

Design and Performance of Low-Latency Viterbi Algorithm Architectures on ASIC and FPGA

Patan Jani Basha

Department of Electronics and Communication Engineering,
Nimra College of Engineering,
Ibrahimpatnam, Vijayawada, A.P-521456, India.

Syed Ali Hussain

Department of Electronics and Communication Engineering,
Nimra College of Engineering,
Ibrahimpatnam, Vijayawada, A.P-521456, India.

ABSTRACT:

The Viterbi algorithm is generally used in decoding convolutional codes utilized in communications. In this paper Viterbi decoder and convolution encoder is designed, used in speech to text conversion. We can utilize three variations for re-computing with operands and its alterations. The modified full adder circuit is subjected to low power consumption and low area. This can be implemented in Application Specific Integrated circuit (ASIC). This paper explains reduction of power and area by utilization of modified full adder. In this, Viterbi decoder is designed by utilizing the modified full adder. Viterbi decoder and convolution encoder are implemented in speech to text conversion application.

Index terms: Looking forward the procedure, re-computing through encrypted operands, trellis.

INTRODUCTION

The Viterbi algorithm is used method for decoding the convolution codes [1] effectively. This is mostly used in communication systems. By this algorithm codes are decoded and those are used in different applications like satellite communications, radio relay and cellular relay. The high speed viterbi decoders are implemented by using serializes and desterilizes which having the critical power utilization and area constraints. Serialize deserializers (SERDES) are utilized in optical or magnetic storage memory drives like hard disk drive(HDD) or digital video disc drive[2]. The Viterbi calculation is the comparative procedure for finding all likelihood successions of the states. In resulting sequence we observed the events and high efficiency

which consist of limited number of states possible. There are different types of decoders are available to decode the data. Among these the viterbi decoders are used to decode the data. The viterbi decoders are consist of three major components they are Branch metric unit (BMU), Path metric unit (PMU), Survivor memory unit (SMU). The input signal of viterbi decoder is given to Branch metric unit. The Branch metric unit gives the metrics equivalent to the binary trellis reliant on the received signal. The output of branch metric unit is given to the path metric unit. The path metric unit consist of Add-Compare-Select unit (ACS). It is the core element of path metric unit. The ACS unit updates the path metric units. SMU manages the survival paths. The output of viterbi decoder is taken at the output of survivor path memory unit. The branch metric unit and path metric unit uses the forward logic. The Add Compare Select unit uses the criticism circles. The speed is constrained by utilizing the include look at select unit. To breakdown the cycle bound for the viterbi decoder of limitation length K [4]-[10] M-step look forward systems are used. The speed of circuit is based on add-compare-select unit. One trellis step can combine the several trellis steps by using look ahead technique. The efficiency can be improved by using pipelining technique in add-compare-select unit. The iteration can be avoid by using pipelining technique. It is used in high speed communication systems. The add-compare-select unit front end of the Branch Metric Computation (BMP).

Cite this article as: Patan Jani Basha & Syed Ali Hussain, "Design and Performance of Low-Latency Viterbi Algorithm Architectures on ASIC and FPGA", International Journal & Magazine of Engineering, Technology, Management and Research, Volume 6 Issue 11, 2019, Page 43-49.

