilities where both multiplication and addition are also needed to compute the variables of this algorithm. The fixed-point representation of the MAP decoding variables usually appears between 16 to 24 bits for a QPSK signal constellation. Based on the above hardware requirements, the MAP algorithm is not practical to implement in a chip.

The logarithmic version of MAP algorithm [5-7] and the Soft Output Viterbi Algorithm (SOVA) [8-9] are the practical decoding algorithms for the implementation. These algorithms are very less sensitive to SNR mismatch and inaccurate estimation of the noise variance and their fixed-point representation of their variables require approximately 8bits for a QPSK signal constellation. SOVA has the least computational complexity and contains the worse bit error rate (BER) performance. Among these algorithms, the Log MAP algorithm [5] has the best BER performance equivalent on the MAP algorithm and the highest computational complexity.

Extrinsic information memory is large and continuously accessed at the time of decoding so it can be tough in real time implementations. For a double binary decoder [14], extrinsic information is transferred to bit-level information to reduce the EIM size.

Phase Switching Latency is longer than a sliding window appears inevitably between adjacent phases, and it can be relatively long in high-throughput decoders. Note that the setup time is dependent on the SISO architecture, and can be reduced to a half window in the butterfly SISO architecture [15] that calculates the backward and forward state metrics of a window concurrently. The conventional architecture [3]–[5] is assumed in this work, since the butterfly SISO architecture doubles the memory bandwidth.

SISO decoder operates under the sliding window method which is shown in Fig.2. The interleaved phase is denoted with thick border lines to make it different from the in-ordered one. The inputs are channel LLRs and a priori extrinsic LLRs, and the outputs are updated extrinsic LLRs. Since the two recursions proceed in opposite directions, the backward recursion and the calculation of output LLRs for the first sliding window should start after the forward recursion of the same window is completed. In the figure, the PSL is at least as long as the window size, and it can be represented in terms of clock cycles as

$$PSL = T + \Delta \tag{1}$$

Where T is the number of clock cycles taken to process a single window and Δ is the pipeline delay that is dependent on the SISO decoder architecture.

Pure decoding Time (PDT) as the total number of decoding cycles excluding the PSL. Pure decoding time is given as,

