International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Low Complexity Architecture for Max* Operator of Log-MAP Turbo Decoder

Navneet Chouhan[†*] **and D. K. Mishra**[†]

[†]Department of Electronics & Instrumentation Engineering, SGSITS, RGPV, Indore, M.P- 452003, India

Accepted 02 July 2015, Available online 03 July 2015, **Vol.5, No.4 (Aug 2015)**

Abstract

A new approach for finding the two maximum values from the serial inputs is proposed. The architecture given is a hybrid combination of the two approaches namely: (a) tree-structure approach (b) radix approach, arranged preferably in less complex manner. Since, it is observed that the number of registers used and time delay in the design is comparatively less, it is believed to be less complex architecture known so far. Improvement in the memory consumption has also been noticed in the design. The architecture is intended for use in the iterative decoders like Log Map decoder for max operation.*

Keywords: *Radix approach; tree based; Iterative decoders; log-map decoders; max* operator; maximum value generators (mVGs); look-up tables*

1. Introduction

Turbo codes are the most attractive error correcting codes in today's scenario because their performance is near the Shannon limit (Berrou, *et al*, 1993) i.e. near the optimal error correction capability. This technique invented by Berrou in 1993, makes its presence significant in various wireless application standards like 3G and post-3G technologies (3GPP, 3GPP2 and 3GPP-LTE), Wireless LAN and WiMAX etc.

Many algorithms have been developed for achieving the Bit Error Rate (BER) performance of turbo decoders. The Maximum-a-Posteriori (MAP) algorithm provides best BER performance along with the simplicity of logic, but its implementation in VLSI is quite complicated. In logarithmic domain, however, the algorithm becomes simple and easy to implement on hardware, thus making it preferable to use in the turbo decoders.

The Log-MAP algorithm makes use of max* operator for computation of the Log Likelihood Ratio (LLR) values, used for decision making. However, this operator needs the first and second maximum values among the input sequence for its operation. The design for finding first maximum is simple, but the architecture for finding the second maximum value becomes bulky and complex because of numerous repetitions in the hardware.

Several researches have been devoted for the work of reducing implementation complexity of max* operator both at algorithmic and the implementation level. The architecture with Tree Structure (TS) (Martina *et al*, 2013) resulted in reduced area consumption along with improvement in speed of design; however undesirable repetition of hardware is found on analysis. Another approach that implemented radix architecture i.e. parallel design approach overcame this problem, however area complexity and feasibility of design is observed to be its drawback.

Here we proposed a hybrid architecture for 8 input maximum value generator unit, designed for max* operator, with the goal to overcome lacking of previous works. The proposed architecture shows improvement in terms of complexity, memory consumption and delay.

The simulation of the proposed work is done using Questa-10.e (vendor Mentor Graphics) using System-Verilog HDL and the operation is verified with the help of assertions. Synthesis is done on Xilinx ISE 9.2 using Verilog HDL and the results obtained are analyzed in terms of the delay and the number of comparators used. The design is also implemented in Cadence Virtuoso and delay and power results are obtained for the 8 input unit.

The contents in the paper are arranged as follows: Section 2 is a brief description of Turbo encoding and decoding with Log-MAP algorithm. Further the algorithm of TS approach and radix approach for finding the two maximum values from the given input sequence is discussed in Section 3. The implementation of proposed work is shown in section 4 and the simulation results are discussed in section 5 and finally conclusion is given in section 6.

*Corresponding author: **Navneet Chouhan**

2. Turbo Encoder and Decoder

A typical turbo encoder consists of two Recursive Systematic Convolutional (RSC) encoders and an inter-leaver (Sklar, 2nd edition, p. 475-504); the block diagram is as shown in figure 1, where Π is the symbol used for a device called inter-leaver, which randomly shuffles the position of the input bits. The architecture of RSC block consist of shift registers i.e. D blocks along with ex-or operator, as shown in figure 2.

