Area Efficient VLSI Architecture for Reversible Radix-2 FFT Algorithm using Folding Technique and Reversible Gate

1 Mukkamalla Hussainbee, 2Dr.G.LakshmiNarayana , 3Dr.M.M.Raghavendra, 4S.Mohandas
1M.Tech Student, 23Professor, 4Associate Professor
DEPT OF ECE
SVR Engineering College, Nandyal

Abstract: Reversible gates have accessible itself as a famous methodology which has a crucial position in Quantum levels. Nanotechnology integrated circuits hypothetically function at ultra speed & drain very less power. Investigation had done in this paper and viterbi algorithm and focus on idea of reversible gates to crack the usual speed-power relationship. The Viterbi FFT based encoder and FFT based decoder are usually useful to a number of responsive handling models together with decoding of convolutional codes utilized in communication systems. In this manuscript, well-organized error recognition scheme for structures based on low-complexity, low-latency.

INTRODUCTION

1.1VITERBI ALGORITHM

The Viterbi algorithm was introduced in 1967 as an efficient method for decoding convolutional codes [1], widely used in communication systems [2].This algorithm is utilized for decoding the codes used in various applications including satellite communication, cellular, and radio relay. It has proven to be an effective solution for a lot of problems related to digital estimation. Moreover, the Viterbi FFT based decoder has practical use in implementations of high-speed (5 to 10 Gb/s) serializer-deserializers (SERDESs) which have critical latency con-straints. SERDESs can be further used in local area and synchronous optical networks of 10 Gb/s. Furthermore, they are used in magnetic or optical storage systems such as hard disk drive or digital video disk [3].

The Viterbi algorithm process is similar to finding the most-likely sequence of states, resulting in sequence of observed events and, thus, boasts of high efficiency as it consists of finite number of possible states [4–7]. It is an effective implementation of a discrete-time finite state Markov process perceived in memory less noise and optimality can be achieved by following the maximum-likelihood criteria [8]. It helps in tracking the stochastic process state using an optimum recursive method which helps in the analysis and implementation [9, 10].

A top-level architecture for Viterbi FFT based decoders is shown in Fig. 1.1. As seen in this figure, Viterbi FFT based decoders are composed of three major components: branch metric unit (BMU), add-compare-select (ACS) unit, and survivor path memory unit (SMU). BMU generates the metrics corre-sponding to the binary trellis depending on the received signal, which is given as input to ACS which, then, updates the path metrics. The survival path is updated for all the states and is stored in the additional memory. SMU is responsible for managing the survival paths and giv-ing out the decoded data as output.

BMU and SMU units happen to be purely forward logic. ACS recursion consists of feedback loops; hence, its speed is limited by the iteration bound [11]. Hence, the ACS unit becomes the speed bottleneck for the system. M-step look-ahead technique can be used to break the iteration bound of the Viterbi FFT based decoder of constraint length K [12–18]. A look-ahead technique can combine several trellis steps into one trellis step, and if M > K, then throughput can be increased by pipelining the ACS architecture, which helps in solving the problem of iteration bound, and is frequently used in high-speed communication systems.

Branch metric precomputation (BMP) which is in the front end of ACS is resulted due to the look-ahead technique and it dominates the overall complexity and latency for deep look-ahead architectures. BMP consists of pipelined registers between every two consecutive steps and combines binary trellis of multiple-steps into a single complex trellis of one-step. BMP dominates the overall latency and complexity for deep look-ahead architectures. Before the saturation of the trellis, only add operation is needed. After the saturation of the trellis, add operation is followed by compare operation where the parallel paths consisting of less metrics are discarded as they are considered unnecessary.

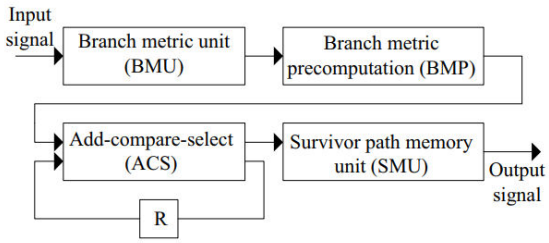

Figure 1: Viterbi FFT based decoder block diagram.

Although Viterbi algorithm architectures are used commonly in decoding convolutional codes, in the presence of very-large-scale integration (VLSI) defects, erroneous outputs can occur which degrade the accuracy in decoding of convolutional codes.

