

DECODING CONVOLUTIONAL CODES IN VLSI

P.G. Gulak, V.P. Roychowdhury and T. Kailath
Information Systems Laboratory
Stanford University

Abstract

Decoders for Convolutional Codes can be realized by resorting to a well-known interconnection scheme in parallel processing referred to as the shuffle-exchange network. The result is significant in that layout techniques that achieve a proven lower bound on implementation area in a VLSI medium have recently been established for the shuffle-exchange graph. The strategy involves constructing a fine-grain parallel processing network consisting of many simple processors that reside on a single die, connected to perform the Viterbi Algorithm in a highly parallel way. The concept can be shown to apply universally to all rate- $1/n$ convolutional codes, including those that are generated by shift register sequences containing simple forms of feedback. The state transition diagram of rate- k/n convolutional codes can be mapped to the Cartesian product of shuffle-exchange graphs. The Kronecker product of the corresponding adjacency matrix allows for an equivalent algebraic description of this concept.

1. Introduction

In this paper, we present mathematical insights that will allow us to apply the technological capabilities of VLSI to a specific problem that is of considerable interest in high-speed reliable digital communications. Specifically, we pursue the development of high performance VLSI circuits for implementing rate- k/n convolutional Viterbi decoders by performing certain graph operations on the state transition diagram. Advantageous graph theoretic properties emerge when the structure of the decoder takes the form of a shuffle-exchange graph.

Shuffle-Exchange (SE) graphs were originally proposed in 1971 as an efficient interconnection methodology for parallel computation [1]. With the advent of VLSI, the question of how to best layout the SE graph on a grid using as little area as possible has been extensively investigated. Using these concepts, convolutional decoding can be implemented with a network of simple processors that are interconnected using a shuffle-exchange scheme and that are programmed to implement the Viterbi Algorithm (VA). The result is significant in that layout techniques that achieve a proven lower bound on implementation area in a VLSI medium have recently been established for the shuffle-exchange graph. Hence, we have proven lower bounds on the implementation area of these types of decoders.

The attractiveness of this approach is that it produces layouts that are technology independent, regular, easily partitionable into identical building blocks and extendable to various constraint lengths. In addition, state information is processed simultaneously for all states exploiting the parallelism of the algorithm, therefore making it an excellent candidate for high-speed data communication applications. This approach is not only applicable to the decoding of convolutional codes, but also to the maximum likelihood sequence estimation on intersymbol interference channels and to the coded modulation receiver concepts of Ungerboeck. The results of this paper are developed first for rate- $1/n$ codes and then extended to rate- k/n convolutional codes.

2. Rate $1/n$ Codes

The basic theory behind convolutional codes (CC) and the Viterbi decoding algorithm is widely available in the literature; a good introduction is given by Forney [2]. The state transition diagram of the

encoder ultimately defines the complexity of the corresponding Viterbi decoder. The traditional metrics of complexity such as equivalent gate count loses its former relevance as the technology of VLSI is now capable of realizing chips with hundreds of thousands of transistors [3]. What is not especially clear or obvious, until this time, is the type of topologies that are appropriate for Viterbi decoders, and how "expensive" (with respect to an area criterion) is the wiring, for embedding these transistors. This section attempts to provide mathematical insights into this problem.

A binary FIR CC with memory v , can be implemented with a v stage binary shift register having only simple feedforward interconnections between adjacent memory elements. The number of memory elements is defined to be the constraint length of the code. The state diagram for such a shift register circuit has $N = 2^v$ possible states and is known as a de Bruijn graph. The de Bruijn diagram can be described by an adjacency matrix B that has nonzero entries at b_{ij} for all $b_{ij} = b_{(x_{v-1}, \dots, x_0)(x_{v-2}, \dots, x_0 x_{v-1})}$ and $b_{ij} = b_{(x_{v-1}, \dots, x_0)(x_{v-2}, \dots, x_0 \bar{x}_{v-1})}$, where each state or node, labelled from 0 to $N-1$, is associated with an equivalent binary v -tuple (x_{v-1}, \dots, x_0) and \bar{x}_m is the logical complement of x_m .

