VLSI design of a novel area efficient architecture of metric compression turbo decoding

G. Surekha[1] R. Bhavana[2]
[1]Associate professor, Department of ECE, GRIET, Telangana, India.
[2]M.Tech Student, Department of ECE, GRIET, Telangana, India.

Abstract: The turbo code is one of the most attractive forward error correction codes, which can provide near-optimal bit error rates (BERs) of Shannon's limit. Due to this feature in this paper to improve the memory in the decoding architecture a new technique named NII metric compression is proposed that instatements measurements for loosening up the capacity requests of turbo interpreters. The projected plot stores just the scope of state measurements and also two lists of the most extreme and least qualities, while the past pressure techniques need to store the greater part of the state-owned measurements for instating the accompanying cycle. We additionally introduce an equipment benevolent recuperation system, which can be actualized by straightforward multiplexing systems. In this paper the proposed structure is designed using 64bits and compared with the 4 bit design. Experimental results are observed by Xilinx ISE 13.2.

Keyword: Turbo Decoding, NII metric compression.

I. INTRODUCTION

The turbo code is one of the most attractive forward error correction codes, which can provide near-optimal bit error rates (BERs) of Shannon's limit. The Turbo-Code encoder is built using a parallel concatenation of two Recursive Systematic Convolutional codes and the associated decoder, using a feedback decoding rule, is implemented as P pipelined identical elementary decoders. Due to the fascinating error-correcting performance, the turbo codes have been applied to various wireless communication systems to meet the evolving data rate requirements of emerging wireless communication technologies, many parallel architectures have been proposed to implement high throughput turbo decoders. For achieving a high decoding throughput, an aggressive puncturing on long turbo codes is normally defined at the recent wireless standards. The extreme case of 3GPP LTE-advanced specification, for example, necessitates a code length of 6144 bits and a code rate of 0.95.To minimize the performance loss in decoding of high rate code words, the next-iteration initialization (NII) scheme is widely accepted for the initialization of backward recursions instead of the traditional dummy calculation method. Turbo decoders for modern wireless communication systems are required to support a wide range of code rates. The maximum supported code rate has strong impact on the choice of turbo decoder algorithm and architecture. This paper explores the problem of achieving high performance with turbo decoders at high code rates, and provides solutions on algorithmic and architectural level. A standard-compliant turbo decoder ASIC prototype for 3GPP Evolved EDGE has been implemented in 0.18 µm CMOS, and corresponding measurements proof the results of our analysis.

II. TURBO DECODING

Decoding of error correcting codes is basically a comparison of the probabilities for different code words - or with convolutional codes, different paths in the trellis. When we talk about probabilities, it is always the probability of some event given a certain amount of information about this event. This is especially clear when we talk about probabilities of something that has already happened - which is always the case in coding theory. What we mean when we talk about the probability that x was sent, p(x), is the probability that x was sent given the amount of information that we have about the event. Usually that is only the received noisy version of x - and of course knowledge of the coding scheme, transmission link etc.

For turbo codes we have two encoded sequences. Clearly we must start by decoding one of them to get a first estimate of the information sequence. This estimate should then be used as a priori information in the decoding of the second

encoded sequence. This requires that the decoder is able to use a soft decision input and to produce some kind of soft output.

Use of the system in practice has shown that if we subtract the log-likelihood ratio of the a priori information after each constituent decoder and make a number of decoding iterations we get a system that is working remarkably well - for many applications it actually outperforms the previously known systems. Still, we must conclude that the final result after turbo decoding is a sub-optimal decoding due to the loops in the decision process. For low signal-to-noise ratios we may even see that the decoding does not converge to anything close to the transmitted codeword. The turbo decoder is shown in Figure 1.