1.2 FAULT DIAGNOSIS

A fault in a system can be defined as a deviation from the expected working of the system which can be due to a defect of some components of the circuit. They can be temporary or permanent. Permanent faults are called as Solid or Hard faults and can result due to the wear-ing out or breaking of components. Temporary faults can be referred to as soft faults and these faults can be classified as intermiitent or transient as it occurs only at certain intervals of time. An intermittent fault occurs when the component is developing a permanent fault. A transient fault can result due to some external disturbance like power supply fluctuations. Depending upon the effect of faults, they can be classified as parametric or logical.

1.3 OBJECTIVE

In this thesis, we explore two approaches for two variants of sub-parts in the Viterbi algo-rithm. Specifically, we note that both area/power consumption and throughput/efficiency degradations need to be minimized with respect to the proposed approaches; thus, we ex-plore signature-based approaches resulting in better efficiency at the cost of area/power con-sumption, and recomputing with encoded operands to achieve permanent and transient er-ror detection. For detecting the errors in the ACS unit, we utilize three variants, i.e., re-computing with shifted operands (RESO) [47], proposed modified RESO which has slightly less fault resilience effectiveness; yet, lower induced overhead, and recomputing with ro-tated operands (RERO) [48]. Our architectures also include hardware redundancy techniques through signature-based detection. Specifically for the adder components, we utilize a num-ber of variants of self-checking based on two-rail encoding. The architectures to which the schemes have been applied consist of two types of low-latency and low-complexity structures of Viterbi FFT based decoders [3] with slight modifications.

We summarize the contributions of this thesis as follows:

We propose error detection methods for the modified Viterbi FFT based decoder with the consider-ation of objectives in terms of performance metrics and reliability. The error detection approaches along with the modifications help achieving high error coverage and through the proposed improvements, performance boost can be achieved. Variants of recomput-ing with encoded operands on a number of architectures within the modified Viterbi FFT based decoder as well as signature-based approaches (including modified self-checking based on two-rail encoding) are presented as well. The mechanisms for making the proposed structures immune to faults have not been presented before.

We have extensively simulated the proposed error detection architectures and the ob-tained results help in benchmarking the error coverage. The results of our simulation show that the reliability of the proposed architecture can be ensure

II.LITERATURE SURVEY

2.1EXISTING METHOD

This section focuses only on branch metric computation, leaving aside the operations of compare-and-discard. An optimal approach of BBG is taken into consideration in order to remove all redundancies which are usually responsible for longer delay and extra complexity, since various paths share common computations. Branch metrics computation is said to be carried out sequentially for a conventional Viterbi FFT based decoder. When two consecutive binary-trellis steps are combined, for each state, there are two incoming and two outgoing branches, and the computational complexity is $4 \times N$. As the results do not depend on the order of the trellis combination, the way the trellis steps are grouped and combined helps in determining the computational complexity. The combination in a backward nested procedure

can be ex-plained as follows. The main M-step trellises are divided into two groups consisting of m_0 and m_1 trellis steps. The binary decomposition on each subgroup goes on till it becomes a single trellis step. The decomposition helps in removing maximum possible redundancy and, thus, helps achieve minimum delay and complexity. Finally, it can be verified that the complexities involved in the BBG approach are less as compared to the ones in the intuitive approach.

2.1.1 Look-ahead-based Low-Latency Architectures

This approach is a highly-efficient design approach based on the BBG scheme for a general M which provides less or equal latency, and also has much less complexity compared to other existing architectures [3]. For constraint length K and M-step look-ahead, the execution of BMP is done in a layered manner. An M-step trellis is a bigger group consisting of $\frac{M}{K}$ sub-groups with a trellis of K-step. Thus, the total numbers of P1 processors needed are $\frac{M}{K}$ and each P1 is responsible for computing K-step trellises. Accordingly, we have the complexities and latencies of P1 and P2 as $Comp._{P1} = N(\sum_{i=2}^{k} 2^i) + N^2$, $Comp._{P2} = N^2(N-1) + N^3$, and $Lat._{P1,P2} = K$, where $N = 2^{k-1}$ is the number of trellis states. For P1 processors, the complexity of add operation is $N\sum^{k} = 2^i$ and that of the "compare" operation is N^2. Similarly, for P2 i2processors, the complexity of add operation is $N^2(N-1)$ and that of the compare operation is N^3. For both P1 and P2 processors, the latency is same, i.e., K; however, the complexity of P2 is larger than that of P1. As the BBG approach is very efficient in computing the branch metrics, more operations of trellis combination can be allotted into BBG-based P1 processors in order to reduce the number of P2 processors as they are expensive in terms of complexity. The trellis Steps L, which is computed in the P1 processors, has the constraint of being less than $2 \times K$ in order to make sure that the latency feature is not lost. The number of groups N_g can be determined by $Ng = 2^{\lfloor \log2(\frac{M}{K}) \rfloor}$.

