International Journal For Technological Research In Engineering
Volume 1, Issue 9, May-2014 *ISSN (Online): 2347 - 4718*

DESIGN OF ADD COMPARE SELECT UNIT BASED TURBO DECODER ARCHITECTURE FOR 4G WIRELESS NETWORKS

Saranyadevi. M[1], Visalatchi. D[2], Dr. G. K. D. Prasanna Venkatesan[3]
[1]PG Student, VLSI Design, [2]Assistant Professor, [3] HOD
ECE Dept, PGP College of Engineering and Technology, Namakkal, India.

Abstract: Turbo decoder is typically one of the most computation-intensive parts in a 4G wireless receiver. Increased complexity and performance requirements and the need to reduce power and area are significant challenges for Turbo decoder hardware implementation. The push for multi-mode wireless physical layer (PHY) brings additional challenges for Turbo decoder design, lookup table-log-BCJR (LUT-Log-BCJR) architectures having low processing power consumption. In this project Proposed system used to achieves a low area and a low power consumption by decomposing the LUT-Log-BCJR architecture into its most fundamental Add Compare Select (ACS) operations and perform them using a new low-complexity ACS unit and validate ACS based architecture in the circumstance of the Long Term Evolution (LTE) turbo decoder and demonstrate the architecture that it has an order of magnitude lower chip area than the most recent LUT-Log-BCJR architectures. The ACS based LUT-Log-BCJR architecture is used to avoid the wastage of energy when compared in the conventional architecture and this method is done by using redesign the timing of the conventional architecture that allows its components to be efficiently merged. This produced a low number of low complexity functional units to perform the entire LUT-Log-BCJR algorithm. At finally the ACS based architecture is used to reduce the hardware complexity and increase the throughput, furthermore this architecture results in a low area and a high clock frequency, which achieve low power consumption.

General Terms: Wireless Networks, turbo codes.
Index Terms: Energy-efficient, error-correcting code (ECC), LUT-Log-BCJR algorithm, ACS unit and operations.

I. OBJECTIVE

The main objective of our projects is reducing area and energy efficient in LUT-LOG-BCJR Architecture by using Add Compare Select (ACS) unit in parallel. This ACS units is performs in ACS operation per clock cycle and results stored in registers and register banks. ACS based architecture having equal lengths; this is used to avoid watage of energy. The control signals of the ACS unit are provided by the operation code, which can be used to perform the function.For example Four ACS operation equivalent to a max* calculation to be performed in four clock cycles using a single ACS unit.

II. ACS UNITS

In decoder implementations the maximum operating frequency is limited by the recursive state metric computation, which cannot simply be pipelined or parallelized due to the presence of feedback loop. To reducing the complexity of the state metric recursions to shorten the critical path and to reduce area and power consumption. Implementation of the ACS unit as shown in fig. where error correction term is selected from LUT and added to output of the ACS unit to performance loss associated with the use of the max-log approximation. Because state matrics increase with time and normalization is performed by subtracting a bias that is computed outside the ACS recursion, further increasing complexity.

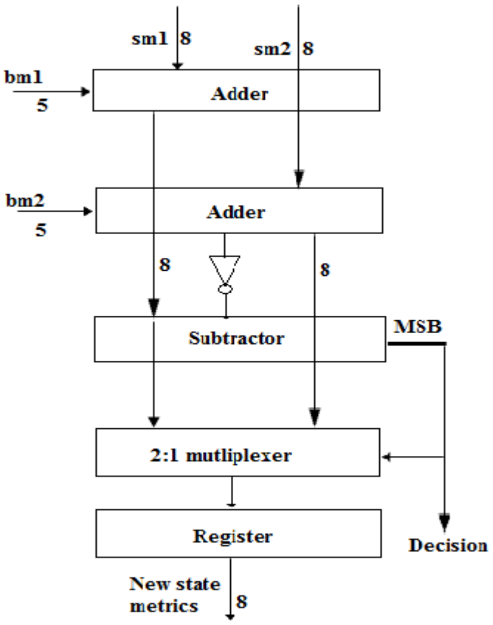


Fig.1: Block diagram of ACS unit

The cost of the lookup table (LUT) approach by using extrinsic scaling to close the error rate performance gap between the MAP and max-log algorithm. To solve the problem of increasing state metrics, the max-log –MAP algorithm has been implemented with modulo normalization for the state metrics. This technique known from Viterbi decoder implementation achieves the renormalization with a controlled in the data path and requires an additional 3 input XOR gate in each ACS unit. Block diagram of ACS unit is given below in Fig 1.