This uses the look-ahead technique. The branch metric calculation made out of pipelined enrolls between each two stages. BMP joins the parallel trellis of various strides into a solitary complex trellis of one stage. Just the include task is before the immersion of trellis. The add operation is performed before the compare operation formerly the saturation. If the corresponding paths are having the low metrics, these are neglected. Within the sight of Very Large scale integration (VLSI) the viterbi calculation is utilized in interpreting the convolution codes. Due to the faulty outputs the accuracy of decoding the convolution codes are degraded. In digital systems the faults are can happened through different ways including the alpha elements produced by bundle decay and cosmic rays which are generating the energetic neutrons. In advanced technologies the faults can be occurred by the device reduction, low power supply voltages and high operational frequencies. These can expand the possibility of transient mistakes. This can fundamentally influence the unwavering quality of calculations. Due to the cosmic rays a single event affronts and single event transients are produced. The vast beams make the vivacious protons and neutrons, warm neutrons, arbitrary clamor, or flag honesty issues, these all subsequent in gadget deficiencies. In different domains of hardware architectures and different arithmetic unit sub mechanisms [11] [12] error detection is important. In order to achieve the fault resistance, reliability and error recognition the natural faults are used. In reliable designs have discussed to counter natural or mischievous errors [13]. Through the concurrent error detection the cryptographic architectures are immune [14]-[24]. In this paper we use two methods in the viterbi algorithm. The area and power consumption are reduced with respect to the proposed architectures. To achieve the transient and per meant faults the computing with encoded operands are used. We use the three variations i.e. recomputed with shifted operands (RESO) [25], and modified computing with shifting operands which having the low fault and recomputing with encoded operands (RERO) [26] are used to detect the faults in the add-compare select unit. Hardware redundancy methods through the signature

based architectures are also used in RERO and RESO architectures. We use different types of self-checking methods based on two-rail encoding to adder components. These architectures are applied to low power consumption and lesser area structures of the viterbi decoders [2]. We encapsulates as follows, We propose a viterbi decoder and convolution encoder. The viterbi decoder uses the modified full adder [31] which is implemented by using multiplexers. The goals as power utilization and area. Different types of recomputing with encoded operands are used in viterbi decoder which is having the modified full adder. These methods uses the self-checking adder is used with two rail encoding [11]. We have reproduced the proposed mistake identification structures and acquired the consequences of our reenactments. Our proposed viterbi decoder and convolution encoder is implemented in Application Specific Integrated Circuits (ASIC). The proposed viterbi decoder is utilizes the low power consumption and area. In this both convolution encoder and viterbi decoder is designed. This is used in speech to text conversion application. The performance metrics are calculated for both viterbi decoder and convolution encoder is calculated.

Low Latency Viterbi Decoder

This section concentrate on branch metric commutation and the operations of add compare is neglected. To evacuate all redundancies an ideal methodology of adjusted paired gathering (BBG) [2] is utilized. The redundancies will in charge of longer deferral and additional unpredictability, the diverse ways share calculations. In conventional viterbi decoder the branch metric computations are performed sequentially. To provide equal or less latency look ahead based approach is used which is based on balanced binary grouping and it is highly efficient design. The balanced binary grouping is less complex than other methods. The branch metric precipitation is done in layer manner proposed for constraint length K and M-step look ahead technique. In VLSI architectures there is a chance of occurrence the transient and permanent faults. The erroneous outputs in the architectures will happen due to

the distinct and various stuck at faults. We will consider in all cases and simulate to reduce the power and area computations.

Modified Full Adder

We develop recomputing with rotating operands, though the activities revamped with various operands to recognize the blunders. In first time the operands are given as normally as inputs. The second step is known as recomputed step. In this recomputed method, the operands are rotated or shifted and given as inputs. These operands are encoded first and functioned after decoding. At output stage the correct results are obtained. To identify both the temporary and stable faults signature based schemes are used. To decrease the area and power intake, through the error recognition code like hamming codes, redundancy in time, adding insignificant area is above at the expense of higher total time is executed.

Csa and Pcsa Architectures

The parallelization of include and contrast activities and in ACS itself is done to expand the ACS engineering is quick. For accomplishing the low power utilization the quantity of states is multiplied and channel reaction is reached out by an additional piece To eliminate the parallelism the add operation is operated by compare process. For initial steps there is no compare operation for remaining steps there is a add operation followed by compare operation. Add and compare operations are done in sequential manner. The order of add compare operation is changed to compare-add which is used in Carry-Select-Add (CSA) unit. To reduce the power and area than the CSA unit we are using the recomputing with encoded operands with CSA. Pre computed carry-select-add (PCSA) is a speed optimized type. The PCSA architecture is used only for large K values and small M values. We use the signature based methods for both CSA and PCSA units. A single stuck at fault can cause the fault or error in the result. For the adder modules self-checking adders based on two rail encoding is used. The figs. 1 and 2 shows the CSA unit and PCSA unit. The CSA unit consist of two multiplexer, one sub tractor

and adder circuits. Previously the adder circuit consist off full adder but in proposed the adder circuit is replaced with modified full adder circuit for reducing the power consumption and area for architectures. In PCSA the original design consist of 2 muxs. The unique and duplicated multiplexers are compared by expending XOR gate.