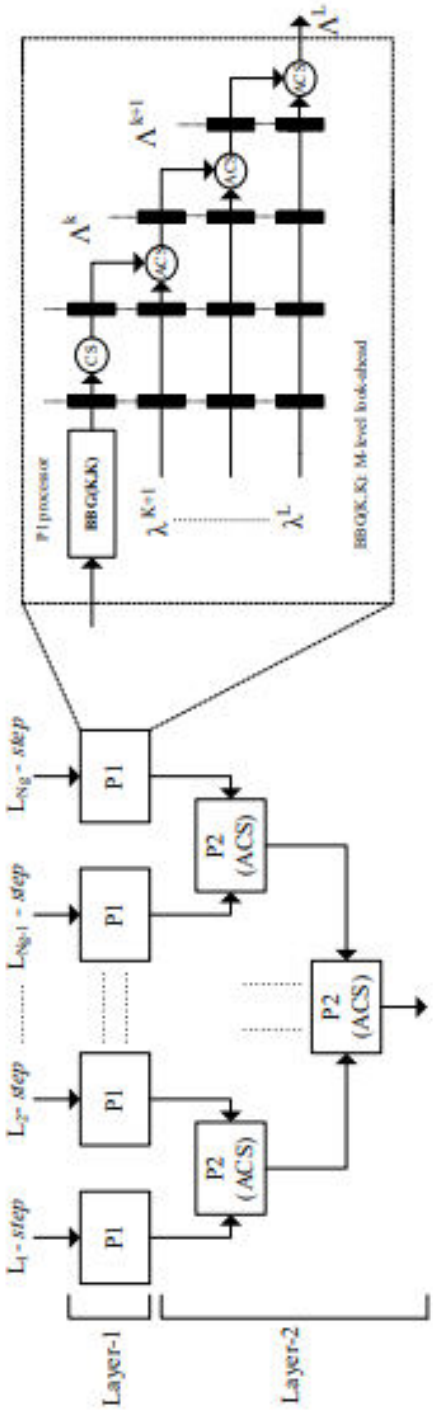

Figure 2: Overall layered structure including the P1 processor architecture.

III.PROPOSED METHOD

Fast Fourier Transform (FFT) calculation is broadly utilized in many flag preparing and correspondence frameworks. The FFT calculation is analyzed one of the rudimentary calculations in numerous DSP ventures. At present, FFT is the imperative building hinder for the portable

correspondences, especially for the symmetrical recurrence division multiplexing based correspondence frameworks, for example, advanced video furthermore, sound telecom, uneven advanced supporter circle (ADSL) [1]. Be that as it may, a period multifaceted nature of DFT is O (n2) where a period multifaceted nature of FFT is O(NlogrN). The FFT was found by Cooley and Tukey to effectively accelerate the calculation time. FFT‟s are registered in O (logrN) stages, where N is the length of the change and r is the radix of the FFT breaking down. Here N information words are perused and compose by each stage in FFT. T plan of FFT processor is for the most part having four sorts of engineering: Single, double memory, cluster design and pipeline engineering. The FFT designs can be partitioned into two classes: memory-based and pipelined designs [2, 3]. Memory-based models involve a butterfly unit and a certain number of memory obstructs for giving minimal effort plans. In any case, it is troublesome to accomplish ongoing preparing at the low clock recurrence. On the other hand, a pipelined design include different stages to give higher throughput at the expense of something else equipment. Amidst, the best possible FFT estimate shifts for various applications. For precedent, the size can be 128, 256, 512, 1024 or 2048 for WiMAX applications and 256, 512, 1024 or 2048 for DAB frameworks [4]. Henceforth, for a particular application, the asked for FFT center ought to be all around arranged to meet its very own exceptional necessities. The FFT is a run of the mill computation where the memory get to seriously and the high parallelism is required. FFT calculation ought to have pipelined design and parallel design, be ordinary and measured. At calculation level, it should come to the multiplicative multifaceted nature as low as practical. At the structural dimension, utilize the deferral – criticism buffering methodology to lessen the memory measure. It ought to have measured and ordinary modules, neighborhood directing and low control intricacy [5]. FFT is used to change over time space flag to recurrence area flag. It is utilized to figure the DFT adequately. To meet the superior, high speed and ongoing prerequisites of present day applications, equipment fashioners have continuously attempted to perform proficient structures for the estimation of the FFT. The pipelined equipment models are generally utilized, in light of the fact that they give high throughputs and low latencies appropriate for continuous, just as a sensibly low region what's more, control utilization. For the most part DIT manages the info and yield backward arrangement and ordinary grouping separately, while DIF manages information and yield in typical succession and turn around arrangement individually. Just DIT calculation will be contemplated.

