GOOD TRELLISES FOR IC IMPLEMENTATION OF VITERBI DECODERS FOR LINEAR BLOCK CODES

Technical Report

to

NASA

Electrical Engineering Division
Goddard Space Flight Center
Greenbelt, Maryland   20771

Grant Number NAG 5-2938
Report No. 96-002

Principal Investigator: Shu Lin

Department of Electrical Engineering
University of Hawaii at Manoa
Honolulu, Hawaii   96822

May 24, 1996

GOOD TRELLISES FOR IC IMPLEMENTATION OF VITERBI DECODERS FOR LINEAR BLOCK CODES

Hari T. Moorthy, Shu Lin and Gregory T. Uehara

Technical Report Number 96-002

May 24, 1996

Good Trellises for IC Implementation of Viterbi Decoders for Linear Block Codes[1]

Hari T. Moorthy, Shu Lin and Gregory T. Uehara
Department of Electrical Engineering,
University of Hawaii,
Honolulu, Hawaii 96822

Abstract

This paper investigates trellis structures of linear block codes for the IC (integrated circuit) implementation of Viterbi decoders capable of achieving high decoding speed while satisfying a constraint on the structural complexity of the trellis in terms of the maximum number of states at any particular depth. Only uniform sectionalizations of the code trellis diagram are considered. An upper bound on the number of **parallel and structurally identical (or isomorphic)** subtrellises in a proper trellis for a code without exceeding the maximum state complexity of the minimal trellis of the code is first derived. Parallel structures of trellises with various section lengths for binary BCH and Reed-Muller (RM) codes of lengths 32 and 64 are analyzed. Next, the complexity of IC implementation of a Viterbi decoder based on an L-section trellis diagram for a code is investigated. A structural property of a Viterbi decoder called **ACS-connectivity** which is related to state connectivity is introduced. This parameter affects the complexity of wire-routing (interconnections within the IC). The effect of five parameters namely: (1) effective computational complexity; (2) complexity of the ACS-circuit; (3) traceback complexity; (4) ACS-connectivity; and (5) branch complexity of a trellis diagram on the VLSI complexity of a Viterbi decoder is investigated. It is shown that an IC implementation of a Viterbi decoder based on a non-minimal trellis requires less area and is capable of operation at higher speed than one based on the minimal trellis when the commonly used ACS-array architecture is considered.

1. Introduction

Any linear block code can **theoretically** be decoded by applying the Viterbi algorithm to a trellis for the code. Trellises for block codes were first described in [1]-[3]. After Forney's refinement of the structure of these trellises [4], their potential in the **practical** decoding of block codes has been realized by many others who have published extensively on various aspects of the trellis structure of block codes [5]-[25]. In some of the above papers, one goal was to minimize the maximum number of states in the trellis at any depth by considering all possible permutations of the code [6]. For some codes such as Reed-Muller codes, this optimum permutation is known [7]. For most others only bounds are known.

Even when the optimum order of bits is known or a good permutation is known (if the optimum order is unknown), previous work has focussed on minimization of the number of computations required for decoding [12, 22, 24]. If the actual decoding is intended to be performed using a stored program approach that executes the operations needed to decode a received vector sequentially, then this approach will lead to the fastest decoding speed. However, if an IC implementation is intended, then an alternative approach is more suitable. Given a constraint on the amount of hardware (determined by the number of states and the complexity of branches) in the decoder, **decoding must be done as fast as possible; not necessarily with as few computations as possible.** To achieve this end, we propose the use of non-minimal trellises with parallel structure in which the maximum state space dimension is **not greater than** the maximum state space dimension of the minimal trellis of a code. In this paper, certain properties concerning the state connectivity and branch complexity [9] of this non-minimal trellis are derived which demonstrate that the non-minimal trellis implementation would require **less area in an IC implementation** than the corresponding minimal trellis when the ubiquitous ACS array architecture [26]-[28] is used for implementation. We caution that if a different architecture as proposed in [27] or [24] is chosen for implementation, then the trellis structure that is best suited will in general be different from the proposed trellis.

The number of decoding operations required by the standard trellis-based Viterbi decoding algorithm depends on the sectionalization of the trellis used for decoding. Most of the previous works focussed on uniform sectionalization of a trellis, each section consists of the same number of code symbols. However, Lafourcade and Vardy [22] recently showed that non-uniform sectionalization of a trellis often results in less number of decoding operations than uniform sectionalization. They have devised an efficient algorithm for finding optimal sectionalization of a trellis for minimizing the total number of decoding operations required for maximum-likelihood trellis decoding. Optimal sectionalization of a trellis to minimize computational complexity is also investigated in [24]. In this paper, we only investigate

2

good trellises with uniform sectionalization for IC implementation of Viterbi decoders. Particularly, we are concerned with those structures, such as parallel structure, regularity and state-connectivity that: (1) affects the complexity of wire-routing (interconnections) within the IC and chip-size; and (2) facilitate parallel and pipeline decoding process to achieve high decoding speed. Since non-uniform sectionalization of a trellis requires less decoding operations, this advantage over uniform sectionalization and other properties definitely should be investigated for IC implementation of Viterbi decoders to achieve high decoding speed. This investigation is beyond the scope of this paper.

Trellises for block codes are often loosely connected. A properly constructed trellis may consist of many parallel and structurally identical (isomorphic) subtrellises of smaller state space dimension without cross-connections between them. Consequently, identical Viterbi decoders of much smaller complexity can be devised to process the subtrellises independently in parallel without internal communication between them. This not only simplifies the IC implementation but also speeds up the decoding process. For example, the (32,16,8) Reed-Muller (RM) code, also an extended BCH code, has a 4-section, 64-state minimal trellis diagram, which consists of eight parallel and structurally identical 8-state subtrellises without cross-connections among them. As a result, we can devise eight identical 8-state Viterbi decoders to process the 8 subtrellises in parallel without communication between them. At the end, there are 8 survivors (one from each subtrellis) and the best one will be chosen as the decoded codeword. This reduces the implementation of a 64-state Viterbi decoder to the implementation of an 8-state decoder and using 8 copies of it. This parallel structure reduces the wire-routing and internal communications within IC which reduces chip size and improves decoding speed. If the state and branch complexities of each subtrellis is small and the total number of subtrellises is small, all the subtrellis decoders can be put on a single chip, such as for the (32,16,8) RM code [29]. However, if the state and branch complexities are big, then each subtrellis decoder (or several of them) can be implemented on a single chip. This provides flexibility in chip plan and decoder architecture.

The two fundamental bottlenecks to Viterbi decoding (decoding speed) are the internal communications between ACS (add-compare-select) units and comparisons of incoming branches (radix-profile) at each state [28, 30]. Properly designed parallel structure in a trellis would overcome these obstacles without exceeding the maximum state space dimension of the minimal trellis. For example, a (64,40,8) RM subcode which is being considered by NASA for high-speed satellite communications has an 8-section 2048 state trellis. This trellis consists of 32 parallel and structurally identical 64-state subtrellises. The last 4 sections of each subtrellis are a mirror image of the first 4 sections as shown in Figure 3. As a result, a bidirectional decoding can be performed. Furthermore, the maximum component of the

3

radix profile for each half subtrellis is only 8. A 64-state subtrellis decoder can be implemented on a single chip in 0.5 micron CMOS technology which can operate at a decoding speed of 600 Mbps [31]. Other structural properties of the subtrellises for this (64,40,8) RM subcode which simplifies the IC implementation will be discussed later. Parallel structure therefore, offers simplification, flexibility and higher decoding speed for IC implementation. We must note that the parallel structure does not reduce the total number of single-state processors, i.e., number of ACSs.

In this paper, we investigate trellis structures, particularly the parallel structure, of linear block codes for implementation of Viterbi decoders capable of achieving high decoding speed while satisfying a constraint on the structural complexity of the trellis in terms of the maximum number of states at any depth. Only uniform sectionalizations of the code trellis are considered. The organization of the paper is as follows.

In Section 2, using the theory of L-section minimal trellis diagrams, an upper bound on the number of parallel isomorphic subtrellises in a proper trellis for a code **without** exceeding the maximum state space dimension of the minimal trellis of the code is derived. In Section 3, we analyze the trellises for all extended BCH and RM codes of lengths 32 and 64. In Section 4, we define parameters related to the complexity of a Viterbi decoder IC using the ACS-array architecture for linear block codes. Section 5 treats examples and in Section 6 we use the results of this paper to design a trellis for a $(64, 40)$ RM subcode.

2. Trellises with Parallel Structure for Linear Block Codes with Constraint on Maximum State Space Dimension

The objective of this section is to show that we can build a trellis for a linear block code C which is a disjoint union of a certain desired number of parallel isomorphic subtrellises. *Although this trellis is not minimal, its state space dimension at every depth is less than or equal to the maximum state space dimension of the minimal trellis.* The conditions under which such a trellis construction is possible and an upper bound on the number of such parallel subtrellises are derived. In some cases, the minimal trellis itself possesses a parallel structure. The number of such parallel subtrellises (if any) in the minimal trellis is derived.

