VOL. 11, NO. 15, AUGUST 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

PERFORMANCE EVALUATION OF ACS IN VITERBI DECODER USING PARALLEL PREFIX ADDERS

M. Balaji, P. Arun and V. Balamurugan
Electronics and Communication Engineering, Sathyabama University, Chennai, India
E-Mail: balajimadhavan16@gmail.com

ABSTRACT

Modern Digital Communication System usually employ convolutional codes with large constraint length for better decoding action, which leads to large intricacy and power consumption in Viterbi decoders. It is essential to use radix 2 in Viterbi decoder to prune significant portions of trellis state to dramatically power consumption, high rate, and area lessening. In these project, we are using the parallel prefix adder like kogge stone adder in ACS unit for the improvement of Viterbidecoders. To design and increase the performance evaluation of (Add/Compare and select) ACS unit in Viterbi decoder. To analyze the Viterbi Decoder algorithm. Implementation of efficient ACSU with different fast adders. Achieve more efficient power consumption and decreasing computational complexity.

Keywords: viterbi decoder, acs unit, parallel prefix adders.

1. INTRODUCTION

The basic units of Viterbi decoder are branch metrics, Add compare select and Survivor management unit Viterbi decoder for decoding uses the Viterbi algorithm a bitstream that has been encoded using convolutional or trellis code Viterbi has become a standard term for the application of raising programming algorithms to maximization problems involving probabilities. The basic units of Viterbi decoder are branch metrics, Add compare select and Survivor management unit. A Viterbi decoder uses the Viterbi algorithm for decoding a bit stream that has been encoded using convolutional code or trellis code. Viterbi has become a Used for Satellite communication and interrupted communication channels. In principle the best way of decoding against general errors is to compare the received sequence with every possible codes equence. This process is best visualized using a code trellis which contains the information of the state diagram, but also uses time as a narrow axis to show the possible paths through the states. The corresponding output is written over the thick line. The forward and backward motion leaving a condition are labelled with the corresponding output. While a thick line denotes an input bit, zero, and a thick line denotes an input bit, one. The decoding algorithm uses two types: the branch metric and the path metric. The branch metric is a measure of the "distance" between what was sent and what was obtained, and is defined for each arc in the trellis. In hard decision decoding, where we are given a series of digitized equality bits, the branch metric is the Hamming distance between the expected equality bits and the received ones. An example is shown, where the received bits are 00. For each state transition, the number on the arc shows the branch metric for that transition. Two of the branch metrics are 0, corresponding to the only conditions and transitions where the corresponding Hamming distance is 0. The other non-zero branch metrics similar to cases when there are bit bugs. The path metric is a value associated with a state in the trellis (i.e., a value associated with each node). For encoding, it corresponds to the Hamming distance over the most probable path from the starting state to the present state in the trellis. By "most likely", we mean the path with shortest Hamming distance between the starting state and the current state, measured over all possible paths between the two states. The path with the shortest Hamming distance minimizes the total number of bit errors, and is most likely when the BER is low. The key inspection in the Viterbi algorithm is that the receiver can compute the path metric for a (state, time) pair increases using the path metrics of early calculated states and the branch metrics.

2. LITERATURE SURVEY

Sushama *et al.*, 2015, proposed a way to block the codes for transmission over noisy channel. In this system Viterbi is the main block of CDMA or wireless system. And it was designed and targeted to FPGA. The main purpose of this work is to expand the pipeline process to improve, efficient data transmission.

Kim *et al.,* 2015 proposes that radix 4 ACSU architecture for Viterbi decoder and he used retiming method to reduce the critical path delay and have applied the rearranging method proposed to reduce the hardware size. This existing architecture is implemented in FPGA.

Padma *et al* -2015 proposed the accomplishment of Convolutional Encoder and Viterbi Decoder (VD) with a constraint length, K of 9 and a code rate (k/n) of 1/2 using FPGA technology.