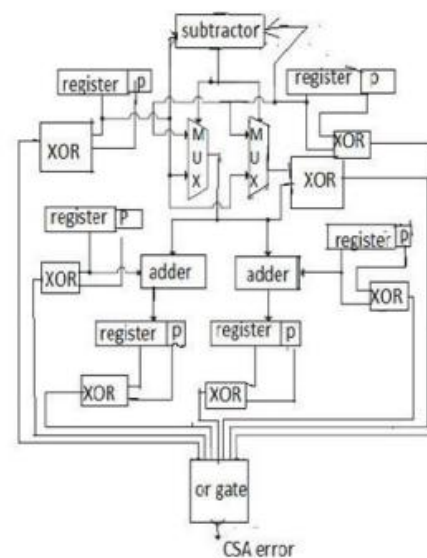


Fig 1.CSA architecture

The output of XOR gate connects to one of the inputs of OR gate. These input and output registers are integrated with other operands. To detect the faults in single bit, multiple bit or interleaved parity bit and cyclic redundancy. In both CSA and PCSA unit's adders are present. These adders are utilized in the self-checking adder. This self-checking adder is implemented by cascading the adders. The self-checking adder is consist of Two pair two rail checkers, multiplexers and adders are connected in parallel. The checker driven by using two pairs of inputs. This is done by using XNOR gates. The two outputs at the checker are formed are in two pair form. If there is a fault at the input of checker the output is not in two rail notation and the error indication flag is elevated to represent then there is a fault error in the system. Check are used. In the figs 1 and 2 the "p" represents the parity bits. To originate the error signal flag an OR gate units are necessary. The Error signal is displayed at the OR gate. If the Error signal is at CSA it can be indicated as CSA_error which is shown in

figures. If an error in at PCSA it can be indicated as PCSA_error.

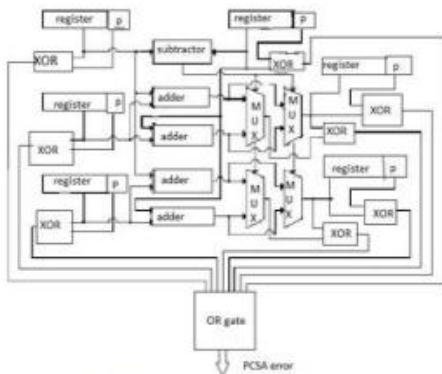


Fig 2 PCSA architecture

By using the self-checking adder, the adders in CSA and PCSA are implemented, shown in Fig. 3. To prefigure the entirety bits and supplemented esteems the altered full viper is utilized. The first estimation of convey in is utilized to choose the aggregate bits. The unique value of carry input is used for selecting the sum bits. The output of both the adders are the output of CSA unit. The output of adders are given as input to the DE multiplexer. By adding an additional adder [31] into the architecture, we calculate the performance. The self-checking adder is shown in Fig. 3. Now the inputs of circuits are given to the two pair two rail checker.