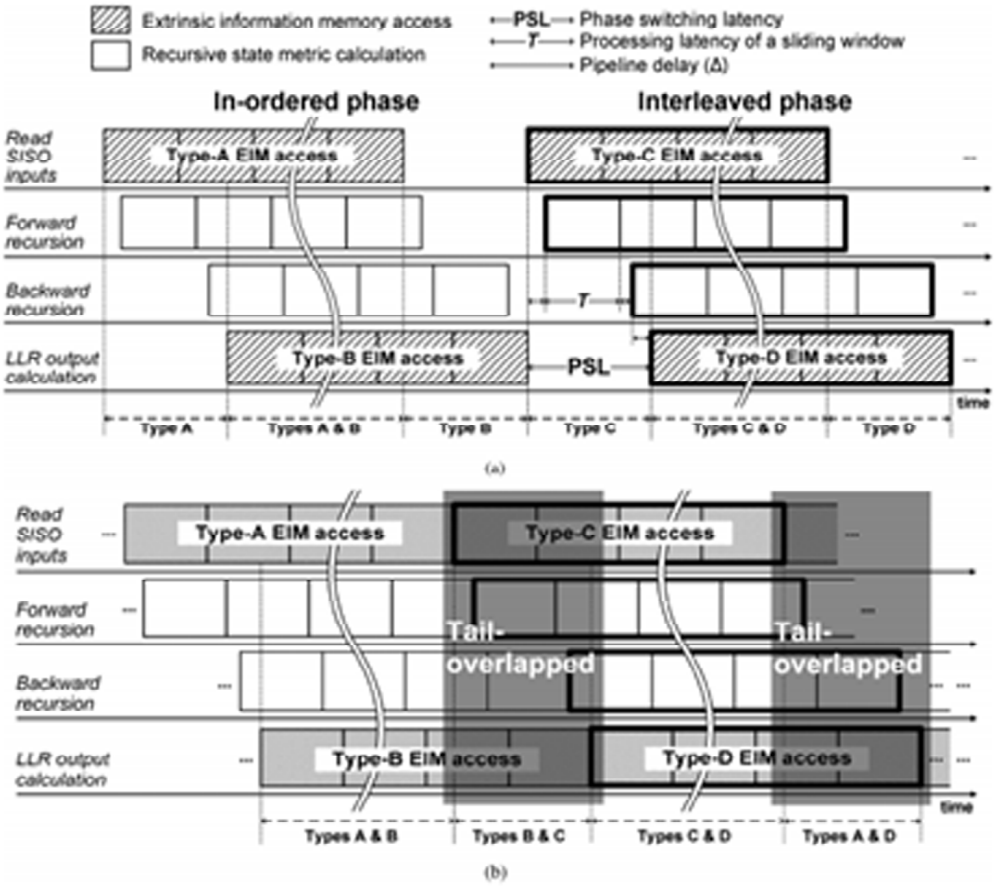


Figure 2: The operation flow of a SISO decoder in (a) the conventional (normal) mode and (b) the proposed tail-overlapped decoding mode under the sliding window method. Access types to the extrinsic information memory are denoted at the bottom.

$$PDT=N/P_v \quad (2)$$

Where N is the code block size, P is the number of parallel SISO decoders and v is the number of output LLRs calculated per cycle. If a SISO decoder is designed based on the radix 2^v architecture, where *v* bits are decoded per cycle. Using the parameters, PSL equation can be rewritten as

$$PSL = L + \Delta v \quad (3)$$

where L is the window size. Finally, the PSL overhead is expressed as

$$\frac{PSL}{PDT} = \frac{P(L+\Delta v)}{N} \quad (4)$$

Tail-over-lapped decoding (TOD) is introduced to maximize the decoding throughput by removing the PSL completely.

3. TURBO DECODING IN MULTI-ASIP ARCHITECTURE

In our multi-ASIP architecture, a code frame with N bits of information is split into P non-overlapped windows. Each window is processed by one ASIP. These ASIPs generate the memory addresses, which implement radix-4 decoding algorithms [5], and exchange log-likelihood ratio (LLRs) and Log-likelihood ratio of extrinsic information (LEs) (extrinsic information) through the NoC and memories. Since the decoding process is programmed in the ASIPs, and the NoC provides full connectivity between ASIPs and memories, this kind of architecture can satisfy the flexibility requirement of various turbo codes.

There are four main components: an input memory, an extrinsic information memory (EIM), 16 SISO de-coders working in parallel, and a main controller. A decoding phase or half iteration consists of loading SISO inputs, performing parallel SISO decoding and storing the output LLRs. The channel output LLRs and a priori LLRs, which are used as input values are read from the input memory and the EIM respectively. A channel output LLR consists of three LLR values corresponding to a systematic bit and two parity bits. On the other hand, the output LLRs resulting from the SISO decoders are stored into the EIM to be used as a priori probabilities in the following phase. Note that up to 16 channel output LLRs can be stored in a row of the input memory, because the interleaver specified in [2] is contention-free and vectorizable if the number of parallel SISO decoders is a factor of the code length [18]. This is also true for the extrinsic LLRs. In addition, the number of SISO decoders will be activated which is referred as the parallel factor and controlled based on the code length.

To avoid performance degradation caused by severe fractioning, at least two sliding windows are allocated to each SISO decoder for all the code lengths specified in [2]. For example, only 8 SISO decoders are participated in decoding a 1024-bit code, while 16 decoders are fully used for decoding a code whose length is 2048 bits or larger.

Table 1
Characteristics of The Proposed Turbo Decoder

Parameter	*Value*
Maximum Code Length (*N*)	6144 bits
#Parallel SISO decoders (*P*)	16
Sliding window size (*L*)	64 bits
#Output bits per cycle (*v*)	2 bits
Miscellaneous delay (Δ)	16 cycles

SOVA algorithm consists of two stages: (a) Calculation of weights of the ML path and (b) Updating of those weights in order to obtain a-posteriori LLR values. The first step involves the calculation of the path metric (PM). Then the trace back (TB) process is carried out inside the decoding window whose length, W_d guarantees the convergence of the surviving paths into the ML path. Once this path is known, the decoded bit sequence is obtained. In order to associate the reliability measure to each decoded bit, the importance of each decision (weight) is performed. When the PMs and the weights for the decoding window W_d have been calculated, the TB is carried out in order to obtain the weights of the ML path Sk. In the second step, the weights are updated as explained in detail in [1] using updating window of length W_u. The initialization implies $PM_0(s) = \log(\delta(s))$ and $S_{K+T} = 0$.