Thulasi Ram. M (2015) proposed the discovery of cyclic redundancy codes (CRC) is primarily used for error detection rather than error correction. This will reduce the hardware part. The Viterbi algorithm (VA) is mainly made to decode the convolutional codes. Encoder consists of less complexity when compared to the decoder. Current design of an Adaptive Viterbi Decoder (AVD) that uses survivor path with parameters for wireless communication in a process to reduce the power, cost and increase in speed at the same time. Viterbi Algorithm (VA) requires an expandable increase in hardware complexity to achieve greater decode accuracy. Constraint length associated with the given bits are vast, hence it needs to implement the larger constraint

length with little hardware and lesser calculations for decode the original data. When the decoding process uses the Modified Viterbi Algorithm (MVA) calculations 50% reduced and reduction in the hardware utilization, which follows the maximum- likelihood path.

Shraddha Shukla (2014) proposed an error free communication system can be realized efficiently by adding suitable excessiveness at the source end to augment error resilience of the coded bit streams. The code can be resumed significantly even in the condition of noisy channels. This paper concluded that the hard decision technique can detect any multiple number of errors which is most likely to occur inside the transmitting media due to presence of an Additive White Gaussian Noise (AWGN). An error less communication system can be realized efficiently by adding appropriate excessiveness at the source end to augment error resilience of the coded bit streams. The code can be recovered significantly even in the condition of noise. The error can be detected and/or corrected by given hardware and software.

Sudharani B K (2015) proposed on the realization and implementation of an efficient logic design of a convolutional encoder and adaptive Viterbi decoder (AVD) called cryptosystem with a constraint length, K of 3 and a code rate (k/n) of 1/2 using Field Programmable Gate Array technology. The survivor path algorithm used the address of the memory unit to select the correct part which concentrates the output code.

Kang and Wilson (1996) proposed an application of present low energy design methodology at various levels. At the architecture level, they suggested partition of major blocks and memory modules to reduce the power dissipation. Energy efficient data transfer and storage organization for MAP turbo decoder was advanced by Curt Schurgers *et al* (1999). Storage bottleneck in data dominated algorithm (MAP turbo decoding) had decreased by Data Transfer Storage Exploration (DTSE) through the steps pre-processing, global dataflow change, global loop and re-indexing. Twice flow structure reduced the decoding delay of one data block. Design and Operation of the original MAP turbo decoder algorithms were extremely important to allow implementation of the decoder

An Intuitive explanation and a made more simple Implementation of the MAP Decoder for Convolutional Codes was discussed by Andrew J. Viterbi (1998) proposedan spontaneous least pathway to understanding the maximum a posteriori19 (MAP) decoder was presented based o imprecise solution. This is shown to be equivalent to dual-maxima calculation combined with forward and backward recursions of Viterbi algorithm computations. The logarithmic form of the MAP algorithm could similarly be reduced to the same form by applying the same imprecise solution. Inversely, if a correction term was added to the approximation, the exact MAP algorithm was resumed.

Hsu *et al* (1998) presented a parallel decoding scheme for turbo codes. They proposed a presented method for decreasing the decoding delay by means of portioning a block into several sub-blocks, which were partially overlapped. The proposed sub-block portion scheme allows for the parallel decoding of each component code by using several sub-block decoders. The number of steps for the recursive computations in each sub-blockdecoder is reduced to O(N/W), where W is the number of portioned sub-blocks. The decoding delay is equivalently one with that of an unoriginal MAP-based turbo-coding system.

3. PARALLEL PREFIX ADDERS

Parallel Prefix Adders have been fixed as the most thorough circuits for binary addition in digital systems. Their regular structure and fast execution makes them particularly enticing for VLSI implementation. The delay of a parallel prefix adder is basically depends on the number of levels in the carry propagation stage. The parallel prefix adders examined in this paper are: Kogge Stone Adder, Brent Kung Adder and Knowles Adder. The performance metrics considered for the study of the adders are: power, delay and area. Using simulation studies, delay, area and power performance of the different types of adder were obtained. It was observed that Kogge Stone Prefix tree adder has better circuit characteristics in terms of accomplish compared to adders realized using other algorithms. The addition of two binary numbers is one of the most important mathematical function in modern digital systems such as microprocessors and digital signal processors. In these systems 0 and 1 adders are used in arithmetic logic units (ALU), multipliers, memory address generation and dividers. The necessity of adders is that it should be fast and efficient in terms of power and chip area. Most often, the highest operating speed of most of the modern digital systems depend on how fast adders can access the data and hence answerable for setting the minimum clock cycle time in processors [16].