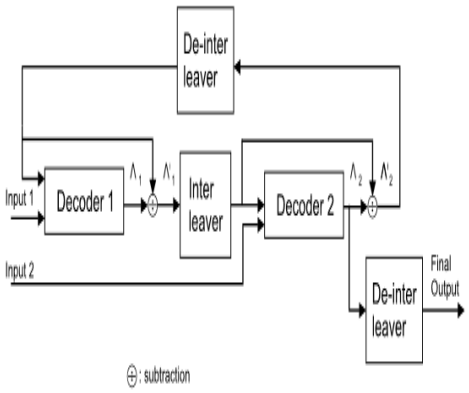

Figure 1 Turbo decoder

The conventional NII technique requires additional memories for storing all of the final backward states of the current iteration, which denote the starting confidence levels of the following iteration. If the sliding-window technique is used for practical realization, moreover, the number of NII metrics to be stored increases drastically according to the number of window boundaries. To mitigate the memory overheads, the static compression scheme in introduces a dedicated transfer function to encode NII metrics into 3 or 4 bits. More recent research presents a dynamic scaling factor for encoding of NII metrics.

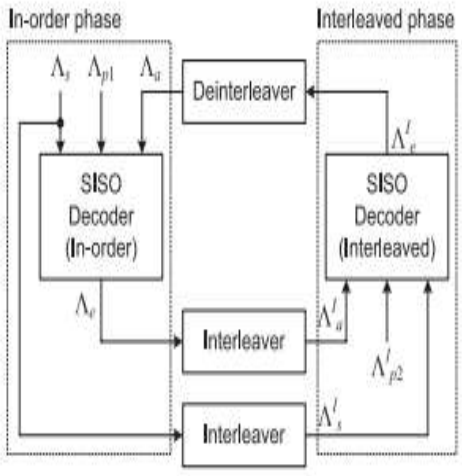

Fig. 2. Generalized turbo decoding architecture metrics

III. Memory-Reduced NII Metric Compression

In the contemporary turbo decoder, the max-log-MAP decoding algorithm is widely adopted due to the simple max operations instead of complicated max-star operations in the MAP algorithm the VLSI design aspect of high speed maximum a posteriori (MAP) probability decoders which are intrinsic building-blocks of parallel turbo decoders. For the logarithmic-Bahl–Cocke–Jelinek–Raviv (LBCJR) algorithm used in MAP decoders, we have presented an ungrouped backward recursion technique for the computation of backward state metrics. Unlike the conventional decoder architectures, MAP decoder based on this technique can be extensively pipelined and retimed to achieve higher clock frequency. Additionally, the state metric normalization technique employed in the design of an add-compare-select-unit (ACSU) has reduced critical path delay of our decoder architecture. We have designed and implemented turbo decoders with 8 and 64 parallel MAP decoders in 90 nm CMOS technology. VLSI implementation of an 8 parallel turbo-decoder has achieved a maximum throughput of 439 Mbps with 0.11 nJ/bit/iteration energy-efficiency. Similarly, 64 parallel turbo-decoder has achieved a maximum throughput of 3.3 Gbps with an energy-efficiency of 0.079 nJ/bit/iteration. These high-throughput decoders meet peak data-rates of 3GPP-LTE and LTE-Advanced standards as the max-log-MAP decoding only focuses on the trellis path having the maximum reliability, it is necessary to determine the most reliable state at the initializing process of

each sliding window. Unlike the previous works preserving each state metric value as much as possible, the proposed NII metric compression considers the range of state metrics denoted as Δx, i.e., the difference between the maximum and minimum state values among the wxth backward state metrics, βwx(•). It is possible to make the bit-width of Δx smaller than d, the bit-width of each state metric, as the saturation of ranges exceeding a certain value is acceptable without degrading the BER performance. Based on the numerous simulations, only 8 bits are enough.