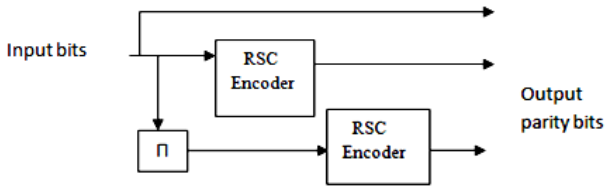

Fig.1 Turbo Encoder

The input data-bits are encoded by the first RSC encoder and output is the first parity bit sequence along with systematic input data stream .The second RSC block encodes the interleaved pattern of the input bits and output is second parity bit sequence. These parity bits are redundant information concatenated with the input bits and are transmitted over the channel.

At the receiver, decoder re-generates the original bits from the received data. The decoder, as shown in figure 3, typically consists of two MAP decoders and an inter-leaver. The output of decoder is computed with the help of three calculation blocks i.e. Path Metric Calculation (PMC) unit, Branch Metric Calculation (BMC) unit and Log Likelihood Ratio (LLR) calculation unit, which helps in simplifying the decision making procedure for the data received.

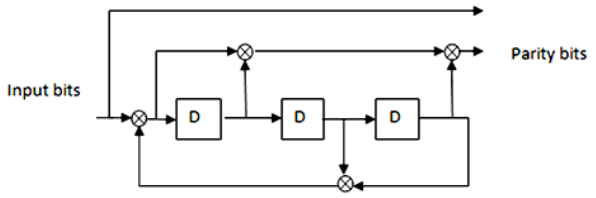

Fig.2 RSC Encoder

The final decision for source bits is made using the LLR values calculated using following equation:

$$L(dk) = \log\left|\frac{P(dk=1/Y)}{P(dk=0/Y)}\right| \quad (1)$$

LLR is a function of probability and it is further decomposed into three terms, as

$$L(dk) = Lapri(dk) + Lc(dk) + Le(dk) \quad (2)$$

where Lapri(dk) is the a-prioiri information of dk, Lc(dk) is the channel information and Le(dk) is the extrinsic. There is possibility that data bits transmitted might get corrupted because of noise in the channel and let the received bits are yk, y_k^{p1} andy y_k^{p2}.

The received data bits along with parity sequence-1 are applied to first decoder and corresponding output is inter-leaved and fed as input to the second decoder with parity sequence-2. In order to improve the precision the output from decoder-2 is de-interleaved and fed again to the decoder-1.

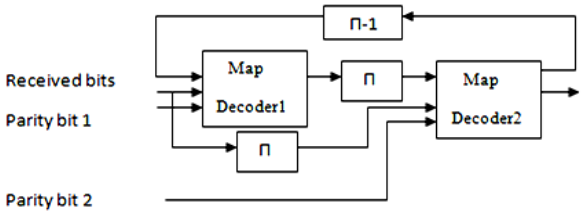

Fig.3 Turbo Decoder

The process is repeated 4-5 times and is called iterative decoding process, after which soft decision is made depending on the output LLR bits so obtained. MAP decoder can employ many algorithms like MAP, log-MAP, constant Log-MAP and MAX-Log-MAP etc, for decoding the received bits.

Each algorithm provides specific improvement in terms of implementation complexity, but with variation in BER performance, however, among them log-MAP algorithm provides optimal results in terms of BER along with reduced complexity and thus preferred for hardware implementation.

Log MAP Algorithm

The logarithm computations (Sklar, 2nd edition, p. 475-504) can be eliminated by the following approximation:

$$\max^*(x,y) \triangleq \ln(e^x + e^y) = \max(x,y) + \log(1 + e^{-|y-x|}) \quad (3)$$

Further this equation is simplified as shown below:

$$max^*\{\chi\} \approx y_1 + \log\left(1 + e^{-\delta}\right) = y_1 + fc(\delta) \quad (4)$$

where χ= {x1, x2….xn} set of n-input values.

$$y_1 = \max\{\chi\} \text{ And } y_2 = \max\left\{\frac{\chi}{y_1}\right\} \quad (5)$$

where y_1 and y_2 are first and second maximum values among the n inputs, respectively and value of constant is given by

$$fc(\delta) = \begin{cases} 3/8, & -2 < \delta < 2 \\ 0, & else \end{cases} \quad (6)$$

A low complexity architecture for hardware implementation of equation 4 and 5 is provided (Martina *et al.*, 2013) using TS approach. This shows good performance in area and delay, but it is observed

that it results in unwanted iterations of the basic hardware in finding the second maximum or value.