The contention-free EIM controller realizing the above strategy is presented in Fig. 4. Note that signal A indicates whether the type of the current EIM access is type A or not, and the other signals, B, C, and D, are similarly defined. The write enable (WEN) signals of the four banks are generated by a WEN controller.

Table II compares the proposed turbo decoder with previous designs. To show the efficiency of the TOD method, the throughput is normalized by the equivalent gate count. It is clear from the normalized throughput that the proposed TOD is effective in enhancing the throughput. Note that the proposed design provides a normalized throughput higher than those of previous works designed for LTE-Advanced applications.

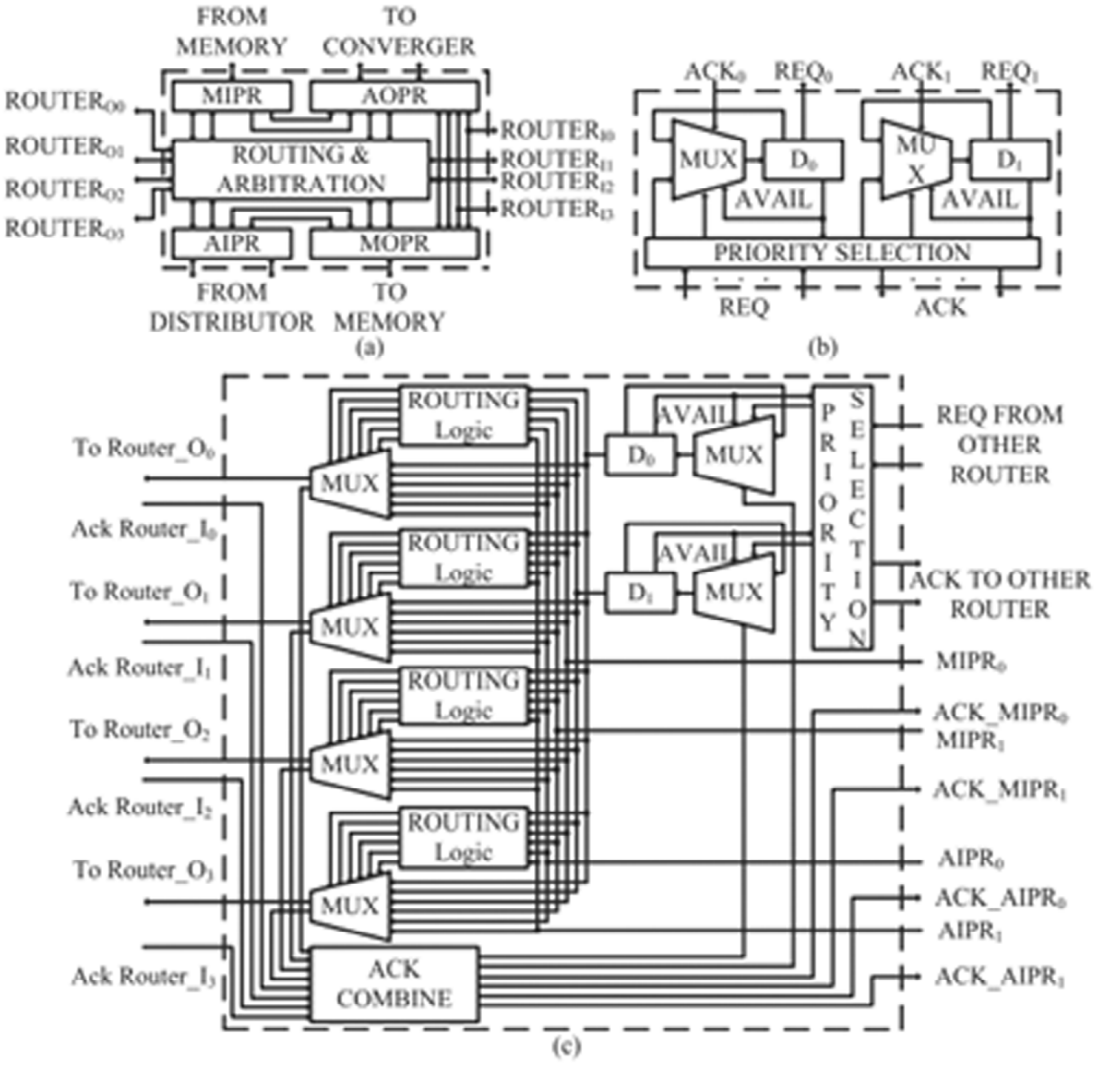


Figure 3: (a) Router architecture. (b) Pipeline registers module. (c) R&A module

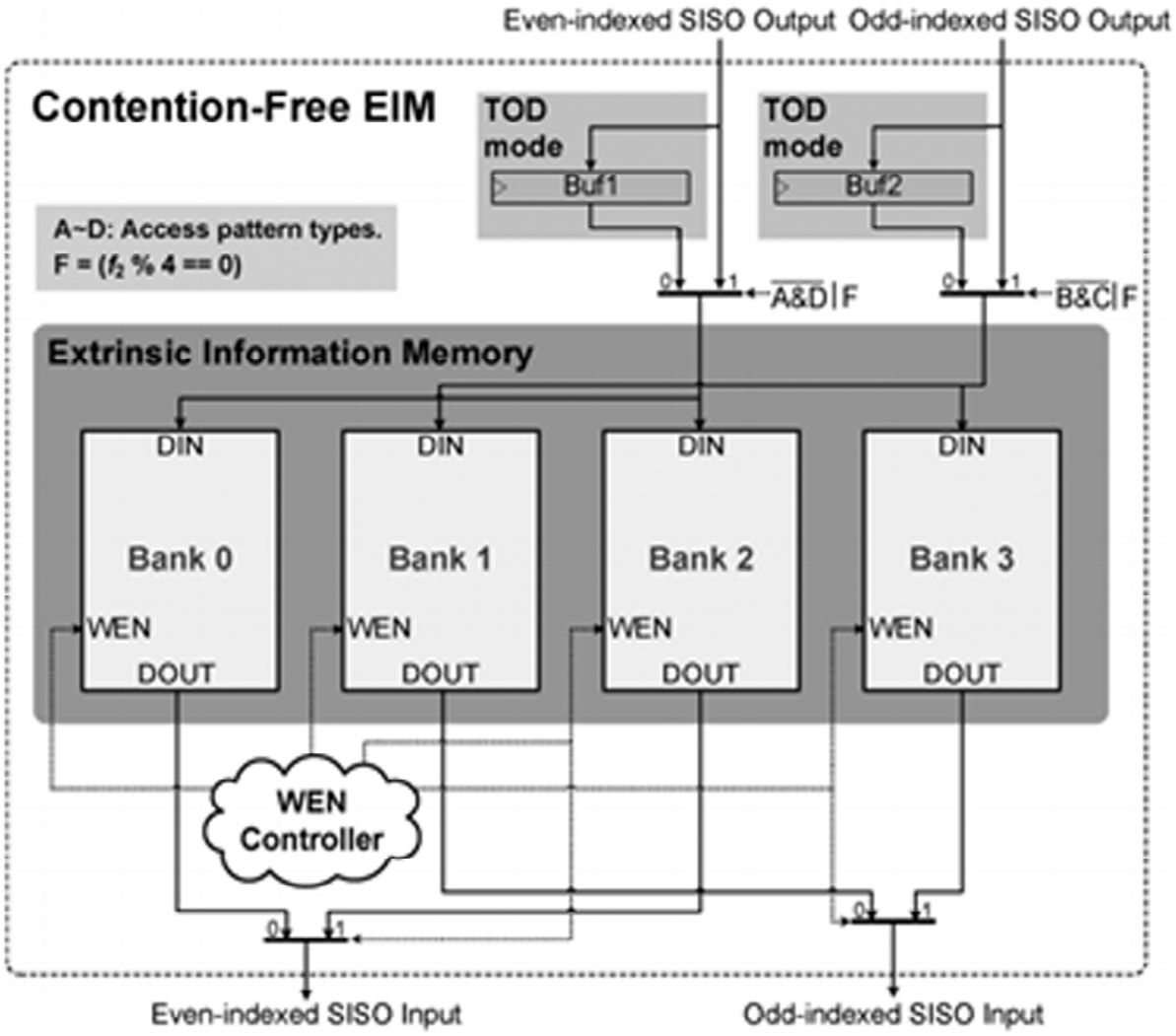


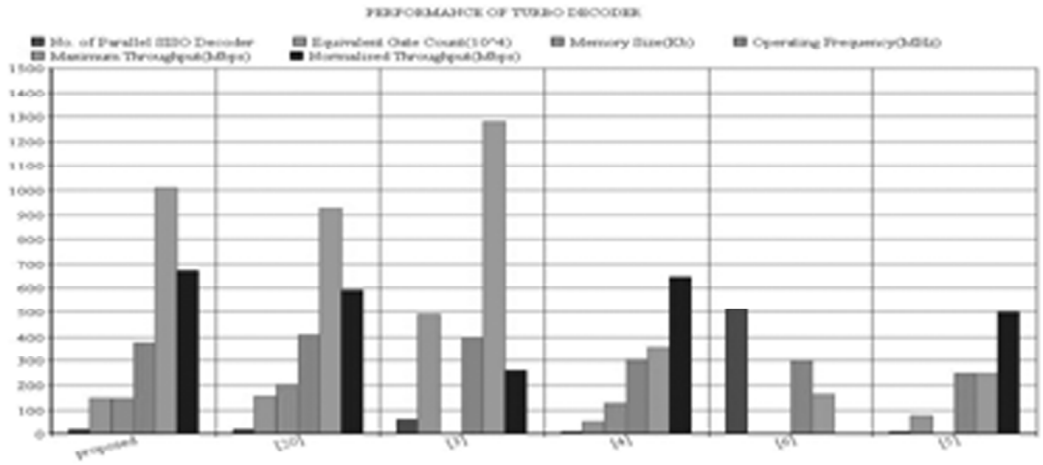