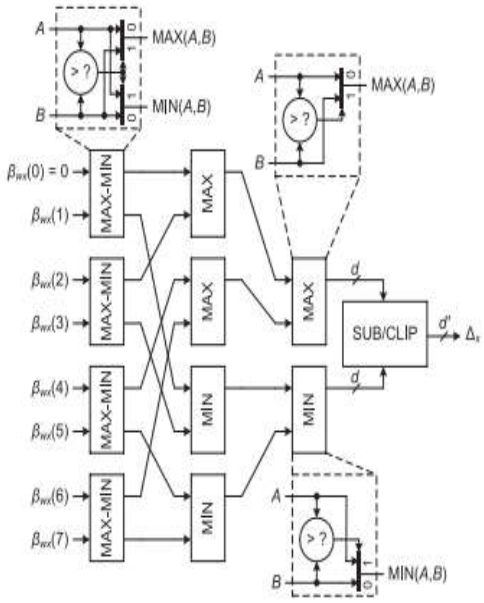

Fig. 3. Cost-effective compressing process of eight state metrics based on the proposed NII metric compression.

IV. PROPOSED SYSTEM

Proposed of another structure by widening the amount of bits to existing system of 64 bits. The entire square outline is same and the fundamental refinement is the amount of bits. So we are using the same existing square charts. The sliding-window framework is extensively recognized for the progressing turbo decoders to diminish the degree of inward supports. In light of the most extraordinary a posteriori (MAP) unwinding computation, each sliding window at first procedures state estimations recursively forward path by using the relating branch estimations. For ease, k forward state estimations of the ith trellis step are portrayed as αi(0), αi(1),...,αi(k − 1). By then, as showed up in Fig. 2, the retrogressive recursion enrolls the outward information of each trellis wander and likewise the accompanying state estimations in switch bearing. Like the forward state estimations, k in switch state estimations of the ith trellis step are addressed as βi(0), βi(1),...,βi(k − 1). Preceding beginning the regressive recursion, it is vital to reasonably introduce the beginning confirmation levels of each retrogressive state. In the NII structure, as appeared in Fig. 2, the toward the end in reverse conditions of every window oblige are anchored to be utilized for the beginning stages at the going with in reverse recursion of the relating stage. Enduring that the greater part of the state estimations are regulated by the zeroth state, i.e., αi(0) = βi(0) = 0, a total of 32 256 bits must be secured to understand the customary NII plot, where d and w are believed to be 12 and 32, independently [8].

To diminish the limit demands caused by the NII contrive, the static encoding procedure is for the most part used in the progressing turbo decoders [8]. In the count, the committed trade work is familiar with keep NII estimations to the force of twos. Fig. 3(a) shows the 3-bit encoding trade work, which maps every data NII metric onto one of the seven encoded regards.

In the contemporary turbo decoder, the greatest log-MAP translating estimation is by and large grasped due to the clear max assignments as opposed to jumbled max-star exercises in the MAP figuring the VLSI plan some portion of highspeed most we have shown an ungrouped in invert recursion system for the estimation of in turn around state estimations. Not at all like the consistent decoder models. VLSI utilization of 8 parallel turbo-decoder has achieved a most outrageous throughput of 439 Mbps with 0.11 nJ/bit/cycle imperativeness capability. Likewise, 64 parallel turbo-decoder has achieved a most extraordinary throughput of 3.3 Gbps with an imperativeness profitability of 0.079 nJ/bit/cycle. These high-throughput decoders meet apex data rates of 3GPP-LTE and LTE-Advanced rules.

As the greatest log-MAP disentangling just spotlights on the lattice way consuming the best steadfastness, it is imperative to choose the most strong state at the instating strategy of each window. Not in any manner like the past works shielding each state metric motivating force anyway much as could be normal, the projected NII metric weight considers the extent of state estimations implied as Δx, i.e., the alteration

between the most outrageous and slightest state regards among the wxth in invert state estimations, βwx(•). It is possible to make the bit-width of Δx more diminutive submersion of achieves outperforming a particular regard is tasteful without degrading the BER execution [9]. In perspective of the different proliferations, only 8 bits are adequate to address Δx, while each one of a kind state metric requires in any occasion more than 12 bits to neutralize surges in the LTE-impelled structures .To give an understanding for the assurance levels of each state at the recovery technique, in addition, we store the records of the most outrageous and slightest states, addressed as IMAX x and IMIN x, separately.