2.1. Preliminaries

We consider only binary (N, K, d_{min}) linear block codes. Let L, M be positive integers such that $LM = N$. The minimal (up to graph isomorphism) L-section trellis, is a well

4

understood graphical representation of the code [5, 9]. Let the sets of states at the end of each section be denoted $\{S_0, S_M, S_{2M}, \ldots, S_{(L-1)M}, S_{LM}\}$. We define a sequence $\{s_0, s_M, \ldots, s_{LM}\}$ called the **state complexity profile (SCP)** of the trellis and given by $s_{iM} = \log_2(|S_{iM}|)$ for $0 \leq i \leq L$. The minimal L-section trellis of a code C has the property that every component of its SCP is less than or equal to the corresponding component in the SCP of any other proper L-section trellis for C. The maximum among the $N+1$ components in the SCP of the minimal N-section trellis ($L = N, M = 1$) for C is denoted $s_{\max}(C)$ and we will denote the maximum of the components in the SCP of the minimal L-section trellis for C as $s_{\max,L}(C)$. For a binary N-tuple $\mathbf{v} = (v_1, \ldots, v_N)$, let $p_{h,h'}[\mathbf{v}]$ denote the $(h' - h)$-tuple $(v_{h+1}, \ldots, v_{h'})$ and let $p_{h,h'}[C] = \{p_{h,h'}[\mathbf{c}] : \mathbf{c} \in C\}$. Let $C_{h,h'}$ be the linear subcode of C consisting of all codewords whose components are all zero except for the $(h' - h)$ components from the $(h + 1)$-th bit position to the h'-th bit position.

In an L-section minimal trellis for a block code, there may be a set of parallel branches between two adjacent states. In such a case, we call the entire set of parallel branches a **composite branch**. Each composite branch in the i-th section $1 \leq i \leq L$, is made up of 2^{P_i} parallel branches where P_i is the dimension of the subcode denoted $C_{(i-1)M,iM}$ [9]. In the i-th section of an L-section trellis for a linear block code $1 \leq i \leq L$, the number of distinct branch metrics that have to be computed is 2^{D_i} where D_i is the dimension of the subcode $p_{(i-1)M,iM}(C)$ and this number is much less than the total number of branches. D_i is the rank of the submatrix formed by M columns from the $((i - 1)M + 1)$-th to the (iM)-th column of the generator matrix of the code and is upper bounded by M. For $1 \leq i \leq L$, let the number of composite branches merging into any state $s \in S_{iM}$ be $2^{\delta_{iM}}$ (it is the same for any state in S_{iM}). For an L-section trellis for C, we define the **converging branch profile (CBP)** as the ordered sequence $\{\delta_M, \delta_{2M}, \ldots, \delta_{LM}\}$. For $0 \leq i < L$, let the number of composite branches emanating from any state $s \in S_{iM}$ be $2^{\lambda_{iM}}$, (it is the same for any state in S_{iM}). The ordered sequence $\{\lambda_0, \lambda_1, \ldots, \lambda_{(L-1)M}\}$ is called **diverging branch profile (DBP)**. Then δ and λ are related as follows:

$$\delta_{iM} = s_{(i-1)M} + \lambda_{(i-1)M} - s_{iM}. \tag{2.1}$$

Based on the theory of L-section trellises [9], it can be shown that

$$\lambda_{(i-1)M} = \dim(C_{(i-1)M,N}) - \dim(C_{iM,N}) - P_i \tag{2.2}$$

which implies that $\lambda_{(i-1)M}$ equals the number of rows of a trellis oriented generator matrix of C whose leading 1 occurs among the positions $\{(i - 1)M, (i - 1)M + 1, \ldots, iM - 1\}$ and whose span is not contained in the i-th section. These dimensions can be easily determined from the trellis oriented generator matrix of the code [9, 16, 20]. The two sequences,

$\{\delta_M, \delta_{2M}, \ldots, \delta_{LM}\}$ and $\{\lambda_0, \lambda_1, \ldots, \lambda_{(L-1)M}\}$ provides a measure of the state connectivity of an L-section minimal trellis. In IC implementation of a Viterbi decoder, δ_{iM} is called a radix number.

2.2. Parallel Trellises

Let **G** be the trellis oriented generator matrix of an (N, K) linear block code C [4]. Let $r = (r_1, r_2, \ldots, r_N)$ be a typical row of **G**. Then, we define the **span** of **r**, denoted $span(\mathbf{r})$, to be the smallest interval $[i, j]$, $1 \leq i \leq j \leq N$ which contains all the non-zero elements of **r**. For a row **r** whose span is $[i, j]$ we also define an **active span** of **r**, denoted $aspan(\mathbf{r})$, as $[i, j-1]$ if $i < j$ and $aspan(\mathbf{r}) = \phi$ if $i = j$. The trellis oriented matrix has the following properties: (1) The leading 1 of every row occurs in an earlier position than the leading 1 of the row below it; (2) The trailing 1 of every row occurs at a different position from the trailing 1 of every other row. Any other trellis oriented matrix for C has the same set of **row spans** although the rows themselves may be different [20]. Let $\mathcal{T}$ be the minimal N-section trellis for C. Given the trellis oriented generator matrix of a code, the state space dimension at any position l is just equal to the number of rows whose active span contain l [20]. For example, consider the following trellis oriented generator matrix:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & r_1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & r_2 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & r_3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & r_4 \end{pmatrix}$$

for which $aspan(\mathbf{r}_1) = [1, 3]$, $aspan(\mathbf{r}_2) = [2, 6]$, $aspan(\mathbf{r}_3) = [3, 5]$ and $aspan(\mathbf{r}_4) = [5, 7]$. For each l, $0 \leq l \leq 8$, counting the number of rows which are active at that l yields the state dimension profile (SCP), $\{0, 1, 2, 3, 2, 3, 2, 1, 0\}$. For $0 \leq l \leq N$, let $s_l(C)$ denote the dimension of the l-th state space of C. Let $s_{\max}(C)$ be the maximum among the state space dimensions. Define the non-empty set,

$$I_{\max}(C) = \{l : s_l(C) = s_{\max}(C)\}. \tag{2.3}$$

Suppose we choose a subcode C' of C such that $\dim(C') = \dim(C) - 1$ and the set of coset representatives $[C/C']$ is generated by the single row $\mathbf{r} \in \mathbf{G}$. From the above statement about $s_l(C)$, it is clear that $s_l(C') = s_l(C) - 1$ for exactly those l where **r** is active. i.e., $l \in aspan(\mathbf{r})$. For other positions $l \notin aspan(\mathbf{r})$ we have $s_l(C') = s_l(C)$. Hence we have the following proposition.

Lemma 1: If there exists a row **r** in the trellis oriented generator matrix **G** for the code C such that $aspan(\mathbf{r}) \supseteq I_{\max}(C)$, then we can form a subcode C' of C generated by $\mathbf{G} - \{\mathbf{r}\}$

6

such that $s_{\max}(C') = s_{\max}(C) - 1$ and $I_{\max}(C') \supseteq I_{\max}(C)$. ∎

In fact $I_{\max}(C') = I_{\max}(C) \cup \{l : s_l(C) = s_{\max}(C) - 1, l \notin aspan(r)\}$. Since **G** is a trellis oriented generator matrix, $\mathbf{G'} = \mathbf{G} - \{\mathbf{r}\}$ is also trellis oriented. We can apply the above proposition again to C' if there exists a row $\mathbf{r'} \in \mathbf{G'}$ with $aspan(r') \supseteq I_{\max}(C')$. This yields a subcode $\tilde{C}$ with dimension smaller by one and $s_{\max}(\tilde{C}) = s_{\max}(C') - 1$. If no such row $\mathbf{r'}$ exists, the proposition cannot be applied and the recursion stops. The above proposition can be generalized.

Let $R(C)$ be the following subset of rows of **G**,

$$R(\mathbf{C}) = \{\mathbf{r} \in \mathbf{G} : aspan(\mathbf{r}) \supseteq I_{\max}(\mathbf{C})\}. \tag{2.4}$$

Let $\rho = |R(\mathbf{C})|$ where $|Q|$ denotes the cardinality of any finite set Q.

Theorem 1: With $R(C)$ defined as above and $\rho = |R(\mathbf{C})|$, let $1 \leq \rho' \leq \rho$. There exists a subcode $\mathbf{C'}$ of $\mathbf{C}$ such that $s_{\max}(\mathbf{C'}) = s_{\max}(\mathbf{C}) - \rho'$ and $\dim(\mathbf{C'}) = \dim(\mathbf{C}) - \rho'$ if and only if there exists a subset $R' \subseteq R(\mathbf{C})$ consisting of ρ' rows of $R(\mathbf{C})$ such that for every l satisfying $s_l(\mathbf{C}) > s_{\max}(\mathbf{C'})$, there exist at least $s_l(\mathbf{C}) - s_{\max}(\mathbf{C'})$ rows in R' whose active spans contain l. The set of coset representatives $[\mathbf{C}/\mathbf{C'}]$ is generated by R'.