Figure 4: Contention-free EIM and its controller that performs the Normal radix-4 decoding and the tail-overlapped decoding.

Table 2

Architecture	*Proposed*	*FA[3]*[b]	*AP[3][7]*[b]
Technology	0.13µm	0.13µm	0.13µm
Clock Freq (MHz)	300	200	200
Area (mm^2)	3.3[a]	5.68	3.45
Max Throughput (Mbps)[c]	565(HSDPA) 694(LTE)	372(HSDPA) 312(LTE)	372(HSDPA) 312(LTE)
Throughput to area Ratio (Mbps/mm^2)[c]	171(HSDPA) 210(LTE)	65(HSDPA) 55(LTE)	112(HSDPA) 90(LTE)
Power Efficiency (mW/Mbps)	0.76(HSDPA) 0.62(LTE)	–	1.36(HSDPA)[d] 1.64(LTE)

3.1. Performance of Turbo Decoder

The aim to increase the throughput by reducing number of parallel SISO decoders can be achieved by this tail-overlapped decoding (TOD).Eventhough the clock frequency increases, with the less use of power and area maximum throughput with less complexity is achieved.



4. CONCLUSION

This paper has presented a promising way that enhances the throughput of LTE-Advanced turbo decoding. The proposed method called tail-overlapped decoding (TOD) totally removes the phase-switching latency. In this brief, we have proposed a novel architecture for a multi-ASIP turbo decoder with new decoding algorithm. By dividing the entire network into several sub networks and adopting the notion of the calculation sequence, we successfully achieved a much higher throughput with a lower total area. Comparisons with previous designs for area, throughput, and power efficiency have been given. Our design techniques can also be used in other applications, such as Low Density Parity check decoding. Moreover, the pro-posed decoder supports both the normal turbo decoding and the TOD methods without any noticeable hardware overhead. As a result, the decoder meets the 4G throughput requirement of 1 Gbps with moderate complexity and provides nearly optimal performance for various code rates.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes,” in Proc. Int. Conf. Commun., Vol. 2, pp. 1064–1070, May 1993.