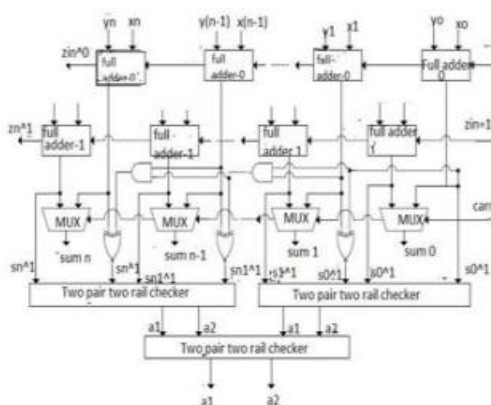


Fig 3 self-checking adder

By using AND gate, OR gate and NOR gate, Two pair 2-rail checker is implemented. The output of Two pair 2-rail checker is 00 or 11 then it indicates error in the data. If the output is 01 or 10 then there is no error in the circuit.

RERO with CSA and PCSA

Detection of errors in CSA and PCSA architectures are implemented by recomputed with rotated operands (RERO). The different types of RERO is shown in fig 4 and 5. In this method more number of cycles produced for completion. In the first time the original operands are given as inputs. For the second cycle the inputs are rotated and given as inputs. By using RERO, both CSA and PCSA consists of four inputs. These inputs are given to the multiplexer as in original form and rotated form. The multiplexer having the select lines. If the multiplexer is set to process first run then the original operands are operated. If the select line sets to second run the rotated operands are given as inputs. In CSA using RERO. By using comparator the inputs are fed into sub tractor whose selecting line is set. The comparator performs the compare and select operations. The add operation is performed by using adder circuit. The outputs of both the adders are the output of CSA unit. These outputs of adders are given as input to the DE multiplexer. The output of DE multiplexer is compared by using XOR gate. If there is an error then the error indication flag is raised.

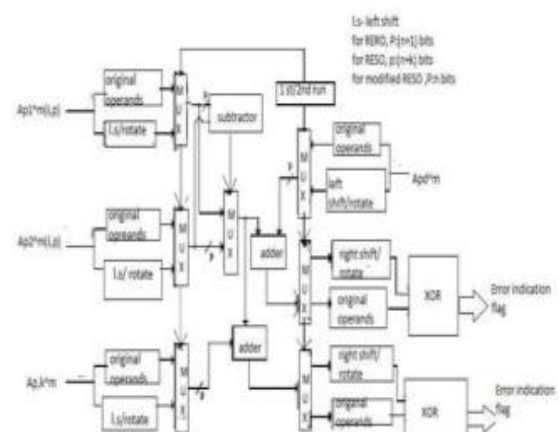


Fig 4 Recomputing with rotated operands using CSA

In PCSA circuit the first input is given to the comparator. The output of comparator acts as a select line to multiplexer. The other inputs are given to the adders circuits. The output of two multiplexer is the output of PCSA unit. If there is no error in the circuit then the output is passing through the separate DE multiplexers are used by the RESO, which performs the

recomputing with shifted operands. In this all, the operands are shifted from left or right by n bits. The first challenge in RERO is to reduce the iteration between most significant bit and least significant bit. The second challenge is to increase the performance from sub-pipelining as a part of ASIC implementations. For storing the decisions the Branch metric unit and Survivor path memory unit uses these registers. We use the parity bits to detect the errors.