3. Algorithm for Finding the Two Maximum Values from the Input Sequence

Two techniques namely, TS approach and Mixed Radix architecture (MRA) approach are discussed here for finding the maximum values from the input numbers and are discussed below.

Tree Structure (TS) Approach

In this approach, a basic four input maximum value generator unit (4-mVG) block is constructed by two 2-mVG blocks and a Connection Unit (CU) which is further combination of 2-mVGs (Wey, Shieh, and Lin, 2008). The 2-mVG receives two inputs and produces the maximum 1st and maximum 2nd, using Maximum Value Unit (MVU) which include 2-mVG again. Similarly, for 4 inputs two blocks of 2-mVG produces respective two maximum values with two minimum ones, and the connection unit identifies the first and second maximum.

Similarly for making 2^K –mVGs unit two 2^{K-1} mVGs units are used along with connection unit. The design performs well in terms of both area and speed. MVG unit (Martina, *et al*., 2013) utilizes TS approach for implementing the max* operator to provide simplified architecture, but more number of repeated blocks for finding the values.

Mixed radix Architecture (MRA)

The procedure to find the first two maximum values out of M assigned ones can be decomposed in $\log_2(M)$ levels where level- l contains M/2 comparing stages (CSs) with $1 \leq l \leq .\log_2(M)$. CSs at the first level receive two input values and sort them, so each CS is made of one comparator.

Each CS at higher levels receives two couples of sorted values from the previous level and outputs are sorted couple. In MRA approach (Amarù, Martina, and Masera, 2012), different levels in a tree use different radix. Each CS is made of three main parts: an array of comparators, some one-hot index generators (OHIGs), and multiplexer-like structures (MLSs).

As discussed, this approach provides better performance in terms of delay and number of iterations for finding the second maximum values. However for higher values of input sequence this architecture might become inflexible and difficult to handle, because of the increasing number of comparators at each level of radix.

4. Proposed Design Architecture

In our work, we proposed architecture for finding two maximum values; designed using hybrid approach and a comparison is made with that of reference design models.

The model for n input sequence uses two n/2-MRA architectures at the initial stage of n-MVG unit as shown in figure 4 (shaded region). The sorted output from first stage, which is a parallel arrangement of comparators, is fed to the next block input which consists of three MVUs. These blocks sort the received input and the output obtained is utilised in LUTs for finding second maximum.

Since second block design is fixed irrespective of the number of input sequence flexibility of design increases, as only first block is needed to be changed for variation in the number of input sequence. Therefore, the over-all power consumed by the design is also reduced along with improved design. A constant value needed for addition as per previous researches, it is calculated as c_0 in the figure with the help of Look-up Tables (LUTs).

5. Result and Discussion

The proposed design is synthesized using Xilinx ISE 9.2 and simulation is done on Questa 10.1e. Design is also implemented on Cadence Virtuoso and the results obtained are as discussed below:

HDL Implementation

The proposed work along with previous work is implemented with System Verilog HDL. Log map decoder is also coded on the HDL and the performance of decoder is analysed with the proposed design and previous design in terms of the number of registers used and the memory consumed.

Results conclude that memory consumption is same in all three design approaches, as shown in table 2, however the number of register are improved. Thus design seems less complex as compared to previous work.