Proof: Suppose $R' = \{\mathbf{r'}_1, \ldots, \mathbf{r'}_{\rho'}\}$ satisfies the conditions in the hypothesis. Since $R' \subseteq R(\mathbf{C})$, $I_{\max}(\mathbf{C}) \subseteq aspan(\mathbf{r'}_i)$ for $1 \leq i \leq \rho'$. Consider the subcode generated by $\mathbf{G} - R'$. For those $l \in I_{\max}(\mathbf{C})$, we can determine $s_l(\mathbf{C'})$ by counting the number of rows $\mathbf{r} \in (\mathbf{G} - R')$ that are active at the position l. But this number is exactly less than $s_{\max}(\mathbf{C})$ by ρ'. For $l \notin I_{\max}(\mathbf{C})$ and satisfying $s_l(\mathbf{C}) > s_{\max}(\mathbf{C'})$, we are assured by the hypothesis that $s_l(\mathbf{C})$ will be reduced by at least $s_l(\mathbf{C}) - s_{\max}(\mathbf{C'})$ thus guaranteeing that $s_{\max}(\mathbf{C'}) = s_{\max}(\mathbf{C}) - \rho'$.

To prove the converse, let $\mathbf{C'}$ be a subcode of $\mathbf{C}$ whose dimension is $\dim(\mathbf{C}) - \rho'$ and satisfying $s_{\max}(\mathbf{C'}) = s_{\max}(\mathbf{C}) - \rho'$. Without loss of generality, we may let $\mathbf{C'}$ be generated $\mathbf{G} - R'$ for some subset R' of the trellis oriented generator matrix $\mathbf{G}$ of $\mathbf{C}$ with $|R'| = \rho'$. Let $\tilde{T}$ be the minimal trellis corresponding to $\mathbf{G}$. Let T' be the minimal trellis for $\mathbf{C'}$. Let $N_l(R')$ be the number of rows $\mathbf{r'}$ in R' such that $l \in aspan(r')$. Then, at every position l, $0 \leq l \leq N$, we have

$$s_l(\tilde{T}) = s_l(\mathbf{C'}) + N_l(R') \geq s_l(\mathbf{C}) \tag{2.5}$$

since $s_l(\mathbf{C})$ is the smallest possible state space dimension. Therefore

$$
\begin{aligned}
N_l(R') &\geq s_l(\mathbf{C}) - s_l(\mathbf{C'}), \\
N_l(R') &\geq s_l(\mathbf{C}) - s_{\max}(\mathbf{C'}).
\end{aligned}
\tag{2.6}
$$

7

For every l, at least $s_l(\mathbf{C}) - s_{\max}(\mathbf{C}')$ rows of R' are active. Also, for every $l \in I_{\max}(\mathbf{C})$, we have $N_l(R') \geq s_{\max}(\mathbf{C}) - s_{\max}(\mathbf{C}') = r'$. So all the rows $\mathbf{r}' \in R'$ satisfy $aspan(\mathbf{r}') \supseteq I_{\max}(\mathbf{C})$. Thus $R' \subseteq R(\mathbf{C})$. ∎

The **utility of the above theorem** is that it shows how to choose a subcode $\mathbf{C}'$ of $\mathbf{C}$ with $s_{\max}(\mathbf{C}') = s_{\max}(\mathbf{C}) - \dim([\mathbf{C}/\mathbf{C}'])$, such that one can build a non-minimal trellis $\mathcal{T}$ for $\mathbf{C}$ with the following properties:

1. The maximum state space dimension of $\mathcal{T}$ is $s_{\max}(\mathbf{C})$.

2. $\mathcal{T}$ is the union of $2^{\dim[\mathbf{C}/\mathbf{C}']}$ parallel isomorphic subtrellises $\mathcal{T}_i$ with each $\mathcal{T}_i$ being isomorphic to the minimal trellis for $\mathbf{C}'$.

3. **Upper Bound on Parallelism:** The smallest such subcode has dimension lower bounded by $\dim(\mathbf{C}) - |R(\mathbf{C})|$. i.e., the maximum number of parallel subtrellises one can obtain with the constraint that the total state space dimension never exceeds $s_{\max}(\mathbf{C})$ is upper bounded by $2^{|R(\mathbf{C})|}$ with $R(\mathbf{C})$ as defined above.

4. **Parallelism of the Minimal Trellis:** The logarithm to the base 2 of the number of parallel isomorphic subtrellises in a minimal L-section trellis for a binary (N, K) linear block code is given by the number of rows in its trellis oriented generator matrix whose active span contains the integers $\{M, 2M, \ldots, (L-1)M\}$ where $N = LM$.

As an example, consider the extended and permuted $(32, 21, 6)$ BCH code. A parity check matrix for this code with an optimum order of bits with respect to trellis state complexity is shown in Figure 1. The set of spans of any trellis oriented generator matrix for this code is given in Table 1. The 4-section minimal trellis has the SCP $\{0, 7, 9, 7, 0\}$ giving $s_{\max,4}(\mathbf{C}) = 9$. This trellis has 2 parallel isomorphic subtrellises. $I_{\max}(\mathbf{C}) = \{16\}$ and it can be verified that $|R(\mathbf{C})| = 9$. In an attempt to build a trellis consisting of 64 parallel subtrellises while satisfying the upper bound of 9 on the maximum state space complexity, we let $\rho' = 6$. So $s_{\max}(\mathbf{C}) - \rho' = s_{\max}(\mathbf{C}') = 3$. The set $\{l : s_l(\mathbf{C}) > s_{\max}(\mathbf{C}')\} = \{8, 16, 24\}$. However, we find that no subset R' of $R(\mathbf{C})$ exists satisfying the conditions in Theorem 1. Hence we **cannot** build a trellis consisting of 64 parallel subtrellises for this code without violating the constraint on the maximum state space dimension. If we choose $\rho' = 5$, then we can find a subset $R' = \{\mathbf{r}_6, \mathbf{r}_7, \mathbf{r}_8, \mathbf{r}_{12}, \mathbf{r}_{15}\} \subseteq R(\mathbf{C})$ that satisfies all the conditions in Theorem 1. Hence choosing the subcode $\mathbf{C}'$ generated by $G - R'$ we obtain a trellis $\mathcal{T}$ for $\mathbf{C}$ consisting of 32 parallel isomorphic subtrellises. Each subtrellis is isomorphic to the minimal trellis for $\mathbf{C}'$ which has $s_{\max}(\mathbf{C}') = 4$.

For the same code, the 32-section minimal trellis has the SCP that gives $s_{\max,32}(\mathbf{C}) = 10$ and $I_{\max}(\mathbf{C}) = \{12, 14, 18, 20\}$. Using Table 1, we find that $|R(\mathbf{C})| = 2$. In an attempt to

build a trellis consisting of 4 parallel subtrellises while satisfying the upper bound of 10 on maximum state space dimension, we let $\rho' = 2$. So $s_{\max}(C') = 8$. The set $\{l : s_l(C) > s_{\max}(C')\} = \{10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22\}$. We find that the subset of two rows having span $[8, 25]$ and $[10, 23]$ satisfy the conditions in Theorem 1.

This decomposition of a trellis into parallel and structurally identical subtrellises of smaller state complexity without cross-connections between them has significant advantages for IC implementation of Viterbi decoding. Identical Viterbi decoders of much simpler complexity can be devised to process the subtrellises independently in parallel without internal communications (or information transfer) between them. Internal information transfer limits the decoding speed [28, 30]. Furthermore, the number of computations to be carried out per subtrellis is much smaller than that of a fully connected trellis. As a result, the parallel structure not only simplifies the decoding complexity but also speeds up the decoding process. For example, the $(32, 16, 8)$ extended and permuted BCH code (also a Reed-Muller code) has a 4-section trellis diagram of 64 states. It can be decomposed into 8 parallel and structurally identical 8-state subtrellises without cross-connections between them as shown in Figure 2. As a result, 8 identical 8-state Viterbi decoders can be devised to process the decoding in parallel. An IC implementation of a Viterbi decoder for this code using a 0.8 micron CMOS technology has been recently completed at the University of Hawaii VLSI Design Center. The decoder is implemented in Xilinx Field Programmable Gate Array (FPGA) chips [29]. The decoder is capable of operating at a speed of 200 Mbps. Custom design of this decoder using 0.5 micron CMOS technology can achieve a decoding speed of 600 Mbps or higher.