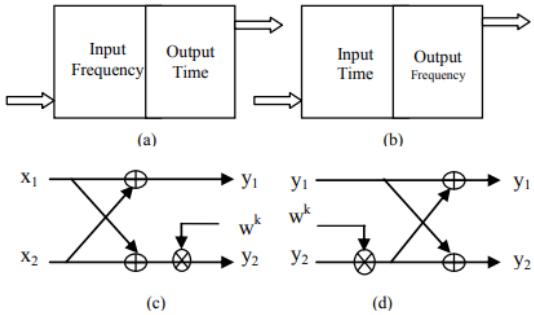

Figure 3: Time and Frequency Signal

II. FAST FOURIER TRANSFORM

There are two sorts concerning FFT calculation formulated by Cooley and Tukey - Decimation-in-Time calculation (DIT) and Decimation-in-Frequency calculation (DIF). The calculation of an arrangement of N-point can be acquired by methods for a double methodology. The info succession x(n) of size 'N' is disintegrated into tests of odd and even and the comparing sub-groupings f1(n) and f2(n) are given by)

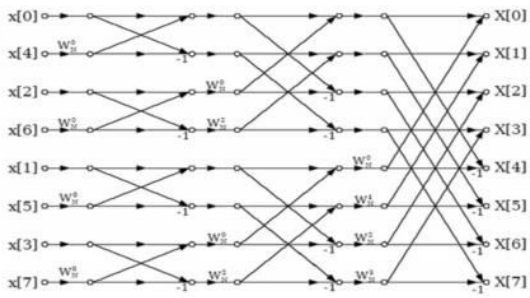

Figure 4: 8-point DIT-FFT Radix-2 Termite

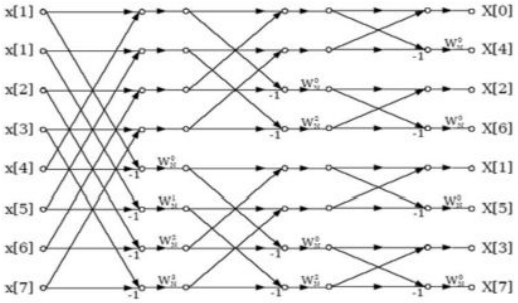

Figure 5: 8-point DIF-FFT Radix-2 Termite

In represent figure 2, demonstrate the termite of radix-2 DIT FFT calculation. In this figure we utilized eight sources of info and eight yields. If there should be an occurrence of DIT the info test is utilized piece inversion arrange while the yield of DIT FFT coadjuvant is produced in normal request. In represent Figure 3, demonstrate the termite of radix-2 DIF FFT calculation. In the event of DIF the information test is utilized in common request while the yield of DIF FFT coadjuvant is produced in bit turned around request

There are several journals discussed regarding Viterbi FFT based decoder [1] realization in VLSI environment. paker, et al. [2] talk about a VLSI method for area examination for a hard-soft decision based FFT based decoder. They portray any calculations that be measured using new ACS unit. Wang, et al. [3] have projected a low-area consumed FFT based decoder with the Transpose-algorithm for trellis approach and trellis modulation. Then, A low-power method is proposed since a FFT based decoder has soaring power dissipation in trellis methods [4]. in addition to power dissipation, authors have studied on speed and delay evaluation in [5] and hardware efficiency and area utilization in [6]. All these literatures are focused to develop veterbi communication system by using conventional CMOS technology based basic gates, as the trellis method is the major approach for decoding, for reducing the number of paths and path delays, the reversible logic is preferable. As it consumes low quantum cost, low area, less power consumption and fewer delays compared to other literatures.