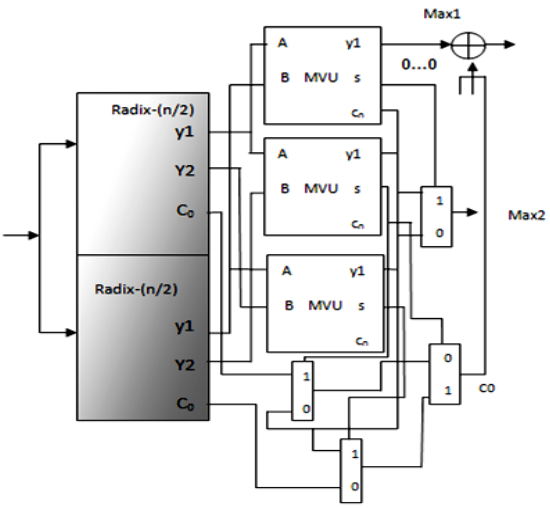

Fig.4 Proposed Architecture for maximum value generator for 8 input

The design is synthesized using Verilog HDL on Xilinx and the results are compared in terms of maximum

delay and the number of comparators used with that of the reference design as shown in Table 1.

The results show about 18% propagation delay improvement w. r. t. tree approach whereas in radix design delay improvement is prominent when input sequence increase beyond 16.

The RTL view is as shown in figure 5. However, the number of comparators used increases about 20% from tree structure but much better than radix approach. This reduces area requirement of proposed design.

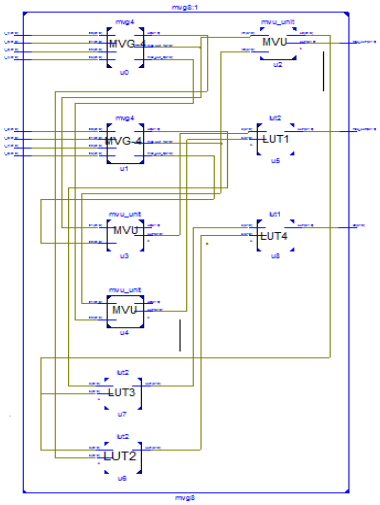

Fig.5 RTL view of 8 mVG unit obtained using Xilinx ISE 9.2

Table 1 Design analysis in terms of delay and comparators used

max* operator	No. of Inputs	Propagation delay (ns)	Comparators used
TS approach	8	15.099	13
	16	20.623	29
	32	26.129	61
Radix-M	8	11.214	18
	16	15.684	42
	32	48.600	136
Proposed architecture	8	11.527	17
	16	16.134	37
	32	21.007	77

Table 2 Analysis of decoder design on basis of registers used and memory consumed

Log-MAP Decoder implementation	Registers used	Memory utilization
TS approach	1033	7.6k
Radix-M	1008	7.6k
Proposed architecture	1017	7.6k

Implementation on Cadence

The simulation of the proposed work is done using 180nm technology on Cadence Virtuoso Version 11.1.0.523.isr15. The schematics of the 8 input mVG unit is obtained using basic logic gates and look-up-tables (LUTs) (shown in figure 6(a)) .The transient analysis is done with the input bits given as voltage, 1.8 V is for logic-1value and 0 V as logic-0, the waveform for dc power analysis is shown in figure 6(b).

