

Analysis of Wave-Pipelined Architecture of ARA-LDPC Codes

M. Anbuselvi
SSN college of engineering
Rajiv Gandhi Salai
Tamilnadu,India

S. Salivahanan
SSN college of engineering
Rajiv Gandhi Salai
Tamilnadu,India

P. Saravanan
SSN college of engineering
Rajiv Gandhi Salai
Tamilnadu,India

ABSTRACT

Accumulate Repeat Accumulate -Low Density Parity Check Codes (ARA-LDPC) are a kind of linear block codes having self error correcting capabilities. It is used to transmit messages efficiently over noisy transmission channels. Due to this, the probability of information loss can be made as small as possible. The inherent feature of these codes is that the data transmission rate approaches Shannon limit, which is the theoretical maximum data transfer rate for a particular noise level. The ARA codes have a fast encoder structure and follow a protograph representation for high speed iterative decoding. Because of these unique features, the ARA-LDPC codes are the most suitable for deep space applications. In this project, an architectural model of ARA-LDPC encoder is designed and simulated in Modelsim, synthesized using Xilinx ISE, for sequential, pipelined and wave pipelined architectures and the performance is analyzed in SYNOPSIS and XILINX environments. The most efficient wave pipelined architecture is implemented in Spartan 3E FPGA for a block size of 1024 bits.

General Terms

LDPC decoder architecture, ARA codes

Keywords

ARA-LDPC, decoder architectures, wave-pipelining.

1. INTRODUCTION

Low density parity check codes are a kind of linear block codes having self error correcting capabilities. The researchers revisited the LDPC codes and extended the work of Gallager in 1993. This coding method helps to transmit messages over noisy transmission channels efficiently. Due to this, the probability of information loss can be made as small as possible.

The important feature of LDPC codes is that the data transmission rate approaches Shannon limit, which is the theoretical maximum data transfer rate of the channel for particular noise level [1]. It is mainly used in bandwidth-limited communication link where designers seek to achieve maximal information transfer, in presence of data-corrupting noise. Furthermore LDPC codes are more suited for implementations that have inherent parallelism. The parity check matrix for LDPC codes is preferred to be sparse i.e. a parity check matrix with less number of nonzero elements in each row and column. Hence the

decoding algorithms are simple and relatively easier to implement. However the encoder complexity is high. Thus, we go for Accumulate Repeat Accumulate (ARA) codes which have fast encoder structure [2]

2. SUBCLASS OF LDPC

2.1 RA codes

Repeat Accumulate (RA) code is a category of Low density parity check codes with excellent error-correction capability and inherently parallelizable decoding scheme. RA codes use fixed repetition for input bits. Moreover, the RA code has a “turbo-like” structure with two constitute codes being a repetition code and a convolutional code, the encoding of which may thereby be implemented with linear complexity. As a result, RA code has been accepted as one of the most successful LDPC codes and is widely used in various communication systems. For RA codes, a parity-check matrix with fully random interleavers may obtain good error-correcting performance, while the hardware implementation complexity becomes very high.

2.2 Punctured RA codes

The process of removing some of the parity bits after encoding with an error-correction code is called puncturing [3]. The effect of this technique is in par with an error-correction code having higher rate, or less redundancy. The puncturing technique increases the system’s flexibility considerably, without significantly increasing its complexity. Rate-1/2 RA code with lower threshold can be realized by puncturing the lower rate RA codes, with the repetition of 3 or higher, provided that the systematic bits are transmitted through the channel [4].

2.3 IRA and ARA codes

An irregular punctured RA code is constructed by using irregular repetition of the input bits. Irregular Repeat Accumulate (IRA) codes are an appealing choice because of its simple encoder and their performance is quite comparative with that of Turbo codes and LDPCs. They can also be decoded with a very low-complexity iterative decoding scheme.[3,4].

The ARA (Accumulate Repeat Accumulate) codes are a subclass of LDPC codes. They have a protograph or projected graph representation which paves way for high speed iterative decoding. Also, the number of iterations required for decoding is less. They have fast encoder structure; hence the encoder complexity is reduced.