3. Trellises of BCH and RM Codes of Lengths 32, 64

Based on the theory developed in the previous section, an analysis of the parallel structure of the trellises for Reed-Muller (RM) and extended binary BCH codes was carried out. The degree of parallelism and the state complexity both depend on the sectionalization of the trellis. In general, it is known that as the number of sections decreases, the state complexity also decreases but the branch complexity in each section increases. We consider all possible uniform sectionalizations in which the number of parallel branches between two connected states is at most two. For example, we consider only 64-, 32-, 16- and 8-section trellises for the (64,42,8) RM code because the 4-section trellis has 32 parallel branches between any two adjacent connected states. The reason is that from an implementation and computational viewpoint, greater than 2 parallel branches between adjacent connected states

are disadvantageous.[2] The results of this analysis are presented in Table 4. Some surprising results are observed. A 32-section trellis for the (32,11,12) extended BCH code can be constructed which consists of 512 parallel 2-state subtrellises. An 8-section trellis for the (64,42,8) RM code can be constructed which consists of 128 parallel 64-state subtrellises. These results do not follow from the squaring construction for Reed-Muller codes or other previously published approaches. They also provide the designer a wide range of choices for trellises from which to choose. In the tables for each code and for each possible choice of the number of sections L, the logarithm to base 2 of the maximum number of parallel subtrellises that can be obtained without exceeding the number of states in the minimal trellis is denoted $P_{\max,L}$. The maximum state space dimension of the L-section subtrellis for the subcode C' is denoted $s_{\max,L}(C')$. The best known order of bit positions with respect to state complexity of BCH codes of length 64 presented in [12, 25] was used to produce the tables.

4. Issues in the IC Implementation of an L-section Trellis-Based Viterbi Decoder

In this section, five key factors affecting the decoding speed of a Viterbi decoder based on the minimal and non-minimal trellis are examined. The non-minimal trellis structure presented in this paper reduces the internal communication and allows independent parallel processing of the subtrellises while decreasing the complexity of a Viterbi decoder IC. We substantiate this claim through analysis in the following subsections.

4.1. Effective Computational Complexity of L-section Trellis

We consider a Viterbi decoder IC based on an L-section trellis with M bits per section for a $(LM, K, d_{\min})$ block code C. While many VLSI structures have been described for a Viterbi decoder [26, 27, 32], the most widely implemented structure is based on **add-compare-select circuits** (ACS) wherein each abstract state in the trellis diagram manifests itself as a physical ACS circuit on the IC and the same ACS's are repeatedly used for all depths in the trellis. The ACS's can be labeled ACS-i for $0 \leq i < 2^{s_{\max,L}(C)}$.

Let γ_i be the time required to process section-i of the trellis. At time $t = 0$, the metrics of the ACS circuits corresponding to the originating state of each parallel subtrellis are initialized to 0. After γ_1 units of time, at $t = \gamma_1$, the ACS-i corresponding to state s_i

---

[2]When there are exactly two parallel branches with complementary labels, the correlation metric for one branch is the negative of other and hence can be obtained by a mere sign inversion

10

at the end of section-1 for $0 \le i < |S_M(\mathbf{C})|$, has the metric of state $s_i \in S_M$. The index of the surviving branch into s_i is also stored in ACS-i. Continuing in this way, at time $t = \gamma_1 + \cdots + \gamma_l$, $1 \le l \le L$, ACS-i corresponding to state s_i, $0 \le i < |S_{lM}(\mathbf{C})|$, will have the metric of $s_i \in S_{lM}(\mathbf{C})$ and a sequence of l survivor branch indices corresponding to the most likely path from the originating state (of the subtrellis to which s_i belongs) to $s_i \in S_{lM}$.

There are as many ACS's as the maximum number of states at any depth in the L-section trellis for the linear block code. In the minimal trellis, whenever the decoder is processing the trellis at a depth at which the state size is less than the maximum state size, a number of ACS circuits are idle and the hardware utilization efficiency is poor. In the non-minimal L-section trellis, the utilization of the ACS circuits that exist in the IC is improved. Since all the subtrellis decoders operate independently in parallel, from the standpoint of speed, the effective computational complexity of decoding a single block (a received vector) is defined as the computational complexity of a single parallel subtrellis (viz. the minimal trellis for the subcode $\mathbf{C}'$) plus the cost of the final comparison among the choices (survivors) presented by each of the subtrellises. The time required for the final comparison is small relative to the time required for decoding a subtrellis and this comparison can be pipelined. Since subtrellises are processed in parallel, the speed of operation is limited only by the time required to process a subtrellis.

Note that both the minimal and non-minimal trellises require the same number of ACS circuits. However, the non-minimal trellis has a larger number of parallel subtrellises as compared to the minimal trellis (which often has none). Hence decoding using the non-minimal trellis with proper structure is faster compared to that using the minimal trellis. Therefore, a system bit rate specification which earlier could be met only by the use of some P number of Viterbi decoders operating simultaneously in parallel can be met with much fewer than P Viterbi decoders. In this manner, the effective computational complexity is a factor affecting the reduction in hardware complexity of an overall decoder.

4.2. Complexity of the ACS circuit

The converging branch profile (CBP) defined as the number of branches merging into a state at each particular depth also affects decoding speed and implementation complexity. This is called radix in IC literature. Let $\delta_{iM}(\mathbf{C})$, $1 \le i \le L$, be the CBP of the minimal trellis for $\mathbf{C}$ with trellis oriented generator matrix $\mathbf{G}$. At depth l, $1 \le l \le L$, the ACS circuits have to perform at least δ_{lM} stages of a tree type [33] two-way comparisons to find the best incoming branch. Hence reduction of the converging branch profile will improve the speed of decoding and reduce the complexity of each ACS circuit. We now show that none of the components in the CBP of the non-minimal trellis is increased. As will be shown by

11

examples in Section 5, most of the components of the CBP are decreased considerably.

Consider a non-minimal trellis for C obtained as the union of 2 parallel subtrellises each isomorphic to the minimal trellis for C', a subcode of C generated by $G - \{r\}$, $r \in G$. Let $\delta_{iM}(C')$, $1 \leq i \leq L$, be the CBP of the minimal trellis for C'. Recall that $s_l(C') = s_l(C)$ if $l \notin aspan(r)$ and $s_l(C') = s_l(C) - 1$, if $l \in aspan(r)$. By equation (2.2), $\lambda_{(i-1)M}(C') \in \{\lambda_{(i-1)M}(C), \lambda_{(i-1)M}(C) - 1\}$. By equation (2.1), $\delta_{iM}(C') > \delta_{iM}(C)$ only if $s_{(i-1)M}(C') = s_{(i-1)M}(C)$ and $s_{iM}(C') = s_{iM}(C) - 1$. But in this case, $(i-1)M \notin aspan(r)$ and $iM \in aspan(r)$. So $\lambda_{(i-1)M}(C') = \lambda_{(i-1)M}(C) - 1$. Therefore, $\delta_{iM}(C') = \delta_{iM}(C)$.

4.3. Traceback Complexity

Consider the problem of traceback to determine the best path through the trellis. In the minimal trellis, the ACS-i corresponding to state $s_i \in S_{lM}$ has to store $\delta_{iM}(C) + P_i(C)$ bits in order to identify which of the $2^{\delta_{iM}(C)}$ composite branches merging into s_i and which of the $2^{P_i(C)}$ parallel branches that form a composite branch survives. Therefore, in the minimal trellis, each ACS-i needs to store $\sum_{i=1}^{L}(\delta_{iM}(C) + P_i(C)) = \dim(C)$ bits in order to identify sequence of surviving incoming branches. In the non-minimal trellis, the storage in number of bits required for each ACS-i is $\sum_{i=1}^{L}(\delta_{iM}(C') + P_i(C')) = \dim(C')$ where C' is the subcode of C corresponding to the subtrellis. Since $\dim(C') = \dim(C) - P_{max,L}(C)$, the ACS's in the non-minimal trellis design require less storage than in the minimal trellis. The combined savings in storage in all the $2^{s_{max,L}(C)}$ ACS circuits is significant.

4.4. ACS-Connectivity

The basic operations performed by an ACS circuit are: addition of branch metrics of the incoming branches to the state metrics of the corresponding originating states, comparison of the resulting sums to find the best, selection of the surviving sum as the new state metric and the corresponding surviving branch label. The ACS-array architecture is dominated by the area required by the interconnections to transfer the state metrics [27]. For a state $s_i \in S_{lM}$, $0 \leq i < 2^{s_l(C)}$, $0 \leq l < L$, let $A_l(s_i)$ denote the set of states in $S_{(l+1)M}$ that are adjacent to s_i. Let $A_l(s_i) = \phi$ if $i \geq |S_{lM}|$. Then in the ACS-array implementation of the Viterbi decoder based on the minimal trellis, a path to transfer the state metric must exist between ACS-i and all ACS circuits that correspond to states in

$$A_0(s_i) \cup A_1(s_i) \cup \cdots \cup A_{(L-1)}(s_i)$$

The above set defines the **connectivity of ACS-i** in the ACS array corresponding to state $s_i \in S_{lM}$. The connectivity of the ACS's corresponding to states in the minimal trellis

12

results in a large amount of area in the VLSI chip being used for wiring [26, 27]. On the contrary, in the implementation of a Viterbi decoder based on the non-minimal trellis, the ACS circuits can be divided into blocks [31] such that the ACS's corresponding to states in a single subtrellis form a block. A particular ACS-i needs to transfer its metric only to a subset of ACS's within its own block. This reduced connectivity results in a reduction of hardware complexity and wiring area. The maximum connectivity of ACS-i is upper bounded by $s_{\max,L}(C')$ in the non-minimal trellis implementation.