[2] Multiplexing and Channel Coding (Release 11) 3GPP TS 36.212 v11.3.0, Jun. 2013.

[3] Y.Sunand J.R.Cavallaro, “Efficient hardware implementation of a highly-parallel 3GPP LTE/LTE-advance turbo decoder,” Integr. VLSI J., Vol. 44, no. 4, pp. 305–315, Sep. 2011.

[4] C. Studer, C. Benkeser, S. Belfanti, and Q. Huang, “Design and implementation of a parallel turbo-decoder ASIC for 3GPP-LTE,”IEEE J. Solid-State Circuits, Vol. 46, No. 1, pp. 8–17, Jan. 2011.

[5] J.-H. Kim and I.-C. Park, “A unified parallel radix-4 turbo decoder for Mobile WiMAX and 3GPP-LTE,” in Proc. IEEE Custom Integr. Circuits Conf., pp. 487–490, Sep. 2009.

[6] M. May, T. Ilnseher, N. Wehn, and W. Raab, “A 150 MBit/s 3GPP LTE turbo code decoder,”in Proc. Design, Autom. Test Eur. Conf., Mar. 2010, pp. 1420–1425.

[7] C.-C. Wong and H.-C. Chang, “High-efficiency processing schedule for parallel turbo decoders using QPP interleaver,” IEEE Trans. Cir-cuitsSyst.I,Reg.Papers, Vol. 58, No. 6, pp. 1412–1420,Jun. 2011.

[8] Physical Layer Procedures (Release 11) 3GPP TS 36.213 v11.4.0, Sep. 2013.

[9] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate, “IEEE Trans. Inf. Theory, Vol. IT-20, No. 2, pp. 284–287, Mar. 1974.

[10] A. J. Viterbi, “An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes,” IEEE J. Sel. Areas Commun., Vol. 16, No. 2, pp. 260–264, Feb. 1998.

[11] H.-M. Choi, J.-H.Kim, and I.-C. Park, “Low-power hybrid turbo decoding based on reverse calculation,” IEICE Trans. Fund. Electron. Comm. Comput. Sci., Vol. E89-A, No. 3, pp. 782–789, Mar. 2006.

[12] M. Zhan, J. Wu, L. Zhou, and Z. Zhou, “A memory access decreased decoding scheme for double binary convolutional turbo code,”IEICE Trans. Fund. Electron. Comm. Comput. Sci., Vol. E96-A, No. 8, pp.1812–1816, Aug. 2013.

[13] C.-H. Lin, C.-Y. Chen, A.-Y. Wu, and T.-H. Tsai, "Low-power memory-reduced trace back MAP decoding for double-binary convolutional turbo decoder," IEEE Trans. Circuits Syst. I, Reg. Papers, Vol. 56, No. 5, pp. 1005–1016, May 2009.

[14] J.-H. Kim and I.-C. Park, "Bit-level extrinsic information exchange method for double-binary turbo codes," IEEE Trans. Circuits Syst. II, Exp. Briefs, Vol. 56, No. 1, pp. 81–85, Jan. 2009.

[15] A. Worm, H. Lamm, and N. Wehn, "A high-speed MAP architecture with optimized memory size and power consumption," in Proc. Work-shop Signal Process. Syst, pp. 265–274, Oct. 2000.

[16] C. Benkeser, C. Roth, and Q. Huang, "Turbo decoder design for high code rates," inProc. IEEE/IFIP Int. Conf. VLSI Syst.-on-Chip, pp. 71–75, Oct. 2012.

[17] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log do-main," inProc. Int. Conf. Commun., Vol. 2, pp. 1009–1013, Jun. 1995.

[18] A. Nimbalker, Y. Blankenship, B. Classon, and T. K. Blankenship, "ARP and QPP interleavers for LTE turbo coding," inProc. Wireless Commun. Netw. Conf., pp. 1032–1037, Mar. 2008.

[19] Y. Wu, B. D. Woerner, and T. K. Blankenship, "Data width requirements in SISO decoding with modulo normalization," IEEE Trans. Commun., Vol. 49, No. 11, pp. 1861–1868, Nov. 2001.