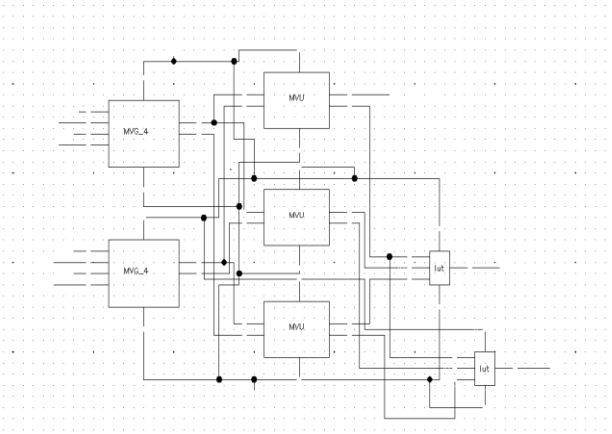

Fig.6 (a) Schematics of 8-mVG on Cadence

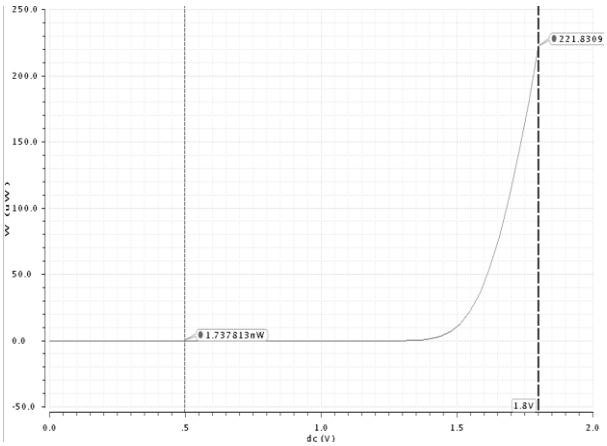

Fig.6 (b) Power graph obtained for dc response of 8-mVG unit

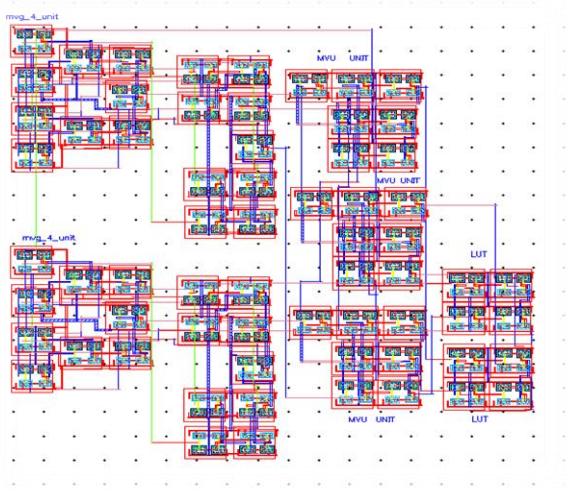

Fig.7 Layout for mVG-8 unit on Cadence

The delay obtained between input and maximum-1 output delay about **250 E-12** seconds. The total time elapsed for dc analysis is 467.929 ms with the peak memory consumption of about **33.6 Mbyte** is reported.

The total static power consumed by the design is about 221 micro-watts, i.e. minimum power consumption by the design. The area occupied by the layout is around $1mm^2$ as shown in figure 7.

Conclusion

These days low complexity design is prominent requirement of iterative decoders. We have tried to propose a design that fulfils the necessity with the implementation of the max* operator using improved architecture. The following improvement is concluded:

1) The complexity is reduced as the number of registers and memory consumption decreases by 50%.
2) The overall synthesis result provides satisfactory performance for propagation delay i.e. the speed of the output is improved.
3) VLSI implementation can be easily achieved with minimum consumption of area and power as shown with back end analysis results on cadence.

References

C. Berrou, A. Glavieux, and P. Thitimajshima (1993), Near Shannon limit error-correcting coding and decoding: turbo-codes', *Proc. IEEE Int. Conf. Commun*., 1064-1070.

L. Bahl, J. Cocke, F. Jelinek, and J. Raviv (1974), 'Optimal decoding of linear codes for minimizing symbol error rate,' *IEEE Trans. Information Theory*, 20, 284-2137.

Maurizio Martina, Stylianos Papaharalabos, Panayiotis Takis Mathiopoulos and Guido Masera (2013), 'Simplified Log-MAP Algorithm for Very Low-Complexity Turbo Decoder Hardware Architectures,' *IEEE Trans. Instrumentation & Measurement*, 18, 9456.

C. L. Wey, M. D.Shieh, and S. Y. Lin (2008), 'Algorithms of finding the first two minimum values and their hardware implementation,' *IEEE Trans. Circuits Syst. I*, vol. 55, no. 11, 3430–3437.

L. G. Amarù, M. Martina, and G. Masera (2012), 'High speed architectures for finding the first two maximum/minimum values,' *IEEE Trans.Very Large Scale Integer. (VLSI) Syst*., vol. 20, no. 12, pp. 2342–2346.

Bernard Sklar, Digital Communication-Fundamentals and Applications, 2nd edition Prentice Hall PTR, Pg.475-504.