4.5. Branch Complexity

The number of distinct branch metrics that have to be computed in section-i of the trellis is a property of the code and is **unaltered** by the parallelization of the trellis. Most IC decoders have a branch metric computational unit where all the branch metrics are calculated and then transferred to the ACS circuits [26, 27]. Because of the interconnection of branches between states in the trellis, routing the branch metrics to each of the ACS circuits requires a large amount of chip area. The trellises we describe show improvement over the minimal L-section trellis on this count because each subtrellis requires only a subset of the set of branch metrics in section-i of the trellis.

Parallelization of the minimal trellis as described in Section 2 may lead to a larger number of total computations being performed in decoding. The number of emanating branches in section-$(l + 1)$ is $|S_{lM}|2^{\lambda_{lM}}$ which may be larger than the corresponding product for the minimal trellis for some values of l, $0 \leq l < L$. However, as explained above, the hardware complexity of the decoder is not affected. We illustrate the reason with an example: The RM (64,42,8) code has a minimal trellis with the $s_{lM} + \lambda_{lM}$ sequence of $\{7, 13, 16, 16, 16, 16, 13, 7\}$. The same sequence for the non-minimal trellis is $\{13, 16, 16, 16, 16, 16, 16, 13\}$ which is larger at positions $\{0, 1, 6, 7\}$. Consider the case when $l = 1$ (other cases are similar). In section 2 of the minimal trellis, each of the 128 ACSs corresponding to states at the end of section 1 has 64 branches emanating from it. In section 2 of the non-minimal trellis, each of the 8192 ACSs has 8 branches emanating from it. Hence the number of operations performed per ACS are fewer in the non-minimal trellis. Hence larger values of $|S_{lM}|2^{\lambda_{lM}}$ represent larger number of operations performed simultaneously in parallel by all the ACS's in the non-minimal trellis.

5. Examples

Consider the (32,21,6) extended and permuted BCH code. The minimal 4-section, 8-bits/section trellis has SCP $\{0, 7, 9, 7, 0\}$. A non-minimal trellis 4-section trellis can be

obtained as the union of 32 parallel isomorphic subtrellises each having SCP $\{0, 4, 4, 4, 0\}$. Thus, Viterbi decoder implementations using the ACS-array architecture for both trellises will require 512 ACS circuits. However, in the minimal trellis, each ACS will require the capability of choosing the best among 64 incoming branches whereas the corresponding number is only 16 in the non-minimal trellis. The problem of routing state metrics is also much reduced since the connectivity of ACS-0 is 128 and that of ACS-i is at least 64 for $1 \le i \le 511$ while the maximum connectivity of any ACS in the non-minimal trellis is only 16. The structural parameters of each of these trellises are summarized in Tables 2 and 3.

Assuming each real number to be quantized to 8-bits the VLSI layouts of a radix-8 ACS and a radix-16 ACS were generated. A modified form of the bit-level pipelined ACS architecture [33] was used for the ACSs. The area required for the radix-16 ACS was 2.7 times that required for the radix-8 ACS. Assuming a factor of 2.5 increase in area per doubling of the radix, we see that 128 ACSs in the minimal trellis have an area 6.25 times larger than their counterparts in the non-minimal trellis implementation. The remaining ACSs require the same area. We see that the device area is reduced by adopting the proposed trellis architecture. Furthermore, the reduction in ACS-connectivity will yield significant reduction in wiring area. The savings in hardware complexity and increase in speed due to the non-minimal trellis approach easily overcomes the extra cost of the final comparison among the 32 choices, (1 from each of the subtrellises) to find the best codeword.

6. Trellises for a $(64, 40, 8)$ subcode of RM $(64, 42, 8)$

A $(64, 40, 8)$ subcode of the Reed-Muller $(64, 42, 8)$ code is proposed to NASA for usage as inner code in a concatenated coding system with the NASA standard $(255, 223, 33)$ Reed-Solomon code as outer code [31]. This RM subcode achieves a 5.3 dB coding gain over uncoded BPSK at the bit error rate of 10^{-6}. The required speed of decoding is 960×10^6 BPSK symbols per second which translates to an information bit rate of 600 Mbps. The coding gain is 0.5 dB less than the coding gain of a similar scheme with the same outer code but the NASA standard rate-1/2, 64-state convolutional code [34] as the inner code. However the $(64, 40)$ RM subcode has a higher rate of 0.626 bits/symbol than that of the convolutional code and thus requires lesser bandwidth. More significant is the fact that a Viterbi decoder for the $(64, 40, 8)$ inner code can be designed to operate at higher data rates than that for the convolutional code using the parallelism of the trellis of the RM subcode. The trellis for the NASA standard 64-state convolutional code does not consist of parallel subtrellises.

Let C denote the RM $(64, 42, 8)$ code and $\tilde{C}$ a $(64, 40)$ subcode of C. If the L-section

trellis on which decoding is based is composed of a union of P parallel isomorphic subtrellises then, the effective computational complexity denoted $A_{\text{eff}}(L)$ is merely that of a single subtrellis plus the cost of obtaining the final decision by comparing outputs of each of the P Viterbi decoders. The value of L which minimizes $A_{\text{eff}}(L)$ with the constraint that the L-section trellis $\tilde{T}$ have a maximum state complexity not greater than $s_{\text{max},L}(\tilde{C})$ (different from $s(C)$) is determined. Note that $s_{\text{max},L}(\tilde{C})$ is a function of the choice of the subcode and we will choose that subcode which has the least $s_{\text{max},L}(\tilde{C})$ for each L. The complexity of each addition, subtraction and comparison is assumed to be equal to one addition equivalent operation.

In the following, the trellis diagrams of various sectionalizations for this RM subcode are given. Their effective computational complexities are computed.

6.1. $L = 4, M = 16$

Let $C_0 = (16,15,2)$, $C_1 = (16,11,4)$ $C_2 = (16,5,8)$ be the corresponding Reed-Muller codes, G_i a generator matrix of C_i and $G_{i/j}$ a generator matrix for the set of coset representatives $[C_i/C_j]$. Let $\times$ denote the Kronecker product. For $L = 4$, the RM $(64,42)$ code has a minimal trellis corresponding to the 2-level squaring construction with a state complexity profile (SCP) $\{0,10,10,10,0\}$ ($s_{\text{max},4}(C) = 10$) and trellis oriented generator matrix

$$
G = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \otimes G_{0/1} +
$$
$$
\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \otimes G_{1/2} +
$$
$$
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \otimes G_2 \tag{6.7}
$$

In order to obtain a $(64,40)$ subcode $\tilde{C}$, one can delete any two of the 64 rows above giving a generator matrix for $\tilde{C}$. The maximum state space complexity $s_{\text{max},4}(\tilde{C})$ of the resulting code depends on which two rows we delete. It is easy to see that in order to have the least $s_{\text{max},4}(\tilde{C})$ which equals 8 we must delete any two of the 4 rows among $(1111) \otimes G_{0/1}$ obtaining an SCP of $\{0,8,8,8,0\}$ ($s_{\text{max},4}(\tilde{C}) = 4$). Using the theory developed earlier, it can be seen that we can obtain at most 4 parallel subtrellises in any 4-section trellis for $\tilde{C}$ without exceeding the allowable $s_{\text{max},4}$ of 8. The effective computational complexity may be computed to give $A_{eff}(4) = 39,682$ addition equivalent operations for the 4-section trellis.