III. INTRODUCTION

Consider the turbo codes have recently found application in wireless communication system, since their near-capacity coding gain is used to helps reliable communication when using a reduced transmission energy and this reduction is offset by the turbo decoder's energy consumption, as well as the energy consumption of the turbo encoder. Therefore, turbo codes designed for energy constrained wireless communication system have to minimize the overall energy consumption. In this papers ACS based turbo decoder architecture delivering low area and low energy consumption. Add compare select (ACS) unit is one of the fundamental components in Viterbi decoder .It is also widely used in many portable applications with limited area and energy efficient. The Bahl-Cocke-Jelinek-Raviv decoder also called Maximum Posteriori Probability decoder (MAP).There is many types of BCJR used in Wireless sensor networks. LUT-Log- BCJR algorithm is mostly used in energy constrained application.

IV. PROCESS OF THE LUT-LOG-BCJR ALGORITHM

The LUT-Log-BCJR algorithm comprise only additions, subtractions and the max* calculation. Each addition and subtraction constitutes a single ACS operation, each max* calculation can be considered equivalent to four ACS operations. A total of (z+2) ACS operations are required to carry out the max* calculation. But only a single ACS operation is required when z=0 or when employing the Max-Log-BCJR algorithm, which approximate by the max operation. Same way fewer ACS operations are required, when employing the Constant-Log-BCJR algorithm. These alternative algorithms reduce the hardware complexity and increase the throughput, therefore reducing the energy consumption and this is achieved at the cost of requiring higher transmission energy to achieve the same BER performance.

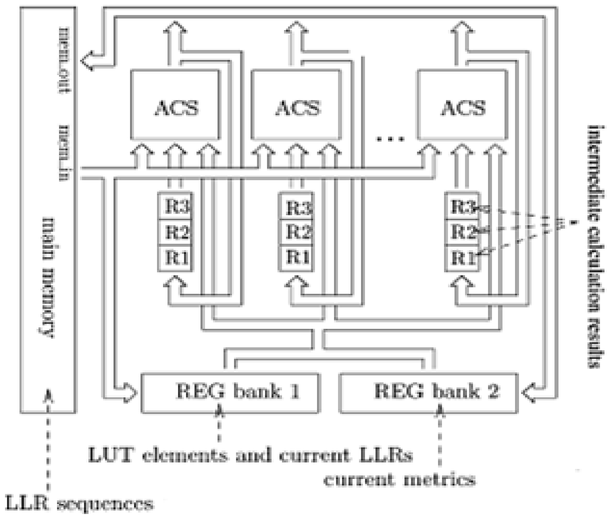


Fig. 2: LUT-LOG-BCJR Architecture

Our architecture implements the entire algorithm using 2^m ACS units in parallel, each of which performs one ACS operation per clock cycle. Furthermore, the proposed architecture employs a twin-level register structure to minimize the highly energy-consuming main-memory access operations. At the first register level, each ACS unit is paired with a set of general purpose registers R1, R2, and R3. These are used to store intermediate results that are required by the same ACS unit in consecutive clock cycles. This allows the four ACS operations equivalent to a calculation to be performed in four consecutive clock cycles using a single ACS unit. The second register level comprises REG bank 1 and REG bank 2, which are used to temporarily store the LUT-Log-BCJR variables between consecutive values of the bit index during the recursions decoding processes. The REG bank 1 comprises registers for the *a priori* LLRs and dummy registers for the required LUT constants. The sets of current metrics are stored in REG bank 2. The main memory stores all the required *a priori* LLR sequences and extrinsic LLR sequences during the decoding process and the state metrics from the previous window, which facilitates the processing of the entire LUT-Log-BCJR algorithm. Since the proposed architecture supports a fully parallel arrangement of an arbitrary number of ACS units.

V. IMPLEMENTING THE ACS UNIT

The ACS unit is responsible for implementing the state metric computation. A direct implementation requires two identical data paths each with two additions, a comparison, and a selection. The basic data path is shown in Fig.3. Note that although the branch metrics are positive numbers with 5 bits each, and state metrics are required to have 7 bits. After some amount of time, the state metrics will overflow so a trick called modulo normalization is used. The upshot is that one extra bit is required in the state metric (8 total) and the MSB of a simple subtraction can be used for the compare operation. Lastly, an array of 2:1 multiplexers is used to pick the minimum.