III. PROPOSED METHOD

The expense and defer figurings are indistinguishable to the 4- bitb snake/sub tractor in figure 9. We have design 4-bit full sub-tractor/adder with the help of DKG Gate. If the fourth input of the DKG Gate is '0' then output of the DKG Gate as a adder and fourth input of the DKG Gate is '1' then output of the DKG Gate as a sub-tractor

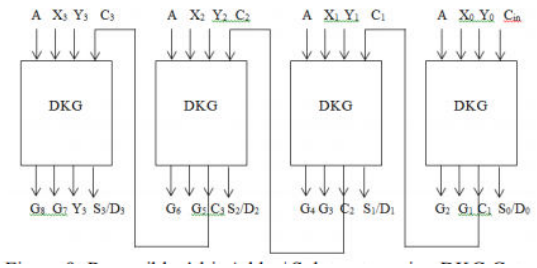

Figure 6: Reversible 4-bit Adder/ Sub tractor using DKG Gate

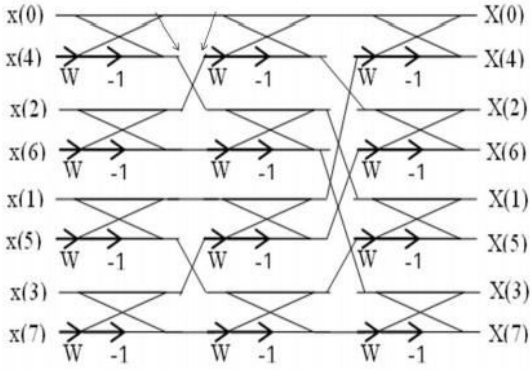

Figure 7: DIT FFT Radix-2 Termite algorithm

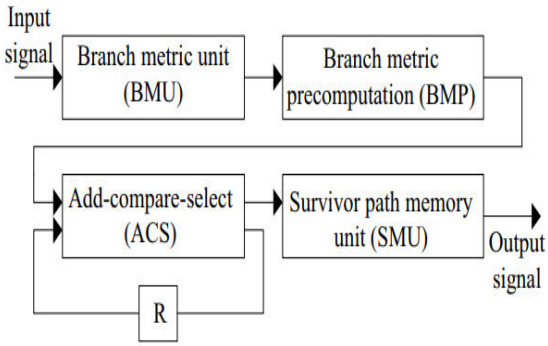

Figure 8: Viterbi FFT based decoder

A finest structure for Viterbi FFT based decoders is shown in Fig.4. As set up in figure, Viterbi FFT based decoders are prepared out of three important segments: Add compare select unit (ACS), branch metric unit (BMU), and survivor path memory unit (SMU). BMU creates the path to data levels matching to paired encoded trellis operands, thus syndrome of those particular inputs are calculated and applied to ACS unit. By comparing the each and every bit position with parity check matrix values and stores the compared results in the Register unit R again the stored values will be further processed for location detection by using feedback mechanisms. The error locations are identified by using the branch units generated in BMP. After finding the error location for correction of those values

SMU unit will be useful. By using error identified path, its path metric will be recalculated and error corrected, thus final decoded data will be generated.

Figure 5: convolutional FFT based encoder

Generally, viterbi FFT based decoder used to decode the convolutional encodes operands. Here, D0, D1, D2 and D3 are the input data sources, and corresponding outputs are Out0 to out 6. For generating the outputs, convolution FFT based encoder utilizes the generator matrix G, it consists of identity matrix and parity symbols. The parity symbols P1, P2, P3 are user defined, accordingly connections between reversible HNG gate has done. Output encoded frame format is out= [P1, P2, P3, D0, D1, D2, D3];

P1=XOR (D0, D1, D2);

P2=XOR (D0, D1, D3);

P2=XOR (D0, D2, D3);