6.2. $L = 8, M = 8$

Let $C_0 = (8,8,1)$, $C_1 = (8,7,2)$ $C_2 = (8,4,4)$ $C_3 = (8,1,8)$ be Reed-Muller codes. For $L = 8$, the RM $(64,42,8)$ code has a minimal trellis (with 2 parallel subtrellises) corresponding to the 3-level squaring construction with a SCP $\{0,7,10,13,10,13,10,7,0\}$ ($s_{max,8}(C) = 13$) with trellis oriented generator matrix

$$\acute{G} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & r_0^0 \end{pmatrix} \otimes G_{0/1} +$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & r_0^1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & r_1^1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & r_2^1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & r_3^1 \end{pmatrix} \otimes G_{1/2} +$$

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \otimes G_{2/3} + I_8 \otimes G_3 \qquad (6.8)$$

The $(64,40)$ subcode $\tilde{C}$ with the best SCP is obtained by deleting the rows $r_0^0 \otimes G_{0/1}$ and any one among the three rows $r_1^1 \otimes G_{1/2}$. This code $\tilde{C}$ has SCP $\{0,6,8,11,8,11,8,6,0\}$ ($s_{max,8}(\tilde{C}) = 11$). Repeating a similar analysis, it is seen that one can obtain at most 32 parallel subtrellises in any 8-section trellis for $\tilde{C}$ without exceeding the maximum allowable state space complexity of $s_{max,8}(\tilde{C}) = 11$. Each subtrellis has the SCP $\{0,6,6,6,3,6,6,6,0\}$ and from knowledge of its trellis structure the effective complexity is $A_{eff}(8) = 12,822$ addition equivalent operations.

6.3. L = 16, M = 4

Let $C_0 = C_1 = (4,4,1)$, $C_2 = (4,3,2)$ $C_3 = (4,1,4)$ $C_4 = (4,0,\infty)$ be Reed-Muller codes. For $L = 16$, the RM $(64,42,8)$ code has a trellis oriented generator matrix given by

$$G = G_{RM(16,5,8)} \otimes G_{1/2} + G_{RM(16,11,4)} \otimes G_{2/3} + G_{RM(16,15,2)} \otimes G_{3/4}. \qquad (6.9)$$

where $G_{RM(n,k,d)}$ denotes a trellis oriented generator matrix for the (n,k,d) Reed-Muller code. For $L = 16$, the RM $(64,42,8)$ code has a minimal trellis (with no parallel subtrellises) corresponding to the 4-level squaring construction with a SCP $\{0,4,7,10,10,13,13,13,10,13,13,13,10,10,7,4,0\}$ ($s_{max,16}(C) = 13$). The $(64,40)$ subcode

$\tilde{C}$ with the best SCP is generated by $\tilde{G} = G - \{r_1^1 \otimes G_{1/2}, r_2^1 \otimes G_{1/2}\}$ where r_1^1 and r_2^1 are the two rows with span $[2, 15]$ and $[3, 14]$ in the trellis oriented generator matrix for RM(16, 11, 4). The SCP of the minimal trellis for $\tilde{C}$ is $\{0, 4, 6, 8, 8, 11, 11, 11, 8, 11, 11, 11, 8, 8, 6, 4, 0\}$ ($s_{\max,16}(\tilde{C}) = 11$). By analysis, one can obtain at most 8 parallel subtrellises in any 16-section trellis for $\tilde{C}$ without exceeding the allowable $s_{\max,16}(\tilde{C})$ of 11. Each subtrellis has SCP $\{0, 4, 6, 8, 6, 8, 8, 8, 5, 8, 8, 8, 6, 8, 6, 4, 0\}$. The resulting effective computational complexity is $A_{eff}(16) = 23,174$ addition equivalent operations.

6.4. L = 32, M = 2 and L = 64, M = 1

When L = 32, $s_{\max,32}(\tilde{C}) = 12$. The maximum number of parallel isomorphic subtrellises possible without exceeding the allowable $s_{\max,32}(\tilde{C}) = 12$ in any 32-section trellis for the (64, 40) subcode $\tilde{C}$ is at most 4. So $A_{eff}(32) \geq 37,476$. When L = 64, $s_{\max,64}(\tilde{C}) = 12$. Furthermore, no parallel subtrellises are possible without exceeding the allowable $s_{\max,64}(\tilde{C}) = 12$. Hence $A_{eff}(64) = 198,000$.

From the above analysis, we see that the 8-section trellis for the (64, 40) RM subcode results in the least effective complexity. A VLSI implementation of a high-speed decoder for the (64, 40) RM subcode is under way. The decoder is based on the 8-section trellis which is a union of 32 parallel isomorphic subtrellises with a maximum of 64-states each. A schematic of the subtrellis is shown in Figure 3. Note that the last 4 sections of the subtrellis form a mirror image of the first 4 sections. This structure allows us to perform bidirectional decoding from both ends of the subtrellis simultaneously [10, 31, 35]. Sections 1 through 4 and sections 8 to 5 (in reverse order) are processed at the same time and path information corresponding to the most likely paths into the center 8 states which are the destination states are stored. The two path metrics (one from each side) at a center state are then added. This gives path metrics of 8 final survivors and the path with the largest path metric is the most likely path through the subtrellis. Since the resolution is done at the center of the subtrellis, the bottleneck of decoding caused by the large radix at the center states is avoided. This bidirectional decoding can be achieved by either using two identical subtrellis decoders working from both directions or using only one decoder to process the subtrellis in a concurrent bidirectional execution sequence as shown in Figure 4. The second approach simply exploits the use of pipelining in the ACS implementation and the mirror symmetry of the subtrellis about the center axis. The bidirectional decoding results in advantages in speed and implementation. A block diagram for the overall decoder is shown in Figure 5. We further note that sections 2, 3, 6 and 7 of each subtrellis decompose into 8 parallel, 8-state, fully connected isomorphic sub-subtrellises as depicted in Figure 3. This fact can be used to further reduce implementation complexity and increase the decoding speed.

7. Conclusion

We have presented an approach for decomposing the minimal trellis of a binary linear block code into a non-minimal trellis composed of parallel components. This approach allows parallel processing of the subtrellises and does not increase the maximum number of states. Hence it has significant speed advantage. In addition, it also reduces the IC area requirements. Given a linear block code, we have estimated the limits to the benefits of this approach and its dependence on the uniform sectionalization of the trellis. The branch complexity of the non-minimal trellis relative to the minimal trellis can be larger in some sections. However, this does not increase the hardware complexity. Since the application of this method depends only on the generator matrix of the code, it can be applied to arbitrary linear block codes.

References

[1] L.R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal Decoding of Linear Block Codes for Minimizing Symbol Error Rate," IEEE Transactions on Information Theory, Vol. 20, pp. 284-287, 1974.

[2] J.K. Wolf, "Efficient maximum likelihood decoding on linear block codes using a trellis," IEEE Transactions on Information Theory, Vol. 24, pp. 76-80, Jan. 1978.

[3] J.L. Massey, "Foundation and methods of channel encoding," Proc. Intl. Conf. Information Theory and Systems, NTG-Fachberichte, Berlin 1978.

[4] G.D. Forney,Jr., "Coset codes - Part II: Binary lattices and related codes," IEEE Transactions on Information Theory, Vol. 34, pp. 1152-1187, 1988.

[5] D.J. Muder, "Minimal trellises for block codes," IEEE Transactions on Information Theory, Vol. 34, pp. 1049-1053, Sept. 1988.

[6] Y. Berger and Y. Be'ery, "Bounds on the trellis size of linear block codes," IEEE Transactions on Information Theory, Vol. 39, 1993.

[7] T. Kasami, T. Takata, T. Fujiwara and S. Lin, "On the optimum bit orders with respect to the state complexity of trellis Diagrams of binary linear codes," IEEE Transactions on Information Theory, Vol. 39, Jan. 1993.

[8] T. Kasami, T. Takata, T. Fujiwara and S. Lin, "On complexity of trellis structure of linear block codes," IEEE Transactions on Information Theory, Vol. 39, No. 3, pp. 1057-1064, May 1993.

[9] T. Kasami, T. Takata, T. Fujiwara and S. Lin, "On structural complexity of the L-section minimal trellis diagrams for binary linear block codes," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E76-A, No. 9, pp. 1411-1421, Sept. 1993.

[10] T. Kasami, T. Takata, T. Fujiwara and S. Lin, "On branch labels of parallel components of the L-section minimal trellis diagrams for binary linear block codes," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E77-A, No. 6, pp. 1058-1068, June 1994.

[11] G.D. Forney, Jr. and M.D. Trott, "The Dynamics of Group Codes: State Spaces, Trellis Diagrams and Canonical Encoders," IEEE Transactions on Information Theory, Vol. 39, No. 5, pp. 1491-1513, Sept. 1993.

[12] A. Vardy and Y. Be'ery, "Maximum Likelihood Soft-Decision Decoding of BCH Codes," IEEE Transactions on Information Theory, Vol. 40, March 1994.

[13] G.D. Forney, Jr., "Dimension/Length Profiles and Trellis Complexity of Linear Block Codes," IEEE Transactions on Information Theory, Vol. 40, No. 6, pp. 1741-1751, Nov. 1994.

[14] G.D. Forney, Jr., "Dimension/Length Profiles and Trellis Complexity of Lattices," IEEE Transactions on Information Theory, Vol. 40, No. 7, pp. 1753-1772, Nov. 1994.

[15] G.D. Forney, Jr., "Trellises old and new," Communications and Cryptography, Edited by R.E. Blahut and D.J. Costello, U. Maurer and T. Mittelholzer, pp. 115-128, Kluwer Academic Publishers, 1994.