Kogge Stone adder

The schematic of Kogge Stone Adder is shown. It is widely used in high performance applications. The general concept of Kogge Stone adder is almost the same as that of the carry look ahead adder except for the second step, called parallel carry prefix chain. In the first level (L=1), generates and propagates of 2-bit are computed at the same time. In the second level (k=2), generates and propagates of 4bit are calculated by using the result of 2-bit in level 1. Therefore, the actual carry-out value of the 4th bit would be available while the calculations in level 2 are being computed. In the third level (L=3), the carry-out of the 8th bit is computed by using the 4th bit carry result. The same method adopted in level 3 is applied to get carry-out values of the 16th bit and the 32nd bit in level 4 and level 5. All other carries of bit are also computed in parallel. In Figure 1, red boxes are propagate (P) and generate (G) generators for each bit of two inputs. Yellow boxes contain propagate block and generate block and the delay of one yellow box is equal to two gate delay (D). The blue boxes keep the original generate value transmitted from the previous level. In each level, because

all carries are calculated in parallel, the delay is the running time of single yellow box [18].

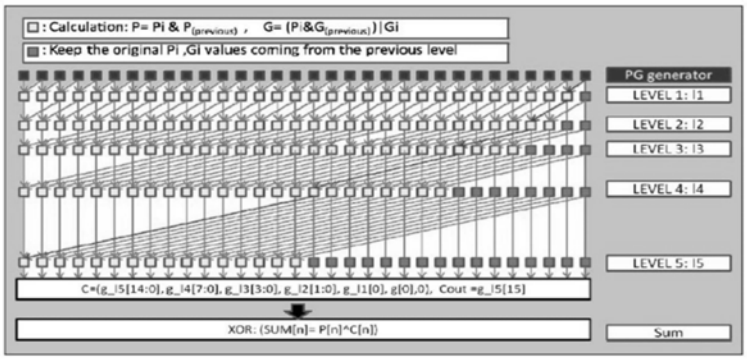

Figure-1. Kogge Stone adder.

Brent kung adder

This gives the schematic of the Brent Kung Adder. Brent Kung is a parallel prefix form of the carry look ahead adder. In carry lookahead adder, as the size of the input operands is increased the delay of the result is also increased. Therefore the idea here is to have a gate level depth of O (log2(n)). It takes less area to implement than the other prefix adders such as Kogge Stone adder and it also has less wiring congestion. Instead of using a carry chain to calculate the output, the method shown in Figure-2 is used. This will reduce the delay without compromising the power performance of the adder [17].

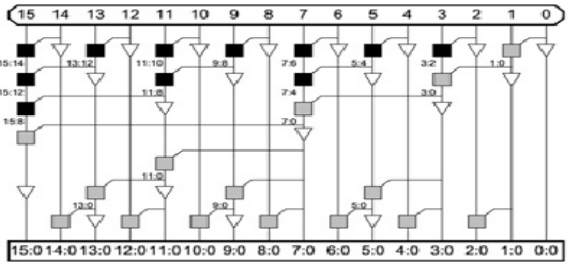

Figure-2. Brent Kung adder.

Knowles adder

Knowles adder is similar to Kogge Stone Adder, but it has different logic to calculate the output. It illustrates a 16-bit Knowles Adder [16].

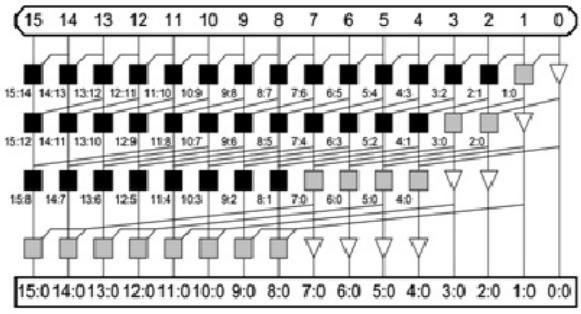

Figure-3. Knowles adder.