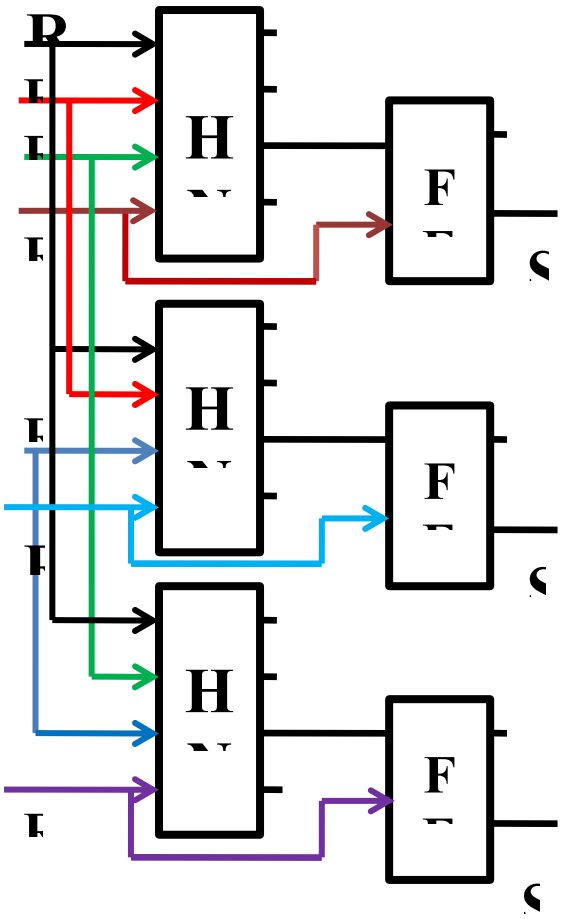

Figure 9: Branch metric unit and precomputation

After successful completion of encoding operation, the encoded codeword's are transmitted into channel. Generally the channel consists of lots of Gaussian noise, random noise and AWGN noise. The encoded codeword's will be affected by this noise, thus error will be added. Thus, the major task of the FFT based decoder is to remove the error from encoded data instead of decoding it. For this purpose the BMU block in the viterbi FFT based decoder is useful to check the error status. It will monitors the every codeword and if error presents, it will identify the type of noise added then it will alerts the ACS block, if there is no error BMU block simply decodes the codeword's by using branch metrics. These branch metrics are formed by the multiple combination parity symbols with their low to high probabilities.

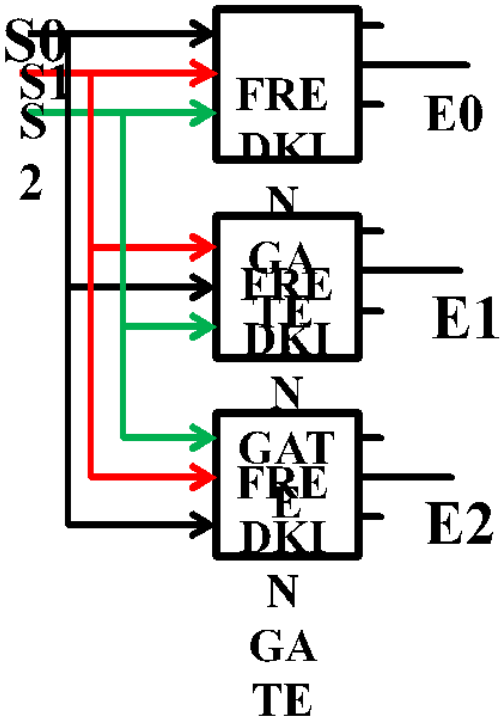

Figure 10: Add compare and Selection unit

After identifying the error and noise type status in BMU unit, it is necessity to locate the error in codeword. Thus, the error location identification will be done by the ACS unit. Generally, the ACS contains the trellis methodology to identify the error location. But here, as we are using the reversible methodology, here approximation of BMU syndromes methodology has been implemented. By using the minimum of two between each BMU metric, the comparison of coefficients has done. These compared coefficients will then apply to revisable fredikin gates for approximate addition.