The ARA code is a precoded repeat accumulates (RA) code with puncturing. It can also be a precoded irregular repeat accumulates (IRA) code, where precoder is a simple accumulator [3]. The amount of performance improvement with the use of precoder is called precoding gain. Low thresholds with the constraints on maximum bit node degree can be achieved with ARA codes, compared to RA, IRA, or unstructured irregular LDPC codes. ARA codes provide a **near Shannon limit performance** when compared to the other subclasses of LDPC codes. [5]

3. ARA ENCODER

The ARA code is a precoded Repeat Accumulate (RA) code with puncturing. It can also be a precoded Irregular Repeat Accumulate (IRA) code, where simply an accumulator is chosen as the precoder. The precoder implies single memory recursive feedback with binary addition, that is, performs EX-OR operation between the bits. The amount of performance improvement with the presence of precoder is known as the precoding gain. Puncturing is the technique of removing certain parity bits in a periodic manner. This technique increases the flexibility of the system without significantly increasing its complexity [6]. Here, predefined puncturing patterns – X0, 0X, 00X are used where 0's indicate the puncturing positions. Repeat 3 blocks involves the fixed repetition of each bit, three times. Interleaving is the process of interchanging the bit positions. Improvement in performance occurs because the interleaving avoids the occurrence of burst errors [7].

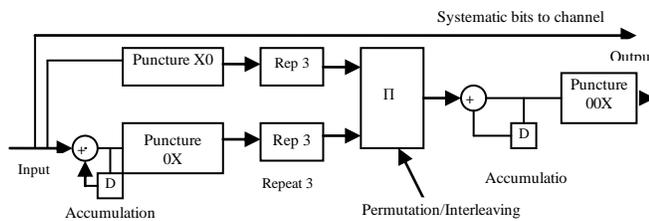


Fig 1: ARA encoder

The serial concatenation consists of outer accumulator, middle repetition, and inner accumulator, we call it Accumulate-Repeat-Accumulate (ARA) encoder. Low thresholds with the constraints on maximum bit node degree can be achieved with ARA codes, compared to RA, IRA, or unstructured irregular LDPC codes. [8]. Here a rate half ARA encoder is chosen, that is, if the size of the input message bits is N, then the final codeword obtained is of size 2N[9].

4. ARA DECODER

ARA-LDPC is an error correcting code. Thus its decoding process is very important. When a block of encoded message bits are transmitted through a channel, few of its bits are liable to be corrupted due to the noise present in the channel. This may result in a logic '1' being misinterpreted as logic '0' or vice versa. The decoder has to correct such errors and decode the correct message that was transmitted. The parity check matrix and the Tanner graphs are used for this purpose. The H matrix (parity

check matrix) is essential for decoding any class of block code. In general, the G matrix (generator matrix) is used at the transmitter side for encoding the input message and the corresponding H matrix which is derived from the G matrix is used at the receiver side for decoding purposes, but in ARA, G matrix or H matrix is not used for encoding purposes. This is how the encoder complexity is reduced in the encoder. Since H matrix is essential for decoding, we have derived the H matrix which is corresponding to the encoder block diagram for rate 1/2 ARA codes. H matrix can be derived only from G matrix. Therefore the steps for derivation of G matrix are given first, followed by the steps to convert G matrix into corresponding H matrix.

4.1 Constructing generator matrix

A novel method for constructing the generator matrix has been proposed. The flow of G matrix construction constitutes accumulation, puncturing two bit pattern, repetition with three bit pattern, followed by the interleaver. At the end of the interleaving step, the modulo-2 addition is performed, representing the parity in terms of 1's in the G matrix. The corresponding G matrix constructed for our set of code is given as.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The corresponding parity check matrix [H-matrix], is

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

4.2 Hard decision decoding

Hard decision algorithm can be used for decoding an ARA LDPC code [10]. It is an iterative and a highly parallelizable algorithm. Here messages are exchanged along the edges between the bit and check nodes in a Tanner graph. The steps of the algorithm are elaborated below

1. In the first step, all v-nodes c_i send a "message" to their c-nodes f_j having the bit they believe to be the correct one for them. At this stage the only information a v-node c_i has, is the i-th bit received from C.
2. Next, every check node f_j calculates a response to every connected bit node. The response message having the bit, that f_j believes to be the correct one for this v-node c_i assuming that the other v-nodes connected to f_j are correct.

The important point to be noted is that, the decoding algorithm may terminate at this point, if all check equations are fulfilled. In other way, the looping can be stopped, by setting a threshold on the number of loops.

3. In the final phase, the v-nodes receive the messages from the check nodes and verify with the H-matrix, to decide upon the decoding output. This can be done easily with a majority voter.

4. Go to step 2.

The above steps are done iteratively till the error in the received codeword is corrected and finally original message is obtained. There are mainly two types of stop rules that can be used, most of the time both are used in the same system. The first rule is allowing only a certain number of iterations in the decoder, the other rule is to determine whether the condition $C.H^T = 0$ is satisfied. This iterative decoding approach used makes it possible to implement parallelizable decoders.