4. RESULTS AND DISCUSSIONS

The simulation results are shown from Figures 4, 5, 6 and 7 with using Radix2 and different adders

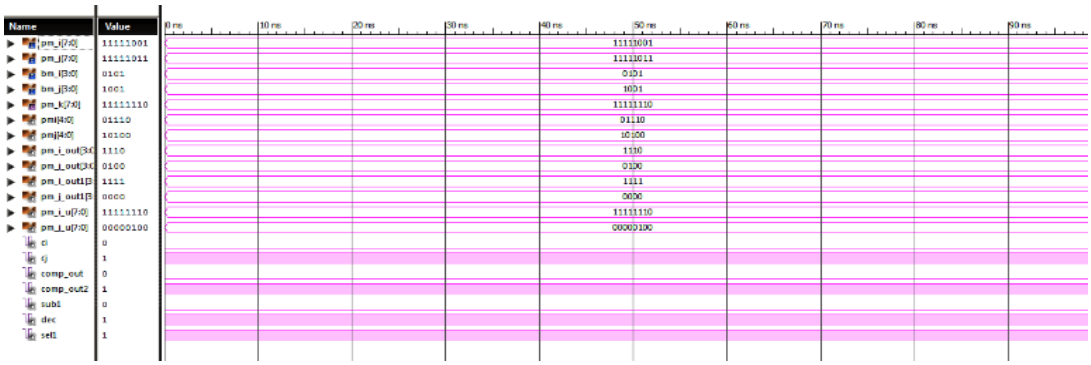

Figure-4. Simulation wave form using Fast Radix 2 ACSU with CLA.

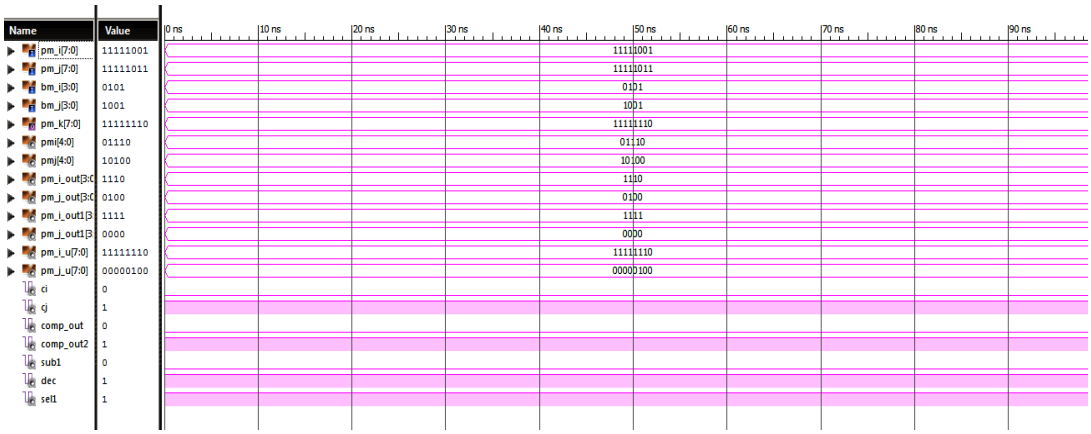

Figure-5. Simulation wave form using Fast Radix 2 ACSU with Sparce KS adder.