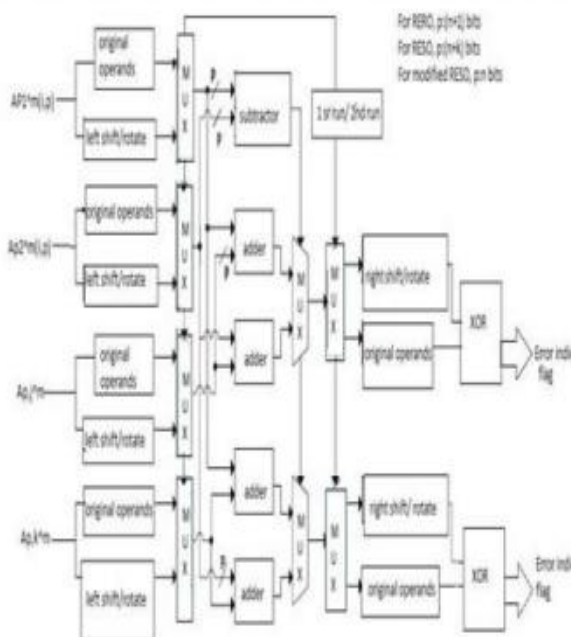


Fig 5 Recomputing with encoded operands using PCSA

The modified full adder is shown in Fig 6. By using multiplexer the full adder circuit is implemented. Due to this implementation the power consumption and area of Viterbi decoder is reduced than the precious architectures which is shown in results section. This modified full adder is placed in self-checking adder in place of adder circuit. This total modified self-checking adder is placed in CSA adder circuit. In general full adder circuit is the XOR gate is supplanted with multiplexer. The XOR gate consumes more power than the multiplexer. In the three inputs the one input and its complemented input is given to the multiplexer and the remaining inputs are given to the XOR gate. The full adder circuit is implemented using 4:1 multiplexers. Each 4:1 mux is composed of three 2:1 multiplexers.

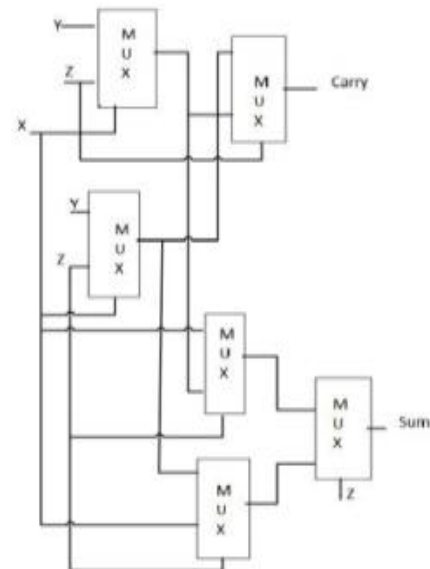
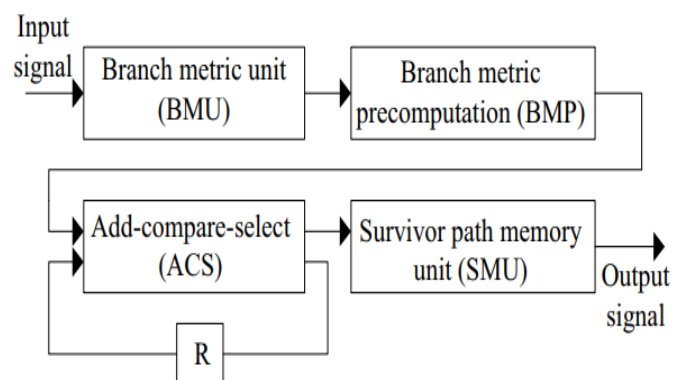


Fig 6 Modified Full Adder (MFA) circuit

Convolution encoder being a finite state machine used for encoding the given input data. This convolution encoder is implemented with modulo-2 operation and register. The register is used to store the modulo-2 operation outputs. The D-flip flop is used as register. The modulo-2 operation is performed by using XOR operation. In communication system the Viterbi decoder is used. In this work, the speech to text conversion is taken. The convolution encoder and viterbi decoder is applied to design the speech to text conversion application.