4.3 Tanner graph

Tanner graph is a bipartite graph, in which the nodes of the graph are separated into two distinctive sets and edges connect the nodes of two different types [1]. The two set of nodes in a tanner graph are called bit nodes (f-nodes) and check nodes (c-nodes). Bit nodes correspond to bits of the codeword or equivalently, to columns of the parity check matrix. Check-nodes correspond to parity check equations or equivalently, to rows of the parity check matrix [1].

Tanner graph provides complete graphical representation of the code and also helps in describing the decoding algorithm. Tanner graph shown in figure 2 represents the same code as in the above parity check matrix. The tanner graph is represented for the (8,15) set of nodes.

The formation of such a graph is rather transparent. It consists of 8 check nodes (the number of parity bits) and 8 bit nodes. Check node f_i is connected to bit node c_j if the element h_{ij} of the parity check matrix (H) is 1.

The sample of the check node processing and bit node processing datasets is shown in the table 1 and table 2 respectively. The table 1 shows the computation done at the check node unit with the data received from bit node unit. The table 2 shows how the data from check node unit is manipulated to interpret the decoded value.

Table 1. Sample of Check node processing data

Check node	Received / Sent	C node Messages
f_0	R	$C_6 \rightarrow 1$ $C_8 \rightarrow 1$
	S	$1 \rightarrow C_6$ $1 \rightarrow C_8$
f_1	R	$C_0 \rightarrow 1$ $C_1 \rightarrow 1$ $C_6 \rightarrow 1$ $C_9 \rightarrow 1$
	S	$1 \rightarrow C_0$ $1 \rightarrow C_1$ $1 \rightarrow C_6$ $1 \rightarrow C_9$
f_2	R	$C_2 \rightarrow 0$ $C_3 \rightarrow 1$ $C_6 \rightarrow 1$ $C_{10} \rightarrow 0$

S	$0 \rightarrow C_2$ $1 \rightarrow C_3$ $1 \rightarrow C_6$ $0 \rightarrow C_{10}$
---	--

Table 2. Sample of Bit node processing data

Bit node	C value	Messages from Check nodes	Decoded Value
C_0	1	$f_1 \rightarrow 1$ $f_3 \rightarrow 1$ $f_5 \rightarrow 1$ $f_6 \rightarrow 0$ $f_7 \rightarrow 1$	1
C_1	1	$f_1 \rightarrow 1$ $f_3 \rightarrow 1$	1
C_2	0	$f_2 \rightarrow 0$ $f_4 \rightarrow 0$ $f_5 \rightarrow 0$	0
C_3	1	$f_2 \rightarrow 1$ $f_4 \rightarrow 1$ $f_5 \rightarrow 1$ $f_6 \rightarrow 0$ $f_7 \rightarrow 1$	1

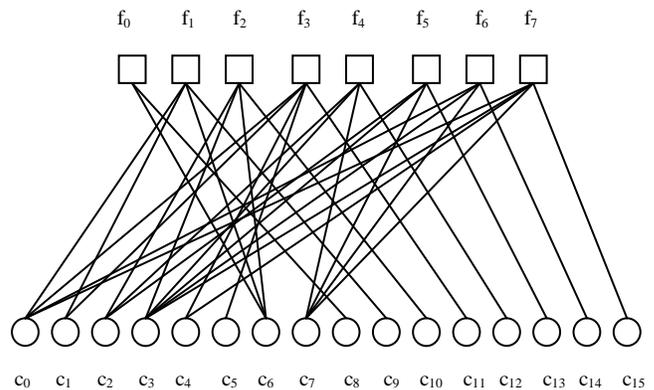


Fig 2 : Tanner graph for (16, 8) ARA-LDPC codes

5. WAVE-PIPELINING

Wave pipelining is a design methodology which can increase the clock frequency of digital systems. It is also known as maximum rate pipelining. Unlike ordinary pipelining, wave pipelining does not require internal clock elements to increase throughput. The synchronization of internal computations is achieved by balancing inherent RC delays of combinational logic elements, thus allowing circuits to be pipelined at a very fine grain level [11].

The speed at which logic can propagate through the circuit depends on the difference between the longest and shortest path delays, not on the longest path delay.