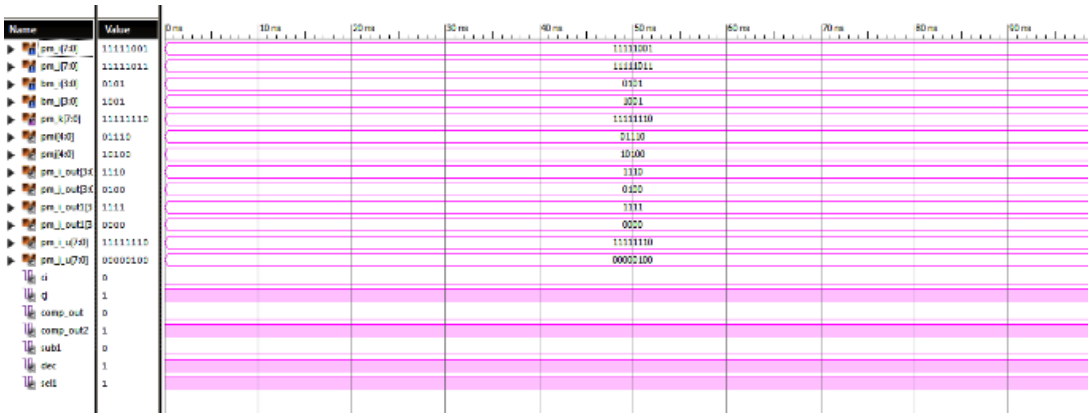

Figure-6. Simulation wave form for Fast Radix 2 ACSU with Brent Kung adder.

The Fig describes the simulation output of Fast Radix 2 Add-Compare-Select Unit based on Brent Kung adder. The test inputs are pm_i, pm_j, bm_i, bm_j and the out waveform is pm_k. The test input values are pm_i = 11111001, pm_j = 11111011, bm_i = 0101, bm_j = 1001 and the new path metric value is pm_k = 11111110. Remaining wave forms in the output window are internal signals of the architecture. The Fig describes the Register Transfer Level Schematic for Fast Radix 2 Add-Compare-Select Unit based on Brent Kung adder.

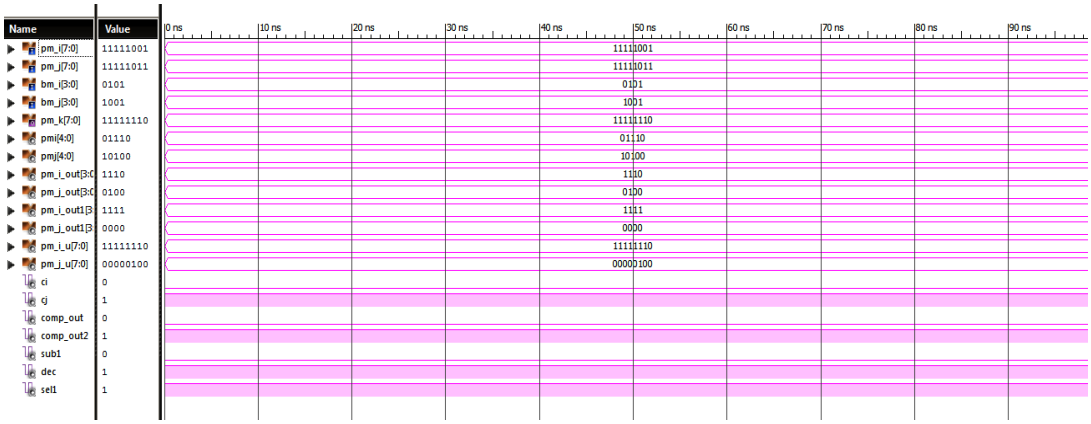

Figure-7. Simulation wave form for Fast Radix 2 ACSU with Knowles adder.

The above figure describes the simulation output of Fast Radix 2 Add-Compare-Select Unit based on Fast Radix2, Sparce KS, Brent Kung, Knowles adders. The test inputs are pm_i, pm_j, bm_i, bm_j and the out waveform is pm_k.The test input values are pm_i = 11111001, pm_j = 11111011, bm_i = 0101, bm_j = 1001 and the new path

metric value is pm_k = 11111110. Remaining wave forms in the output window are internal signals of the architecture. The Fig describes the Register Transfer Level Schematic for Fast Radix 2 Add-Compare-Select Unit based on Knowles adder.

Table-1 shows the comparison results for high speed radix-2 ACSU with different fast adder techniques. To reduce the computational time, State exchange method is adopted. Hence, this method will reduce the time and power consumption of ACSU.

Table-1. Comparison report of ACSU with different adders.