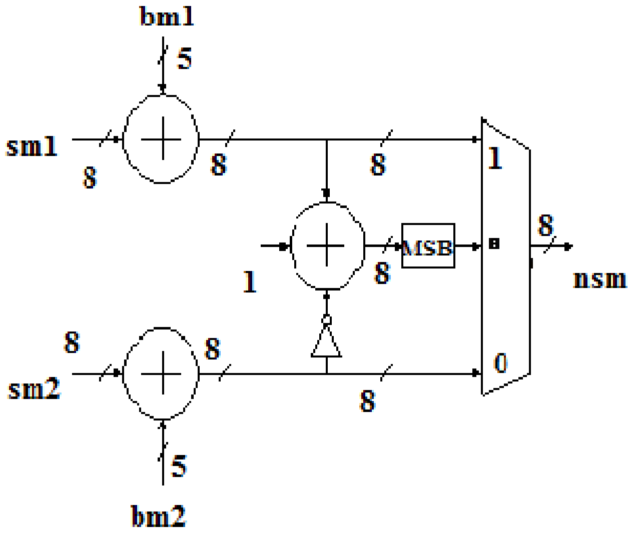


Fig. 3: Data path of ACS Unit

VI. ACS UNIT

This is the diagram of basic ACS unit and it performs one ACS operation per clock cycle. The control signals of the ACS unit are provided by the operation Code of O, O= {O0, O1, O2, O3, O4, O5}, which can be used to perform the functions listed in Table. Note that the operation code O = 101100_2 approximates the absolute difference between two operands, as required by $\max^*(p,q)$. Note that a simpler ACS unit implementation is too helped by this deliberately introduced inaccuracy, which can be trivially canceled out during the $\max^*$ calculation. And a $\max^*$ calculation can be performed with the following four operations, which store intermediate results in the registers R_1,R_2, and R_3, of Fig 2.

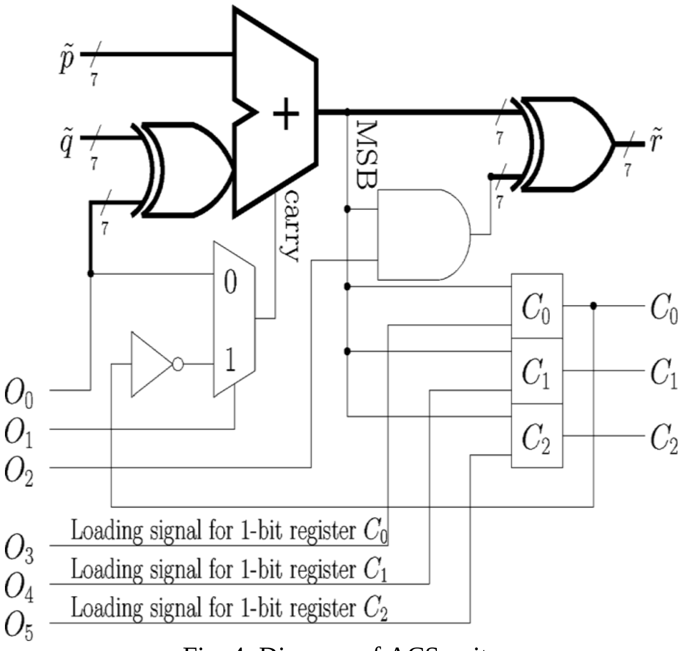

$\tilde{p}$
$\tilde{q}$
7
+
MSB
carry
$\tilde{r}$
0
1
C_0
C_1
C_2
O_0
O_1
O_2
O_3
O_4
O_5
Loading signal for 1-bit register C_0
Loading signal for 1-bit register C_1
Loading signal for 1-bit register C_2

Fig. 4: Diagram of ACS unit

VII. PROBLEM EXISTING

In wireless sensor networks, there are many types of BCJR used to design the architecture. It's used to get the output for much application. In Existing System conventional LUT-Log-BCJR architecture was used, which employs the sliding-window technique are used to generate the LLR sequence. Each of these windows is generated separately, using a forward, a pre-backward and a backward recursion. This architecture designed for achieving a high transmission throughput that is in excess of 100 Mbit/s.

Draw backs

The recursions involve calculations that must be performed in series and additional hardware are used to during synthesis.