In traditional pipelining, the improvement in throughput can be obtained with the overheads in latency, cycle time, area, and power consumption. The time required for signals to propagate out of the synchronizing elements and for the unintentional clock skew in the arrival of the synchronizer clock signal contributes cycle time overhead. Latency through the traditional pipeline is defined as the total elapsed time from the time of introduction of data at the input to the first stage of the pipeline to the time the results of computations performed

on that data arrive at the output of the final stage of the pipeline [11]. The additional transistors and wires used to implement the synchronizing latches or registers, and from the increased clock buffer area and power needed to drive the clock inputs to the synchronizers, contributes the area and power overhead.

Wave-pipelining depends on the finite propagation delay of signals through a combinational digital circuit to store data. The subsequent data transfer through the combinational network to another synchronizing element is taken care in wave pipelining. Wave pipelined designs guarantees that the subsequent data will not interfere with the current data wave. In this manner, multiple waves of data are simultaneously propagates along distinct regions of the logic network.

Wave-pipelining is a technique to increase the effective number of pipeline stage in digital system without increasing the number of physical registers in the pipeline [12]. There exist many paths having delays that are much smaller than the critical delay path. There is room for improvement in the clock speed where the non-critical paths remain idle. The clock speed can be increased if the idle time of the non-critical paths can be reduced.

6. IMPLEMENTATION

The ARA-LDPC codes are shannon limit approaching, error correcting codes. The ARA-LDPC encoder has been designed and analyzed for various architectures like the sequential, two stage pipelining, three stage pipelining and the Wave pipelining. The most efficient architecture is implemented in FPGA for a block size of 1024 bits and a word length of 8 bits. The corresponding ARA-LDPC decoder has also been designed and implemented in FPGA for a block size of 1024 bits and a word length of 8 bits. The above specifications correspond to deep space applications.

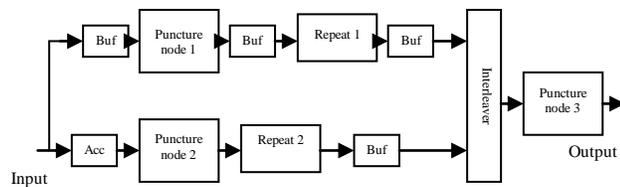


Fig 3: Block diagram of the Wave-pipelined ARA encoder

The Wave-pipelined ARA encoder is designed with the concept of delay equalization. With the analysis of delay among various paths, insertion of buffers is done at the different stages. This form of delay equalization, reduces the critical path delay, thereby optimizes the design. The block diagram of the Wave-pipelined ARA encoder is shown in figure 3.

The ARA codes are systematic codes. Hence in the ARA encoder, the message bits are transmitted along with the parity bits. For every N bits of input, we get 2N bits as output. The ARA-LDPC Encoder has been simulated in Modelsim and synthesized using XILINX-ISE 9.1i for various architectures. The most efficient Wave Pipelined ARA-LDPC encoder has been

implemented in SPARTAN-3E (XC3S-250E) FPGA Kit for a block size of 1024 bits and a word length of 8 bits.

Table 3. Synthesis report of the different architectures of ARA-LDPC

	Sequential	Two stage	Three stage	Wave pipelined
Gate count	182	459	971	115
IOB FF	16	44	40	8
4 input LUTs	8	12	20	8
Bonded IOB	24	49	73	25
Number of slices	4	6	10	5

The ARA-LDPC encoder has been synthesized using Xilinx-ISE 9.1i and Synopsys, for different architectures like the sequential, two stage pipelined, three stage pipelined and the wave pipelined architectures. The synthesis results are shown in Table 3. It is inferred that the gate count increases with the increase in the number of pipelining stages, because of the insertion of the additional registers and flipflops needed for the temporary storage of data between clock cycles.

It is also inferred that the wave pipelined ARA-LDPC Encoder has got the minimum total equivalent gate count value, when compared to the other architectures. Hence it is found that the wave pipelined ARA-LDPC encoder is **Area Efficient** when compared to the other architectures. The timing analysis of the ARA-LDPC encoder, synthesized using different architectures has been performed. Delay values for the various paths between input and output were taken from the Static Timing Report and the critical path delay was calculated. The results obtained are shown in Table 4.

Table 4. Timing analysis of ARA encoder

	Sequential	Two stage pipelined	Three stage pipelined	Wave-pipelined
Critical path delay(ns)	61.05	22.139	27.525	17.855

It is inferred that the wave pipelined ARA-LDPC Encoder has got the **minimum critical path delay** (latency) involved. The area analysis of the ARA-LDPC encoder, synthesized using different architectures, has been performed and the following results were obtained

Table 5. Area analysis of ARA encoder

	Seq- uential	Two stage pipe- lined	Three stage pipe- lined	Wave- pipe- lined
Area (μm^2)	12013.55	15818.68	23479.86	8322.94

It is inferred that the area occupied by the ARA-LDPC Encoder increases with the increase in the number of pipelining stages, because of the insertion of additional registers and flipflops needed for the temporary storage of data, between clock cycles. It is also inferred that the area occupied by the wave pipelined ARA-LDPC encoder is less when compared to the other architectures. Hence it is found that the wave pipelined ARA-LDPC encoder is **Area Efficient**.