Adeptly, the proposed weight system endeavors to offer the correct information of the apex differentiates by giving up the precision of each state estimations, while the past works simply consider the approximations of each state metric. Multiplication results, exhibit that the BER of the proposed work using exact Δx is basically indistinguishable to those of the past NII metric encoding designs. Subsequently, the proposed estimation diminishes the amount of limit bits fundamentally without degrading the screw up changing capacity. Right when the degree of a sliding window is set to 32, for the occasion of 6144-piece turbo codes, our weight contrive uses only 5376 bits for NII information, which is 6 times not as much as the standard estimation. Like the past works, the proposed NII metric weight similarly requires additional computations for the encoding and deciphering shapes.

In the proposed weight, as exemplified in Fig. 3 overseeing eight in turn around states implied as βwx(•), the amount of examinations can be uncommonly diminished by sharing the transitional results. To find Δx capably, three basic modules are utilized in the proposed designing. The SUB/CLIP unit figures the last yield Δx with the reduced piece width d.

It is striking that IMAX x and IMIN x can be easily made by social event the prior examination results [10]. Note that it is hard to diminish the amount of examinations at the past estimation as most of the pressing structures are self-sufficient of each other.

V. RESULTS

Result of the proposed design is implemented using Xilinx ISE for simulation and Synthesis.

Proposed Results.

Simulation.

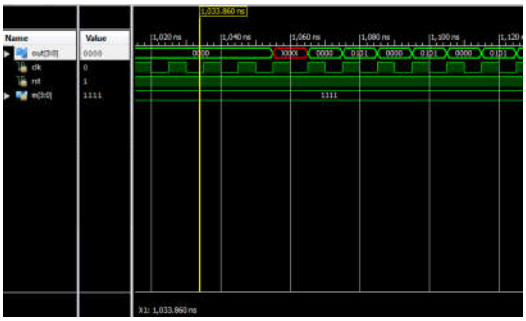

Fig. 4 Simulation.

Synthesis Result:

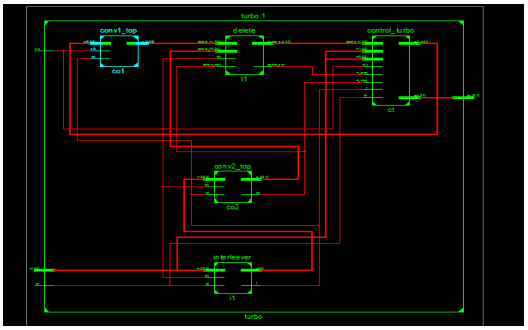

Fig 5 RTL Schematic

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	282	4656	6%
Number of Slice Flip Flops	379	9312	4%
Number of 4 input LUTs	397	9312	4%
Number of bonded IOBs	10	232	4%
Number of GCLKs	1	24	4%

Fig. 6 Design Summary.

```
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
  Total number of paths / destination ports: 48 / 4
-------------------------------------------------------------------------
Offset:              6.487ns (Levels of Logic = 3)
  Source:            ct/cnt_m1_0 (FF)
  Destination:       out<3> (PAD)
  Source Clock:      clk rising

  Data Path: ct/cnt_m1_0 to out<3>
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDRSE:C->Q           13   0.514   0.836  ct/cnt_m1_0 (ct/cnt_m1_0)
     RAM16X1D:DPRA0->DPO   1   0.612   0.387  ct/Mram_d_middle3 (ct/_varindex0002<2
     LUT4:I2->O            1   0.612   0.357  ct/out<2>1 (out_2_OBUF)
     OBUF:I->O                 3.169          out_2_OBUF (out<2>)
    ----------------------------------------
    Total                      6.487ns (4.907ns logic, 1.580ns route)
                                       (75.6% logic, 24.4% route)

=========================================================================
```

Fig. 7 Timing Summary.

Extension Results.