Device utilization summary	CLA	Sparce KS	Brent Kung	Knowles
Number of Slices	20	21	20	23
Number of 4 input LUTs	36	37	36	41
Number of IOs	32	32	32	32
Number of bonded IOBs	32	32	32	32
Path delay	18.4ns	10.9ns	12.0ns	11.5ns
Power Consumption	45 mW	31 mW	33 mW	32 Mw

CONCLUSIONS

In this paper parallel prefix adders are implemented in fast radix 2adder to reduce the computational time of ACSU. The outputs are carried out in Xilinx. Convolutional codes are very popular over both wired and wireless links. In this paper, we have described a new area reduction and architecture of fast radix 4 ACS with same time and high speed application. The efficiency of ACS became increase. The area reduction is achieved by rearranging the radix 4 ACS operations and speed up is done by retiming method. In this paper, fast radix-4 ACS architecture proposed for low cost and high speed applications. The product of the proposed ACS architecture is improved by 11mW less power consumed and path delay also changes according to the adders. Since the proposed architecture is implemented and tested its efficiency, testing in custom design may be needed.

REFERENCES

Viterbi A.J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, in Information Theory, IEEE Transactions on. 13(2): 260-269, DOI: 10.1109/TIT.1967.1054010.

Vikas Gupta, Chanderkant Verma. 2012. My Viterbi vs MATLAB Viterbi. International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, 2(2).

Chakraborty D.; Raha P.; Bhattacharya A.; Dutta R. 2013. Speed optimization of a FPGA based modified viterbi decoder. In: Computer Communication and Informatics (ICCCI), 2013 International Conference on. pp. 1-6, DOI: 10.1109/ICCCI.2013.6466245.

Bobby N.D.; Srivatsa S.K.; Lalkishore. 2011. Implementation of Radix2 ACS in Adaptive Viterbi decoder. in Nanoscience, Engineering and Technology (ICONSET), 2011 International Conference on. pp. 604-606, DOI: 10.1109/ICONSET. 2011.6168043.

Habib I.; Paker O.; Sawitzki S. 2010. Design Space Exploration of Hard-Decision Viterbi Decoding: Algorithm and VLSI Implementation. in Very Large Scale Integration (VLSI) Systems, IEEE Transactions on. 18(5): 794-807, DOI: 10.1109/TVLSI.2009.2017024.

K. S. Arunlal; Dr. S. A. Hariprasad. 2012. An Efficient Viterbi Decoder. International Journal of Advanced Information Technology (IJAIT). 2(1).

Abubeker K.M.; Bushara A.R.; Backer S. 2013. Maximum likelihood DE coding of convolutional codes using viterbi algorithm with improved error correction capability. in Information and Communication Technologies (ICT), 2013 IEEE Conference on , pp.161-164, DOI: 10.1109/CICT.2013.6558082.

Bobby N.D.; Srivatsa S.K.; Kishore L.; Rajiv A.; Suresh S.S. 201. Comparison of fast radix 2 ACS with adaptive fast radix 2 ACS in Viterbi Decoder. in Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), 2013 International Conference on. pp. 1-5, DOI: 10.1109/ICEVENT.2013. 6496563.

Mandwale A.J.; Mulani A.O. 2015. Different Approaches for Implementation of Viterbi decoder on reconfigurable platform. in Pervasive Computing (ICPC), 2015 International Conference on, pp.1-4, DOI: 10.1109/PERVASIVE.2015.7086976.

Kumar Y.R.; Babu A.V.; Naveen Kumar K.A.; Rani Alex J.S. 2014. Modified Viterbi decoder for HMM based speech recognition system. in Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on. pp. 470-474, DOI: 10.1109/ICCICCT.2014.6993008.

Viraktamath S.V; Patil P.H.; Attimarad G.V. 2014. Impact of code rate on the performance of Viterbi decoder in AWGN channel. in Computational Intelligence and

Computing Research (ICCIC), 2014 IEEE International Conference on. pp. 1-4, DOI: 10.1109/ICCIC.2014.7238371.

Megha Talsania. A Comparative Analysis of Parallel Prefix Adders Department of Electrical and Computer Engineering, University of Texas at San Antonio.

Adilakshmi Siliveru, M. Bharathi. 2012. Design of Kogge-Stone and Brent-Kung adders using Degenerate Pass Transistor Logic.

Pakkiraiah Chakali, Madhu Kumar Patnala. 2013. Design of High Speed Kogge-Stone Based Carry Select Adder ISSN: 2319-6378, 1(4).