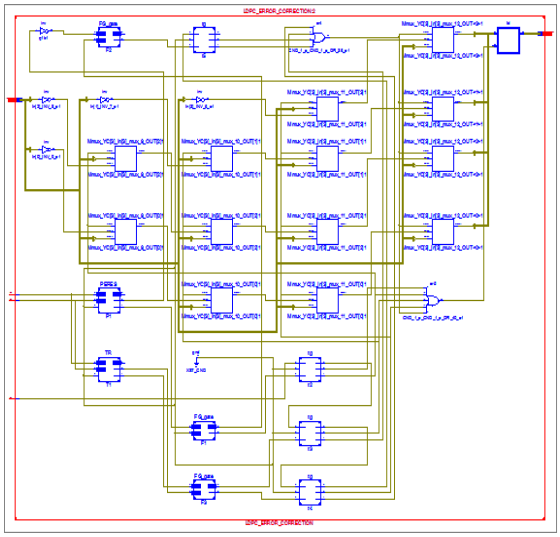

Figure 11: Survival path memory unit

The final error location identification and error correction done in the SMU block, here each and every path will be monitored with respect to the approximation coefficients generated in ACS block. This SMU unit is reconfigurable because for the different types of noises, the error will be altered, According to that noise SMU also reconfigures itself. Here prioritized error elimination metrics will be generated in each path, so they termed as path metrics also. By using the Round-Robin procedure, by comparing one metric to all other paths with equal priority, survived path (error free) will be identified. By performing the bit modifications in these survived paths decoded error free data will generates.

IV.SIMULATION RESULTS

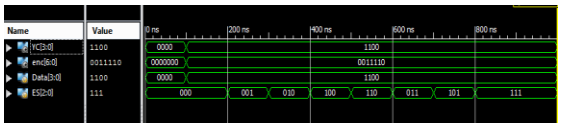

Figure 9: viterbi communication system

Here data is the original input data, and ES is the manual error syndrome input and enc is the final encoded operand. YC is the decoded error free output data so it is same as input data.

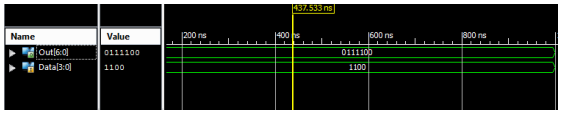

Figure 12: FFT based encoder output

Here data is the original input data and out is the final encoded operand.

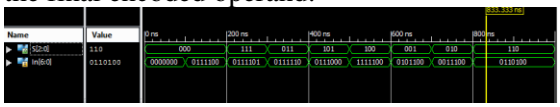

Figure 13: BMU output

Here In is the original encoded operand input data and S is the final BMU error coefficients.

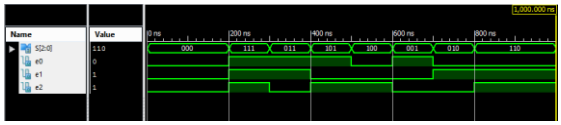

Figure 14: ACS output

Here S is the BMU error coefficients input data and E0, E1 and E2 are the final ACS prioritized error coefficients.

Figure 15: SMU output

Here In is the original encoded operand input data, E0, E1 and E2 is the ACS prioritized error coefficients inputs. YC is the decoded error free output data so it is same as input data.

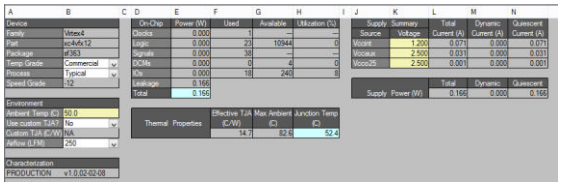

Figure 16: Power consumption

The above result represents the power consumed by using the Xilinx ISE software. The consumed power is 0.166uw.

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	13	5472	0%
Number of 4 input LUTs	23	10944	0%
Number of bonded IOBs	18	240	7%

Figure 17: Design summary

The above result represents the synthesis implementation by using the Xilinx ISE software. From the above table, it is observed that only 23 look up tables are used out of available 10944. It indicates very less area is used for the proposed design.

```
                                    Gate     Net
Cell:in->out      fanout   Delay    Delay  Logical Name (Net Name)
----------------------------------------  ------------
 IBUF:I->O             7   0.754   0.596  ES_0_IBUF (ES_0_IBUF)
 LUT4:I0->O           13   0.147   0.637  SC/f3/Mxor_q_Result1 (S<2>)
 LUT4:I1->O            1   0.147   0.000  SEC/YC_1_mux00001 (SEC/YC_1_mux0000)
 LDCP:D                    0.017          SEC/YC_1
----------------------------------------
Total                      2.298ns (1.065ns logic, 1.233ns route)
                                   (46.3% logic, 53.7% route)
```

Figure 18: Time summary

The above result represents the time consumed such as path delays by using the Xilinx ISE software. The consumed path delay is 2.298ns.