[16] Hari T. Moorthy and S. Lin, "On the Labeling of Minimal Trellises for Linear Block Codes," Proceedings of the International Symposium on Information Theory and Its Applications 1994, Vol. 1, pp. 33-38, Institution of Engineers, Australia.

[17] A. Lafourcade and A. Vardy, "Asymptotically Good Codes have Infinite Trellis Complexity," IEEE Transactions on Information Theory, Vol. 41, No. 2, pp. 555-559. March 1995.

[18] O. Ytrehus, "On the trellis complexity of linear block codes," IEEE Transactions on Information Theory, Vol. 41, No. 2, pp. 559-560, March 1995.

[19] Y. Berger and Y. Be'ery, "Trellis-Oriented Decomposition and Trellis Complexity of Composite-Length Cyclic Codes," IEEE Transactions on Information Theory, Vol. 41, No. 5, pp. 1185-1191, July 1995.

[20] F.R. Kschischang, and V. Sorokine, "On the trellis structure of block codes," IEEE Transactions on Information Theory, Vol. 41, No. 6, pp. 1924-1937, Nov. 1995.

[21] A. Lafourcade and A. Vardy, "Lower bounds on trellis complexity of block codes," IEEE Transactions on Information Theory, Vol. 41, No. 6, pp. 1924-1937, Nov. 1995.

[22] A. Lafourcade and A. Vardy, "Optimal Sectionalization of a trellis," submitted to IEEE Transactions on Information Theory, 1995, to appear.

[23] R.J. McEliece, "On the BCJR Trellis for Linear Block Codes," submitted to IEEE Transactions on Information Theory, 1995.

[24] T. Fujiwara, H. Yamamoto, T. Kasami and S. Lin, "A Recursive Maximum Likelihood Decoding Procedure for a Linear Block Code Using a Sectionalized Trellis Diagram and Its Optimization," (invited paper) Proceedings of The Thirty-Third Annual Allerton Conference on Communication, Control and Computing, Allerton House, Monitcello, Illinois, Oct. 4-6, 1995, also submitted to IEEE Transactions on Information Theory, special issue on Codes and Complexity, 1995.

[25] , T. Fujiwara, T. Kasami, R.M. Zaragoza and S. Lin, "The State Complexity of Trellis Diagrams for a Class of Generalized Concatenated Codes," submitted to IEEE Transactions on Information Theory, 1994. (in revision).

[26] P.J. Black and T.H. Meng, "A 140-Mb/s, 32-State, Radix-4 Viterbi Decoder," IEEE Journal of Solid-State Circuits, Vol. 27, Dec. 1992.

[27] P.G. Gulak and T. Kailath, "Locally Connected VLSI Architectures for the Viterbi Algorithm," IEEE Journal on Selected Areas in Communications, Vol. 6. pp. 526-537, April 1988.

[28] O. M. Collins, "The Subtleties and Intricacies of Building a Constraint Length 15 Convolutional Decoder," IEEE Transactions on Communications, Vol. 40, No.12, pp. 1810-1819, Dec. 1992.

[29] B.S. Vishwanath, "Soft-Decision Viterbi Decoding of the (32,16) Reed-Muller Code and Its VLSI Implementation," M.S. Thesis, Department of Electrical Engineering, University of Hawaii at Manoa, Aug. 1993.

[30] G. Fettweis and H. Meyr, "Parallel Viterbi Algorithm Implementation: Breaking the ACS-Bottleneck," IEEE Transactions on Communications, Vol. 37, pp. 785-789. Aug. 1989.

[31] S. Lin, G. T. Uehara, E. Nakamura and W. P. Chu, "Circuit Design Approaches for Implementation of a Subtrellis IC for Reed-Muller subcode ," NASA Technical Report No. 96-001, February 1996.

[32] H. Thapar and J. Cioffi, "A block processing method for designing high-speed Viterbi detectors," Proceedings of the ICC, Vol. 2. pp. 1096-1100, June 1989.

[33] A. K. Yeung and J. M. Rabaey, "A 210 Mb/s Radix-4 Bit-level Pipelined Viterbi Decoder," ISSCC 1995 Digest of Technical Papers, San Francisco, CA Feb. 1995.

[34] S. Lin and D.J. Costello, "Error Control Coding: Fundamentals and Applications," Prentice-Hall, 1983.

[35] M. Fossorier and S. Lin, "Coset Codes Viewed as Terminated Convolutional Codes," submitted to IEEE Transactions on Communications, February 1995 (revised June 1995).

Table 1: Set of row spans of trellis oriented generator matrix of $(32, 21, 6)$ extended and permuted BCH code

row-#	span	row-#	span
1	[1,8]	12	[12,26]
2	[2,15]	13	[13,20]
3	[3,13]	14	[14,22]
4	[4,14]	15	[15,27]
5	[5,12]	16	[17,24]
6	[6,18]	17	[18,31]
7	[7,21]	18	[19,29]
8	[8,25]	19	[20,30]
9	[9,16]	20	[21,28]
10	[10,23]	21	[25,32]
11	[11,19]		

Table 2: Parameters of 4-section trellis of $(32, 21, 6)$ extended and permuted BCH code

i	0	1	2	3	4
SCP	0	7	9	7	0
CBP	-	0	4	6	7
EBP	7	6	4	0	-

i =ACS-#	Connectivity of ACS-i
0	128
1 - 511	64

Table 3: Parameters of 4-section trellis of $(32, 16)$ subcode of the $(32, 21, 6)$ extended and permuted BCH code

i	0	1	2	3	4
SCP	0	4	4	4	0
CBP	-	0	4	4	4
EBP	4	4	4	0	-

i =ACS-#	Connectivity of ACS-i
0 - 511	16

Table 4: Trellises for all RM and BCH codes of lengths 32, 64

	No of Sections L		64	32	16	8	4	2
1	RM(32,6,16)	$P_{\max,L}(\mathcal{T})$		4	4	4	3	4
		$s_{\max,L}(\mathbf{C}')$		1	1	1	1	0
2	BCH(32,11,12)	$P_{\max,L}(\mathcal{T})$		9	9	9	7	
		$s_{\max,L}(\mathbf{C}')$		1	1	1	2	
3	RM(32,16,8)	$P_{\max,L}(\mathcal{T})$		5	4	5	3	
		$s_{\max,L}(\mathbf{C}')$		4	4	3	3	
4	BCH(32,21,6)	$P_{\max,L}(\mathcal{T})$		2	4	4	5	
		$s_{\max,L}(\mathbf{C}')$		8	6	6	4	
5	RM(32,26,4)	$P_{\max,L}(\mathcal{T})$		1	1	2		
		$s_{\max,L}(\mathbf{C}')$		4	4	3		
6	RM(64,7,32)	$P_{\max,L}(\mathcal{T})$	5	5	5	5	4	5
		$s_{\max,L}(\mathbf{C}')$	1	1	1	1	1	0
7	RM(64,10,28)	$P_{\max,L}(\mathcal{T})$	10	10	10	10	10	10
		$s_{\max,L}(\mathbf{C}')$	0	0	0	0	0	0
8	BCH(64,16,24)	$P_{\max,L}(\mathcal{T})$	14	14	14	12	13	14
		$s_{\max,L}(\mathbf{C}')$	1	1	1	2	1	0
9	BCH(64,18,22)	$P_{\max,L}(\mathcal{T})$	16	16	16	14	16	16
		$s_{\max,L}(\mathbf{C}')$	1	1	1	2	2	2
10	RM(64,22,16)	$P_{\max,L}(\mathcal{T})$	9	9	8	9	6	
		$s_{\max,L}(\mathbf{C}')$	5	5	5	4	4	
11	BCH(64,24,16)	$P_{\max,L}(\mathcal{T})$	11	11	10	11	8	
		$s_{\max,L}(\mathbf{C}')$	5	5	5	4	4	
12	BCH(64,30,14)	$P_{\max,L}(\mathcal{T})$	15	13	14	11	14	
		$s_{\max,L}(\mathbf{C}')$	6	7	6	7	4	
13	BCH(64,36,12)	$P_{\max,L}(\mathcal{T})$	10	9	10	9	8	
		$s_{\max,L}(\mathbf{C}')$	10	10	9	8	8	
14	BCH(64,39,10)	$P_{\max,L}(\mathcal{T})$	7	8	9	10	11	
		$s_{\max,L}(\mathbf{C}')$	13	12	11	9	8	
15	RM(64,42,8)	$P_{\max,L}(\mathcal{T})$	5	6	5	7		
		$s_{\max,L}(\mathbf{C}')$	9	8	8	6		
16	BCH(64,45,8)	$P_{\max,L}(\mathcal{T})$	2	3	4	4		
		$s_{\max,L}(\mathbf{C}')$	12	11	10	9		
17	BCH(64,51,6)	$P_{\max,L}(\mathcal{T})$	1	1	1	2		
		$s_{\max,L}(\mathbf{C}')$	11	11	11	10		
18	RM(64,57,4)	$P_{\max,L}(\mathcal{T})$	1	1	1	2		
		$s_{\max,L}(\mathbf{C}')$	5	5	5	4		

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}$$

Figure 1: Parity check matrix in trellis oriented form of the ex-BCH (32,21,6) code with an optimum order of bits with respect to trellis state complexity.