The power analysis for the ARA-LDPC encoder, implemented in different architectures, has been performed. It is inferred that the wave pipelined ARA-LDPC encoder utilizes less power when compared to the other architectures employed. Hence it is found to be **Power efficient**.

Table 6. Power analysis of ARA encoder

	Seq- uential	Two stage pipe- lined	Three stage pipe- lined	Wave- pipe- lined
Power (μW)	277.36	343.56	386.01	202.87

7. CONCLUSION

This paper is aimed at designing and analyzing the different architectures for the ARA-LDPC encoder and to implement the most efficient architecture in FPGA. As a result, it is found that wave pipelining is the most efficient architecture, in terms of reduced area, power and latency. Hence the Wave Pipelined ARA - LDPC encoder has been implemented in Spartan 3E FPGA for a block size of 1024 bits and a word length of 8 bits.

A corresponding ARA-LDPC decoder has also been designed and implemented in FPGA for a block size of 1024 bits and a word length of 8 bits. The Soft Decision decoding algorithms like the Min Sum algorithm or the Sum Product algorithm can be used for the decoding of the ARA-LDPC codes. The protograph or the projected graph can also be used for the high speed iterative decoding of these codes. The ARA encoder can be implemented using other wave pipelining techniques such as node collapsing and logic restructuring instead of delay buffer insertion.

8. ACKNOWLEDGMENTS

Thanks to our team members, Ms. Saranya, Ms.Sowmya and Ms.Vaidhehi, for their contribution towards the development of this work.

9. REFERENCES

- [1] Robert G. Gallager, "Low Density Parity Check Codes", Cambridge, IRE Transactions on Information Theory, MIT Press, 1962.
- [2] Aliazam, Abbasafar, Dariush Divsalar, and Kung Yao,"A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions", Journal of Systems and Software, 2007.
- [3] Yifei Zhang and William E. Ryan, "Structured IRA Codes: Performance Analysis and Construction", IEEE transactions on communications, vol. 55, No. 5, 2007.
- [4] Marjan Karkooti, Joseph R. Cavallaro and Predag Radosavljevic, "Configurable LDPC Decoder Architecture for Regular and Irregular Codes", IEEE transactions on communications, vol.53, pp. 73 – 88, 2008.
- [5] Aliazam Abbasfar, D.Divsalar and Kung Yao, "Maximum Likelihood Decoding Analysis of ARA Codes", IEEE Communications Society, Globecom, pp. 514-519, 2004.
- [6] K.Andrews, S.Dolinar, D.Divsalar and J.Thorpe, "Design of Low-Density Parity-Check (LDPC) codes for Deep-Space Application", IPN Progress Report, pp.42-159, 2004.
- [7] Dariush Divsalar, Samuel Dolinar, Christopher Jones, Jeremy Thorpe, and Kenneth Andrews, "Constructing LDPC Codes From Loop-Free Encoding Modules", NPO-42042, NASA Tech Briefs, pp. 31, 2009
- [8] S. Dolinar, D. Divsalar, and F. Pollara , "Code Performance as a Function of Block Size", TMO Progress Report, pp.42-133, 1998.
- [9] Henry D. Pfister and Igal Sason, "Accumulate–Repeat–Accumulate Codes: Capacity-Achieving Ensembles of Systematic Codes for the Erasure Channel with Bounded Complexity", IEEE transactions on information theory, vol. 53, no. 6, 2007.
- [10] S.Dolinar and K.Andrews, "Performance and Decoder Complexity Estimates for Families of Low-Density Parity-Check Codes", IPN Progress Report , pp. 42-168, 2007.
- [11] Wayne P Burleson and Wentai Liu, "Wave Pipelining: A Tutorial and Research Survey", IEEE transactions on VLSI Systems, Vol. 6, No. 3, 1998.
- [12] Hirak Kumar Maity, Mitra Barun Sarkar and A.Chakrobarty , "Wave Pipelining: An Analysis for High Performance Digital Circuits", International Journal of Electronic Engineering Research , ISSN 0975 – 6450, Volume 1, Number 3, pp. 269–278, 2009.