The adjacency matrix for the shuffle-exchange graph can be derived by modifying B according to the following transformation T : *Delete element $b_{ij} = b_{(x_{v-1}, \dots, x_0)(x_{v-2}, \dots, x_0 \bar{x}_{v-1})}$ and reinstall at location $b_{kj} = b_{(x_{v-2}, \dots, x_0 x_{v-1})(x_{v-2}, \dots, x_0 \bar{x}_{v-1})}$.*

Hence, this observation allows us to prove a theorem that demonstrates that there exists an equivalence transformation T where

$$B_{deBruijn} \xrightarrow{T} B_{SE}$$

can be interpreted as being functionally equivalent in the context of the VA. The motivation for exploiting such a mapping is provided by the relatively superior graph-theoretic properties of the shuffle-exchange graph which produce tangible economies when considering a VLSI implementation. For instance, the minimum bisection width of the SE graph is smaller than that of the equivalent de Bruijn graph guaranteeing that the SE layout area is always smaller than the equivalent de Bruijn graph when embedded within a VLSI grid model. In addition, a constructive layout methodology has been recently developed for the SE graph [4] that achieves the proven lower bound on layout area of $\Omega(N^2/\log^2 N)$. However, the diameter of the SE graph is increased relative to that of a de Bruijn graph which influences to some extent the maximum edge length of the graph embedding.

To diagrammatically illustrate the consequences of such a theorem, consider the shuffle-exchange VLSI grid model layout in Figure 1. This diagram corresponds to a fine-grain parallel processor that implements a Viterbi decoder for binary FIR CC with memory five ($v = 5$). Each node is an Add, Compare, Select (ACS) processor and a survivor sequence register with a small amount of control logic while edges define processor interconnections. Solid edges are the exchange connections while dashed edges are the shuffle edges.

Convolutional encoders containing feedback connections have a state diagram which can be mapped to a *subgraph* of a shuffle-exchange graph. Feedback elements constrain state trajectories to a subgraph of the shuffle-exchange graph since they can force one to take only a subset of the q possible state transitions out of a particular state in $GF(q)$. Feedforward shift-register encoders can be accommodated in the above

results by resorting to the equivalent systematic feedback encoder.

The consequence of feedback/feedforward elements implies that a convolutional code that punctures state transitions in the trellis diagram always has a Viterbi decoder layout area of the type considered in this paper that is less than or equal to that of the corresponding shuffle-exchange graph. In addition, this leads to a theorem that assures us that there exists a feedback circuit that can delete any set of edges in this type of Viterbi decoder realization. This may be advantageous from a performance point of view if the longest edges or wires are deleted (long wires have more capacitance requiring more drive capability hence limiting ultimate clock speed and data rate). When it becomes desirable that a prespecified set of edges should be deleted to allow better performance, a more compact layout or better partitioning across several chips how do we find the convolutional codes with the best code properties? This open research question illustrates how implementation issues may constrain the definition of what is meant by a "good code", in addition to the usual metrics (such as d_{\min}).

3. Rate k/n Codes

For rate- k/n convolutional codes, the state diagram is much more complex than rate- $1/n$ codes. Rate- k/n convolutional decoders can be realized by resorting to a parallel processing network whose interprocessor connections are defined by the Cartesian product of shuffle-exchange graphs.

In the most general case, a rate- k/n convolutional encoder can be modelled as containing k shift registers of V_i symbols each stored in the $v = v_1 + v_2 + \dots + v_i + \dots + v_k$ memory elements. The state machine description by itself may be fairly complex to describe succinctly (especially in the presence of feedback elements) which has delirious effects on developing a good VLSI layout strategy for Viterbi based convolutional decoders. The approach proposed here is to decompose the state transition diagram into simple factors.