Simulation:

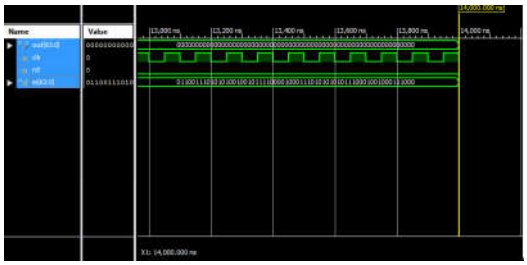

Fig. 8 Simulation

SYNTHESIS Result:

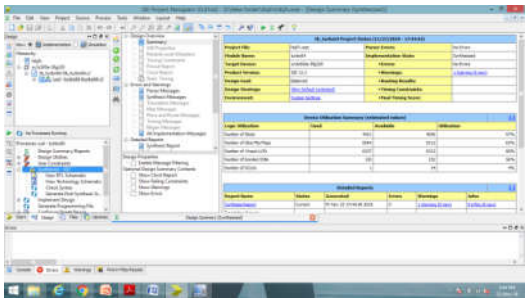

Fig. 9 Design Summary.

```
Data Path: m4/m4/ct/cnt_m1_0 to out<63>
                                  Gate     Net
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  -----------
   FDRSE:C->Q           13   0.514   0.836  m4/m4/ct/cnt_m1_0 (m4/m4/ct/cnt_m1_0)
   RAM16X1D:DPRA0->DPO   1   0.612   0.387  m4/m4/ct/Mram_d_middle3 (m4/m4/ct/_varindex0002<
   LUT4:I2->O            1   0.612   0.357  m4/m4/ct/out<2>1 (out_62_OBUF)
   OBUF:I->O                 3.169          out_62_OBUF (out<62>)
  ----------------------------------------
  Total                      6.487ns (4.907ns logic, 1.580ns route)
                                     (75.6% logic, 24.4% route)

==========================================================================


Total REAL time to Xst completion: 68.00 secs
Total CPU time to Xst completion: 68.42 secs

-->
```

Fig. 10 Timing Summary.

COMPARISON TABLE.

Design	Area			Delay(ns)
	Slices	Flip Flops	LUTs	
Existing System	282	379	397	6.487
Proposed System	4543	6064	6337	6.487

VI. CONCLUSION

The proposed structure is designed with 64 bits and compared with 4 bit design. The design could reach the memory demands of turbo decoders. By storing the precise ranges rather than the individually compressed metrics, the proposed algorithm remarkably reduces the size of NII metric memory while achieving an attractive error-correcting capability. Simulation and Synthesis is observed by Xilinx.

REFERENCES

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

[2] Multiplexing and Channel Coding (Release 11),3GPPTS36.212v11.3.0, Jun. 2013.

[3] G. Wang et al., "Parallel interleaver design for a high throughput HSPA+/LTE multi-standard turbo decoder," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 5, pp. 1376–1389, May 2014.

[4] R. Shrestha and R. P. Paily, "High-throughput turbo decoder with parallel architecture for LTE wireless communication standards," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 9, pp. 2699–2710, Sep. 2014.

[5] L. Li, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "A lowcomplexity turbo decoder architecture for energy-efficient wireless sensor networks," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vo1. 21, no. 1, pp. 14–22, Jan. 2013.

[6] C.Benkeser, C. Roth, andQ.Huang, "Turbodecoder designfor highcode rates," in Proc. IEEE Int. Conf. VLSI-SoC, 2012, pp. 71–75.

[7] C. Roth, S. Belfanti, C. Benkeser, andQ. Huang, "Efficient parallel turbodecoding for high-throughput wireless systems," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 6, pp. 1824–1835, Jun. 2014.

[8] J.-H. Kim and I.-C. Park, "Double-binary circular turbo decoding based on border metric encoding," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55, no. 1, pp. 79–83, Jan. 2008.

[9] I. Yoo, B. Kim, and I.-C. Park, "Memory-optimized hybrid decoding method for multi-rate turbo codes," in Proc. IEEE Veh. Technol. Conf. Spring, 2013, pp. 1–5.

[10] Y. Lee,B.Kim,J.Jung, andI.-C.Park, "Low-complexity tree architecture for finding the first two minima," IEEE Trans. Circuits Syst. II, sExp. Briefs, vol. 62, no. 1, pp. 61–64, Jan. 2015.