Results and Evaluation

Extension diagram



SIMULATION RESULTS

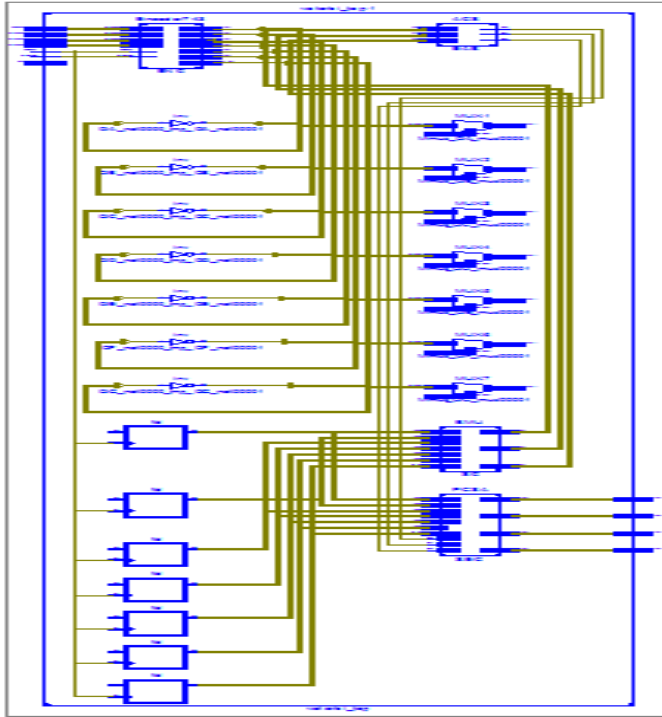


Fig 5.1 RTL Schematic

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	91	16640	0%
Number of Slice Flip Flops	112	33280	0%
Number of 4 input LUTs	150	33280	0%
Number of bonded IOBs	69	309	22%
Number of GCLKs	1	24	4%

Fig 5.2 DESIGN SUMMARY

Timing constraint: Default OFFSET OUT AFTER for Clock 'Clock'
 Total number of paths / destination ports: 6272 / 32

Offset: 12.868ns (Levels of Logic = 6)
 Source: OB_0 (FF)
 Destination: YC1<7> (PAD)
 Source Clock: Clock rising

Data Path: OB_0 to YC1<7>

Cell:in->out	fanout	Gate Delay	Net Delay	Logical Name (Net Name)
FD:C->Q	5	0.591	0.776	OB_0 (OB_0)
LUT2:I0->O	1	0.648	0.452	SC/Mxor_S0_0_xo<0>21 (N111)
LUT4:I2->O	1	0.648	0.563	EAB/e026 (EAB/e026)
LUT4:I0->O	36	0.648	1.406	EAB/e0206 (e0)
LUT4:I0->O	8	0.648	0.900	SEC/Mmux_YC31211 (SEC/N2)
LUT4:I0->O	1	0.648	0.420	SEC/Mmux_YC39 (YC3_2_OBUF)
OBUF:I->O		4.520		YC3_2_OBUF (YC3<2>)
Total		12.868ns	(8.351ns logic, 4.517ns route)	(64.9% logic, 35.1% route)