It is rather convenient to be able to express the structure of a given graph in terms of smaller and simpler graphs. Consider two graphs G_1 and G_2 that have disjoint point sets V_1 and V_2 and line sets X_1 and X_2 . Of interest in this paper is an operation on G_1 and G_2 which results in a graph G whose set of points is the Cartesian product $V_1 \times V_2$. To define the Cartesian product $G_1 \times G_2$, consider any two points $u = (u_1, u_2)$ and $v = (v_1, v_2)$ in $V = V_1 \times V_2$. Then u and v are adjacent (adj) in $G_1 \times G_2$ whenever $[u_1 = v_1 \text{ and } u_2 \text{ adj } v_2]$ or $[u_2 = v_2 \text{ and } u_1 \text{ adj } v_1]$. The overall state diagram can then be identified as being the Cartesian product of the individual shuffle-exchange equivalent state graphs associated with each of the k shift registers in the encoder. Feedback connections, again, only serve to constrain operation of the encoder to a subgraph of that given by the Cartesian product formulation.

A graph is completely determined by its adjacencies which can be conveniently stated in matrix form. The structure of the adjacency matrix that describes the state diagram of rate- k/n convolutional codes can be exploited to advantage when interpreted as a Kronecker product of matrices. This gives us an algebraic alternative to the Cartesian product interpretation. The Kronecker product $P \otimes Q$ of the two matrices P and Q is defined as

$$P \otimes Q = \begin{bmatrix} aA & aB & bA & bB \\ aC & aD & bC & bD \\ cA & cB & dA & dB \\ cC & cD & dC & dD \end{bmatrix} \quad \text{where} \quad P = \begin{bmatrix} a & b \\ c & d \end{bmatrix}; \quad Q = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

When the adjacencies of G_1 and G_2 are given by P and Q , respectively, then the relationship between the Cartesian product of graphs and the Kronecker product of matrices is given by the following

$$(1_P \otimes Q) + (P \otimes 1_Q)$$

where 1_P is a unit matrix of dimension P and $+$ is the logical *or* operation. This forms the basis of a convenient algebraic description of the decoder required.

In this section, we have indicated that the overall state diagram of certain shift register processes can be equivalently described by a graph which is a Cartesian product of subgraphs of this same shift register process. As a consequence, the concept of a quotient network, derived by analogy with the notion of quotient groups in abstract algebra, allows us to define how an arbitrarily large decoder can be virtualized on a fixed (perhaps scarce) amount of physical hardware. If one of the factors of the state diagram is to be the physical realization of this network the other factors define the virtual data transfers required by the Viterbi Algorithm.

4. Discussion

In conclusion, by understanding the underlying algorithmic structure of Viterbi decoders for convolutional codes we have unveiled a new approach for implementing such decoders in VLSI and have provided a justification as to why such a strategy should be used. In a broader context, this work complements the persistent efforts in the literature directed at understanding the underlying structure of the of other types of dynamic programming algorithm applications such as the recognition of context free languages and the dynamic time warping of speech signals.

5. Acknowledgements

This work was supported in part by the U. S. Army Research Office, under contract DAAL03-86-K-0045 by the National Science Foundation Grant DCI-84-21315-A1 and by the Rockwell International Contract INT 6G3052.

6. References

- [1] Stone H.S., Parallel processing with the Perfect Shuffle, IEEE Trans. on Computers, Vol. C-20, No. 2, Feb. 1971, pp. 153-161.
- [2] Forney, G.D., The Viterbi Algorithm, Proc. IEEE, Vol. 61, March 1973, pp. 268-279.
- [3] Thompson C.D., A Complexity Theory for VLSI, Ph.D. Dissertation, Dept. of Computer Science, Carnegie Mellon University, 1980.
- [4] Leighton F.T., Complexity Issues in VLSI, MIT Press, 1983.

7. Figures

Figure 1: Grid model layout of a SE based Viterbi Decoder for a constraint length five convolutional code.