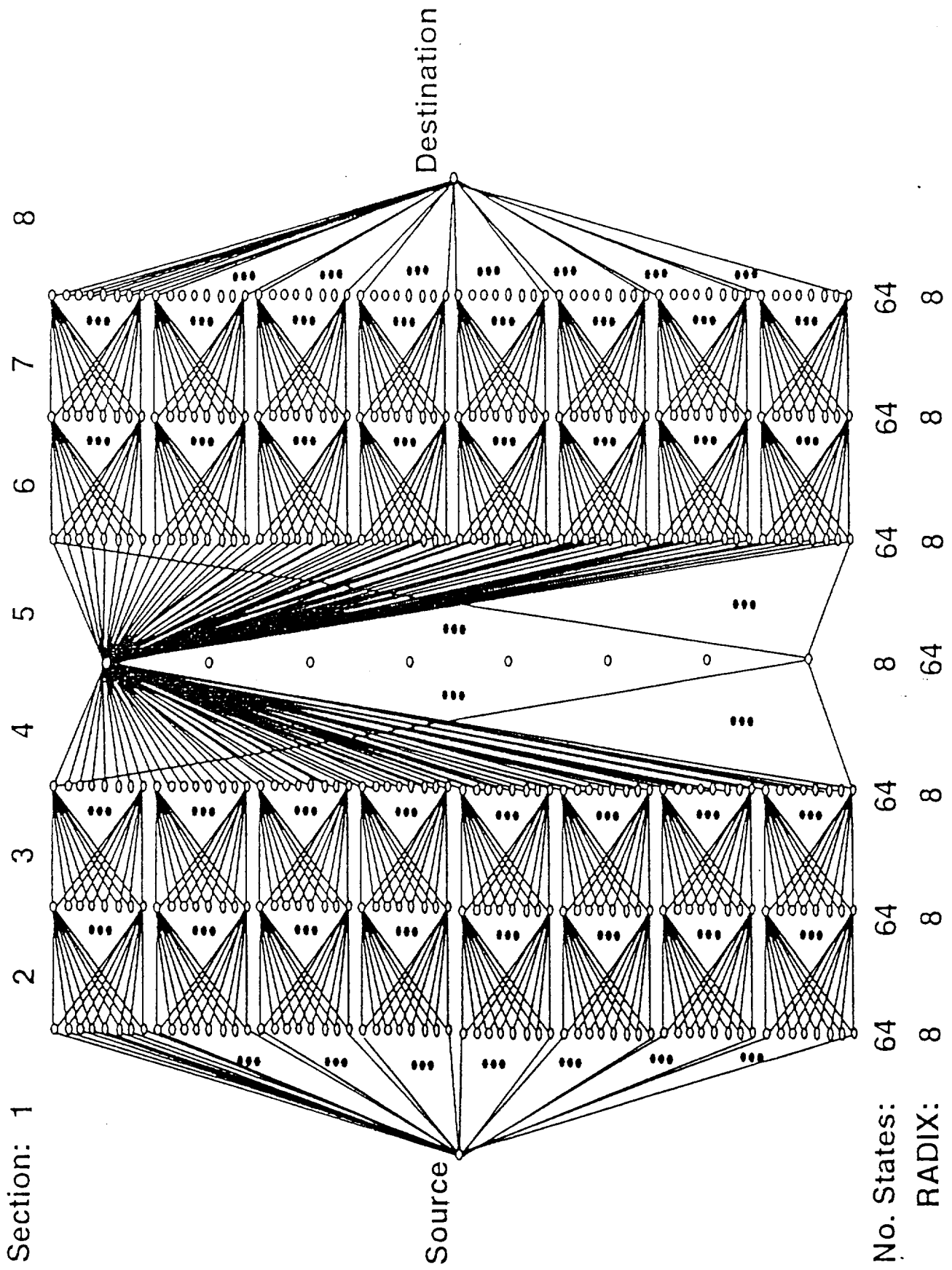Figure 2: A 4-section, 64-state trellis for the $(32, 16, 8)$ RM code composed of 8 parallel and isomorphic 8-state subtrellises

Figure 3: An 8-section, 64-state subtrellis for the $(64, 35, 8)$ subcode of the $(64, 40, 8)$ RM subcode

Sequence for Decoding ⟶ time

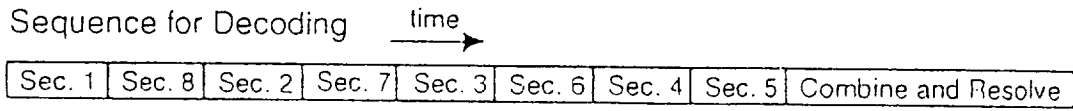| Sec. 1 | Sec. 8 | Sec. 2 | Sec. 7 | Sec. 3 | Sec. 6 | Sec. 4 | Sec. 5 | Combine and Resolve |

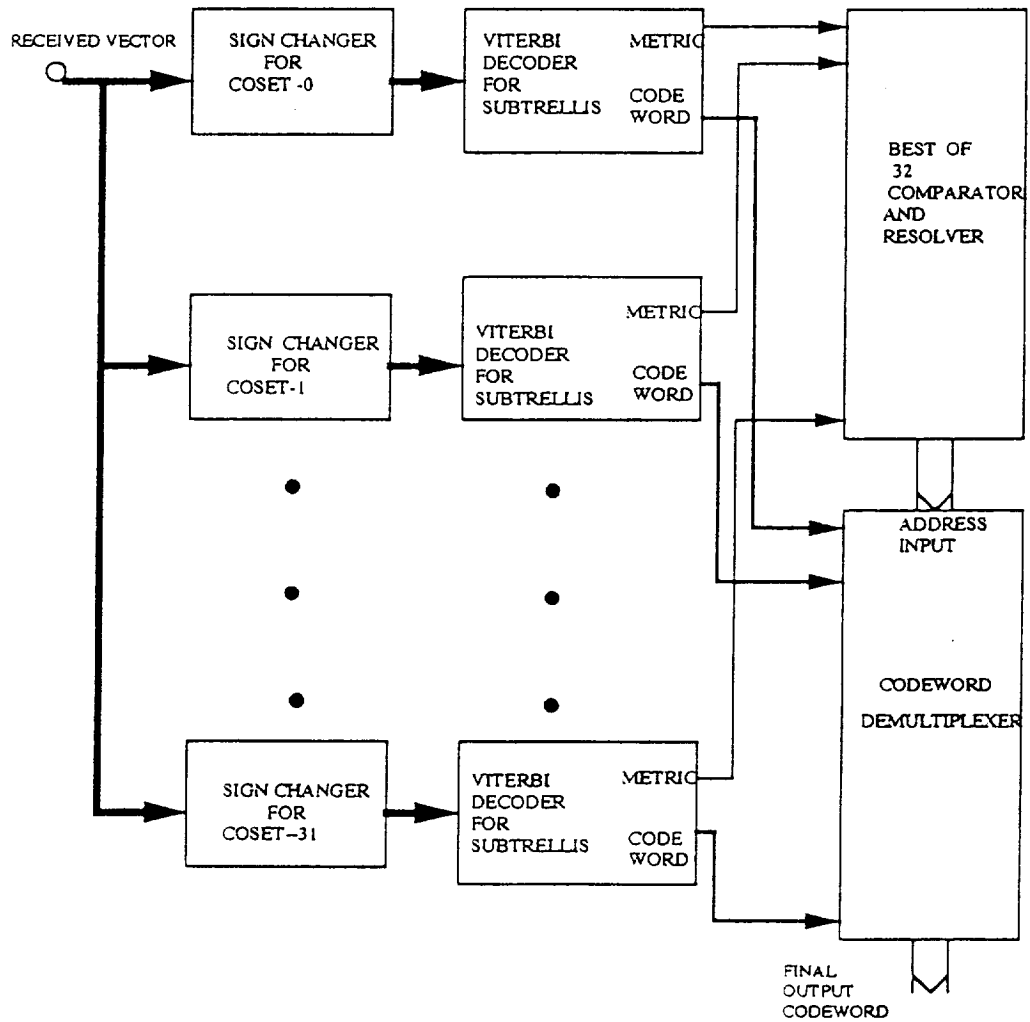Figure 4: Sequence for decoding using concurrent bi-directional execution sequence

Figure 5: Block diagram of overall decoder with 32 Viterbi decoders