- Clock frequency is high
- Throughput is high
- Chip area is high
- Energy consumption and Decoding complexity are high

VIII. PROPOSED SYSTEM

Since limiting the energy efficient and area this proposed system is to redesign the conventional architecture that allows low-complexity functional units, which are collectively capable of performing the entire LUT-Log-BCJR algorithm. Functional units are used eliminating the requirement for additional hardware and to avoid the wastage of energy. Clock gating technique used to reduction in Power and energy consumption. In system decomposed into classic ACS operations. In this system, it does not use separate hardware for the recursion and this architecture implements the entire algorithm using ACS units in parallel each of which performs one ACS operation per clock cycle. This motivate our low complexity energy efficient architecture, which achieve a low area and low energy consumption.

Advantages

- Chip area is low
- Clock frequency also low
- Energy consumption and throughput are low
- Decoding complexity is low.
- Improve the architecture performance.

COMPARISON OF THE IMPLEMENTED TURBO DECODER

Algorithm	This work	LUT-Log
Block size	6144	6144
Technology	90	90
Supply voltage	1.0	1.0
Area A mm^2	0.33	0.35
Gate count	7.5k	7.5k
Memory required	186	188
Clock frequency F (MHz)	159.481	333
Decoding iterations	5	5
Throughput T (Mb/s)	0.99	1.03
Power consumption (mW)	3.99	4.17
Energy consumption (nJ/bit/iteration)	0.36	0.4
$E_b^{tx} + E_b^{pr}$ (nJ/bit) when transmitting over 58m (5 iteration)	37.21	41.92

IX. HARDWARE COMPLEXITY AND ENERGY ANALYZE

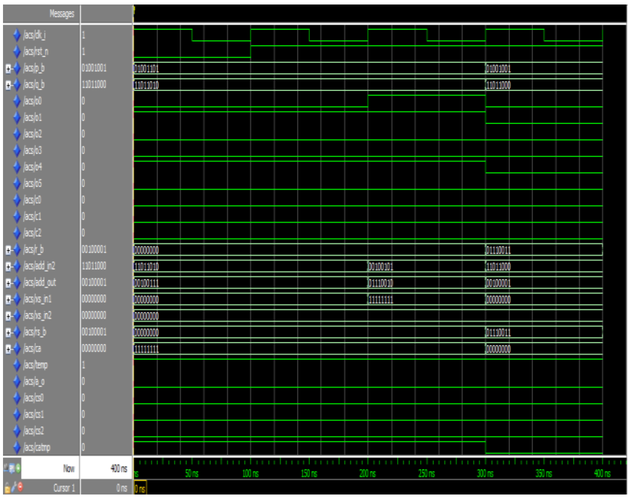
The hardware complexity of the proposed architecture is so low that the chip area is actually dominated by the memory module, which consumes the overall power consumption according to our post-layout simulation results. By contrast, the chip area of conventional LUT-Log-BCJR architectures is typically dominated by the decoder, despite employing similar amounts of memory. In Table I, we compare the proposed architecture to the latest LUT-Log-BCJR and Max-Log-BCJR decoder architectures. The area and power consumptions are estimated based on post-layout simulations. The implementation results arising from different technologies are also scaled to give a fair comparison. As shown in Table I, the energy consumption of the proposed architecture is significantly lower than that of the conventional LUT-Log-BCJR architectures.

X. SYSTEM REQUIREMENTS

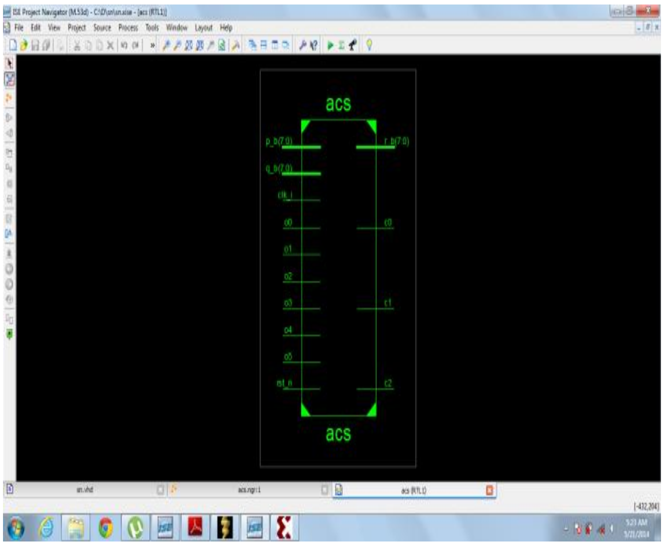
HARDWARE USED:
Processor – Intel core2 Duo
Speed - 2.93 GHz
RAM – 2GB RAM
Hard Disk - 500 GB
Key Board - Standard Windows Keyboard
Mouse - Two or Three Button Mouse
Monitor – LED

SOFTWARE USED
Operating System: XP and windows 7
Front End: Cygwin.