Fig 5.3 TIME SUMMARY



Fig 5.4 ENCODER OUTPUT

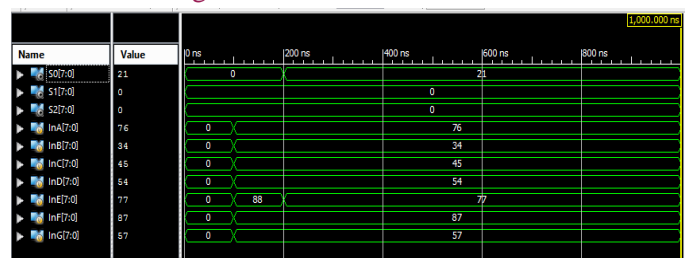


Fig 5.5 BMU OUTPUT

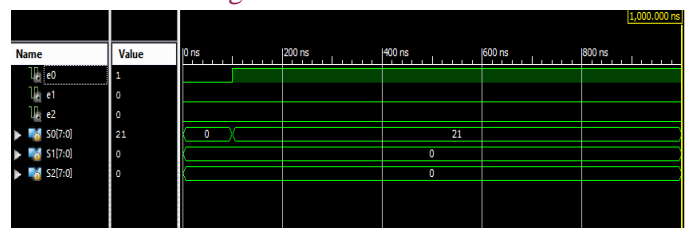


Fig 5.6 ACS OUTPUT

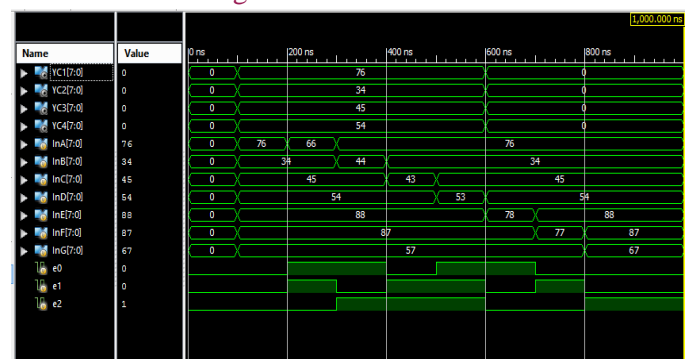


Fig 5.7 PCSA OUTPUT

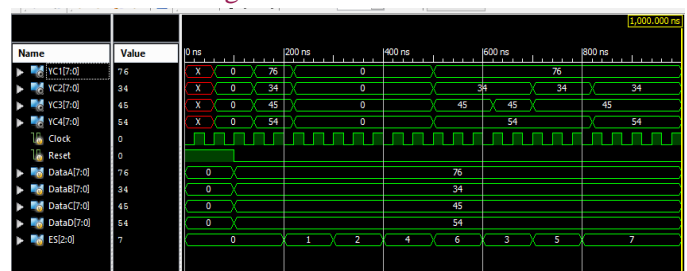


Fig 5.8 VITERBI OUTPUT

CONCLUSION AND FUTURE SCOPE

In this paper, the Viterbi decoder is designed by using modified full adder and convolution encoder is also designed. This can reduce the power consumption and area is reduced. It is designed by using IUS tool, RTL compiler and SOC encounter. In further the power consumption and area are reduced by using another adder circuits. We can use this in communication systems.

References

- [1]. A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," IEEE Trans. Inf. Theory, vol. IT-13, no. 2, pp.260-269, Apr. 1967.
- [2]. R. Liu and K. Parhi, "Low-latency low complexity architectures for viterbidecoders," IEEE Trans. Circuits syst. I. Reg. papers, vol. 56, no. 10, pp.2315-2324, Oct. 2009.
- [3]. K. K. Parhi, VLSI Digital Signal Processing Systems: Design and implementation. Hoboken, NJ, USA: Wiley, 1999.
- [4]. G. Fettweis and H. Meyr, "parallel Viterbi algorithm implementation: Braking the ACS-bottleneck," IEEE Trans. Commun., vol. 37, no.8, pp.785-790, Aug. 1989.
- [5]. V. Gierenz, O. Weiss, T. Noll, I. Carew, Ashely, and R. Karabed, "A 550mb/s radix-4 bit-level pipelined 16-state 0.25um CMOS viterbi decoder", in proc. IEEE Int. Conf. Appl-specific Syst. Archit. Process. Jul. 2000, pp.195-201.
- [6]. P.J. Black and T. H. Meng. "A 140-mb/s, 32-state, radix-4 viterbi decoder." IEEE J. Solid-State Circuits, vol. 27, no. 12, pp. 1877- 1885.
- [7]. T. Gansen, and T. Noll, "implementation of scalable power and area efficient high-throughput viterbi decoder". IEEE J. SolidState Circuits, vol. 37 no. 7, pp. 941-948, 2002.
- [8]. A. Yenug and J. Rabaey: "A 210 Mb/s radix-4 bit-level pipelined viterbi decoder" in Proc. IEEE Conf. Int. Solid-State Circuits, Feb. 1995, pp. 88-89.
- [9]. K. Parhi, "An improved pipelined MSB-first add-compare select unit structures for viterbidecoders," IEEE Trans. Circuits Syst. I, Reg. papers, vol. 51, no.3. pp. 504-511, Mar. 2004.
- [10]. J. J. Kong and K. K. Parhi, "Low-latency architectures for high-throughput rate viterbidecoders," IEEE Trans. Very Large Scale Integer.(VLSI) Syst., vol. 12. No.6, pp. 642-651, Jun. 2004.