V.CONCLUSION

In this thesis, we presented fault diagnosis models for the CSA and PCSA units of lowcomplexity and low-latency Viterbi FFT based decoder. The simulation results for the proposed methods of RESO, RERO, modified RESO, parity and self-checking adder based

designs for both CSA and PCSA units show very high fault coverage (almost 100 percent) for the randomly distributed injected faults. The proposed architectures has been successfully implemented on Xilinx Virtex-6 Family and also by using the 32nm library using Synopsys Design Compiler for the ASIC implementation. Also, the ASIC and FPGA implementation results show that overheads obtained are acceptable. Thus the proposed models are reliable and efficient.

Future scope

This thesis work focussed on performing the fault detection on the CSA unit and the PCSA unit. The work can be extended by performing fault detection for the different binary-trellis groups using the parity registers and duplicating the adders. Recomputing with encoded operands and unified signature-based scheme were used to detect faults in this work. In future, the proposed architectures can be tested with other fault detection techniques like off-line error detection schemes and roving fault detection method.

REFERENCES

[1] Massoud Pedram, "Power minimization in ic design: Principles and applications," ACM Trans. Des. Autom. Electron. Syst., vol. 1, no. 1, pp. 3–56, Jan. 1996.

[2] Qing Wu, M. Pedram, and Xunwei Wu, "Clock-gating and its application to low power design of sequential circuits," Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, vol. 47, no. 3, pp. 415–420, Mar 2000.

[3] G.E. Tellez, A. Farrahi, and M. Sarrafzadeh, "Activity-driven clock design for low power circuits," in Computer-Aided Design, 1995. ICCAD-95. Digest of Technical Papers., 1995 IEEE/ACM International Conference on, Nov 1995, pp. 62–65.

[4] E. Lee and A. Sangiovanni-Vincentelli, "Comparing models of computation," in Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design. IEEE Computer Society, 1997, pp. 234–241.

[5] Gilles Kahn, "The Semantics of Simple Language for Parallel Programming," in IFIP Congress, 1974, pp. 471–475.

[6] Edward A. Lee and David G. Messerschmitt, "Static scheduling of synchronous data flow programs for digital signal processing," IEEE Trans. Comput., vol. 36, no. 1, pp. 24–35, 1987.

[7] E.A. Lee and T.M. Parks, "Dataflow process networks," Proceedings of the IEEE, vol. 83, no. 5, pp. 773 –801, may 1995.

[8] Syed Suhaib, Deepak Mathaikutty, and Sandeep Shukla, "Dataflow architectures for GALS," Electronic Notes in Theoretical Computer Science, vol. 200, no. 1, pp. 33–50, 2008.

[9] Tzyh-Yung Wuu and Sarma B. K. Vrudhula, "Synthesis of asynchronous systems from data flow specification," Research Report ISI/RR-93-366, University of Southern California, Information Sciences Institute, Dec 1993.

[10] Behnam Ghavami and Hossein Pedram, "High performance asynchronous design flow using a novel static performance analysis method," Comput. Electr. Eng., vol. 35, no. 6, pp. 920–941, Nov. 2009.

[11] S.C. Brunet, E. Bezati, C. Alberti, M. Mattavelli, E. Amaldi, and J.W. Janneck, "Partitioning and optimization of high level stream applications for multi clock domain architectures," in Signal Processing Systems (SiPS), 2013 IEEE Workshop on, Oct 2013, pp. 177–182.

[12] Simone Casale-Brunet, Analysis and optimization of dynamic dataflow programs, Ph.D. thesis, STI, Lausanne, 2015.

[13] Endri Bezati, High-level synthesis of dataflow programs for heterogeneous platforms, Ph.D. thesis, STI, Lausanne, 2015.

[14] S. Casale-Brunet, M. Mattavelli, and J.W. Janneck, "Buffer optimization based on critical path analysis of a dataflow program design," in Circuits and Systems (ISCAS), 2013 IEEE International Symposium on, May 2013, pp. 1384–1387.

[15] Xilinx, Analysis of Power Savings from Intelligent Clock Gating, August 2012, XAPP790.

[16] “Open RVC-CAL Applications,” 2014, http://github.com/orcc/orc-apps, accessed 25-February-2014]. [17] M. Canale, S. Casale-Brunet, E. Bezati, M. Mattavelli, and J. Janneck, “Dataflow programs analysis and optimization using model predictive control techniques,” Journal of Signal Processing Systems, pp. 1–11, 2015.