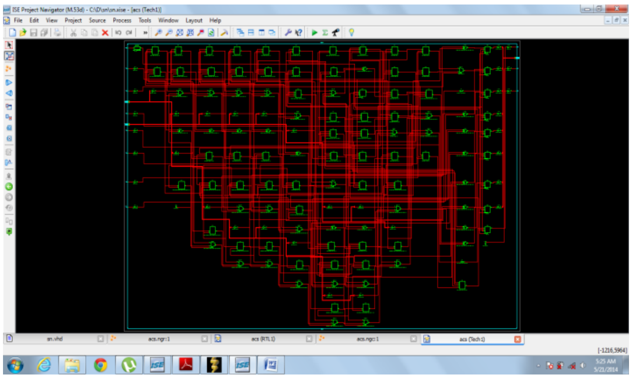
XI. SIMULATED RESULT



XII. RTL SCHEMATICS



XIII. TECHNOLOGY SCHEMATICS



XIV. CONCLUSION

A low-complexity LUT-Log-BCJR architecture, achieves a low area and hence a low power consumption by decomposing the LUT-Log-BCJR algorithm into its most fundamental ACS operations. This ACS based architecture may be readily reconfigured for different turbo codes or decoding algorithms and validated the architecture by implementing an LTE turbo decoder. A Turbo decoder is typically one of the most computation-intensive parts in a 4G wireless receiver. Increased complexity and performance requirements and the need to reduce power and area are significant challenges for Turbo decoder hardware implementation. ACS based turbo decoder produces an architecture comprising only a low number of inherently low-complexity functional units, which are collectively capable of performing the entire LUT-Log-BCJR algorithm. Furthermore, ACS based approach naturally results in a low area and a high clock frequency, which implies a low power consumption when compared to the existing system.

FUTURE WORK:
Different LUT-Log-BCJR architecture and their algorithm further improving for architecture performance.

REFERENCES

[1] Liang Li, Robert G. Maunder, Bashir M. Al-Hashimi, Fellow, IEEE, and Lajos Hanzo" A Low-Complexity Turbo Decoder Architecture for Energy-Efficient Wireless Sensor Networks". vol. 21, no. 1, pp14-22, January 2013.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci,"Wireless sensor networks: A survey," Comput. Netw.: Int. J. Comput. Telecommun. Netw. vol. 52, pp. 292–422, 2008.

[3] P. Corke, T. Wark, R. Jurdak, H. Wen, P. Valencia, and D.Moore, "Environmental wireless sensor networks," Proc. IEEE, vol. 98, no.11, pp. 1903–1917, Nov. 2010.

[4] W.-P. Ang and H. K. Garg, "A new iterative channel estimator for the log-MAP & max-log-MAP turbo decoder in Rayleigh fading channel," in Proc. Global Telecommun. Conf., 2001, vol.6, pp. 3252–3256.

[5] 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, pp. 8–17, 2011.

[6] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," Euro.Trans. Telecommun. vol. 8, no. 2, pp. 119–125, 1997.

[7] L. Li, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "Design of fixed-point processing based turbo codes using extrinsic information transfer charts," in Proc. IEEE Veh. Technol. Conf., 2010, pp. 1–5.

[8] C. Benkeser, A. Burg, T. Cupaiuolo, and Q. Huang, "Design and optimization of an HSDPA turbo decoder ASIC," IEEE J. Solid-State Circuits, vol. 44, no. 1, pp. 98–106, Jan. 2009.

[9] M. C. Valenti and J. Sun, "The UMTS turbo code and an efficient decoder implementation suitable for software-defined radios," Int. J. Wirel. Inform. Netw. vol. 8, no. 4, pp. 203–215, 2001.

[10] G. Barrenetxea, F. Ingelres, G. Schaefer, and M. Vetterli, "Wireless sensor networks for environmental monitoring: The Sensor scope experience," in Proc. IEEE Int. Zurich Seminar Commun., 2008, pp.98–101.

[11] Y. Zhang and K. K. Parhi, "High-throughput radix-4 log MAP turbo decoder architecture," in Proc. Asilomar Conf. Signals, Syst., Comput. 2006, pp. 1711–1715.