

**ISSUES IN THE DEVELOPMENT  
OF A 10 MBPS K=15 VITERBI  
DECODER**

**William P. Osborne**

**February 1994 - June 1994**

**NASA Grant No. NAG 5-1491**

# **ISSUES IN THE DEVELOPMENT OF A 10 MBPS K=15 VITERBI DECODER**

**Semi-Annual Status Report**

**Center for Space Telemetry and Telecommunications Systems Grant**

**Period Covered: February 1994 through June 1994**

**NASA Grant No. NAG-5-1491**

**Principal Investigator: Dr. William P. Osborne**

**New Mexico State University**

**Electrical and Computer Engineering  
Box 30001 - Dept. 3-O**

**Las Cruces, NM 88003**

## TABLE OF CONTENTS

1. Introduction	1
1.1 Usability and Performance of the BVD	1
2. BVD Specifications	4
2.1 Processor Specifications	4
2.2 The Interconnection Problem	7
3. Packaging Considerations	13
3.1 Capacitance and Power Estimates	16
3.2 Description of the Packaging Options	18
4 Conclusion	18

## 1. Introduction

The NMSU Telemetry Center, in collaboration with the NASA Microelectronics Center (MRC) at UNM, has completed a study of the feasibility of building a constraint length 15, rate 1/2 Viterbi Decoder (or BVD) to operate at a rate of 10 Mbps. The BVD, if built, would make TDRSS more accessible to all users, small satellites in particular, by providing an additional 2 dB of link margin, relative to the use of the standard constraint length 7 decoder. The study included the following:

1) Review of the 1 Mbps BVD built by the Jet Propulsion Laboratory (JPL), currently the only BVD in existence.

2) Development of Specifications for the basic processing unit of the BVD (MRC).

3) Analytical Determination of performance of large constraint length convolutional codes.

4) Investigation of the impact of processor design on the overall system design;

5) Two feasible packaging technologies, proposed by Cincinnati Electronics of Ohio.

It was concluded that while the construction of the BVD is feasible, it will require the most advanced packaging technology currently available, and that the project would be best accomplished in an industrial facility. While the size, complexity, and power requirements of the BVD will be extreme, these will impact only the ground station. The spacecraft will incur a minor change in the encoder design, and the increased coding gain will allow a satellite to operate with a smaller antenna.

### 1.1. Usability and Performance of BVD

The BVD would be built to be electrically compatible with the existing interface for the standard constraint length 7 Viterbi Decoder, and be installed in the communication system as shown in figure 1. The Viterbi Decoder decodes convolutionally encoded data to provide forward error correction. When used in a downlink, the convolutional encoder, a simple device, is located in the spacecraft, while the Viterbi Decoder, a complex device, is located at the ground terminal. Figure 2 shows the standard constraint length 7 convolutional encoder, consisting of a 7-stage shift register, and two parity checks. Figure 3 shows the constraint length 15 convolutional encoder, consisting of a 15-stage shift register and two parity

checks. This is the only modification to spacecraft hardware required by the BVD.

The bit error rate performance of the BVD was predicted using the bit error spectrum (BES) algorithm. Discussions of this algorithm are found in the literature [1,2], and the algorithm was implemented in the "C" programming language at NMSU. The BES will produce an upper bound on the probability of error for convolutional codes. The true bit error rates converge to the upper bound at high signal-to-noise ratios, and the BES is also useful for comparing the relative performance gains of various codes. The results are shown in figure 4. The standard constraint length 7 (64-state) code provides a coding gain of 5.6 dB over uncoded BPSK. The BVD provides an additional 2 dB of coding gain over the standard code. This translates directly into a 2 dB savings in transmitter power (or antenna size) in the spacecraft.

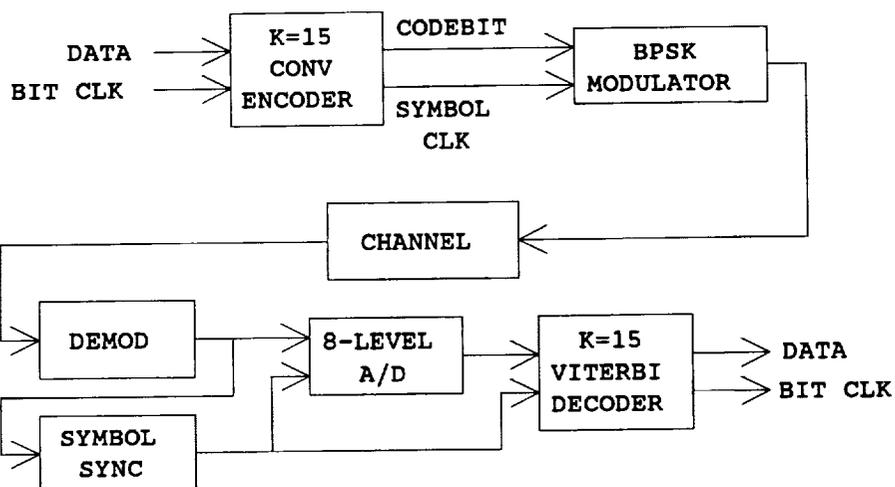


Figure 1. System with K=15 Viterbi Decoder

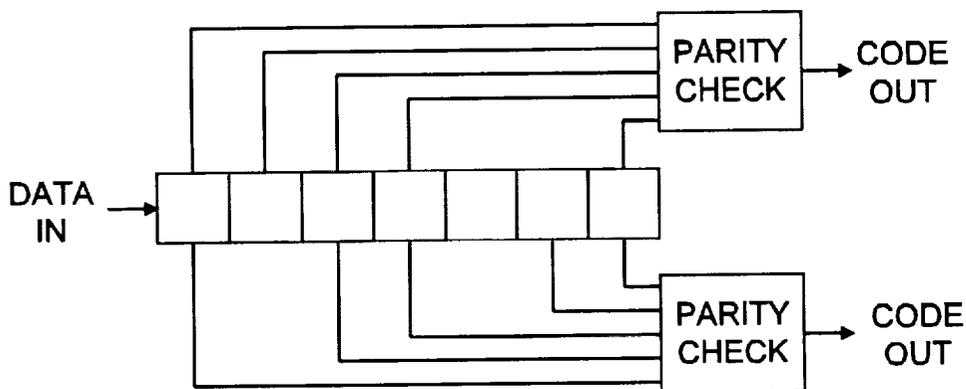


Figure 2. K=7 Convolutional Encoder

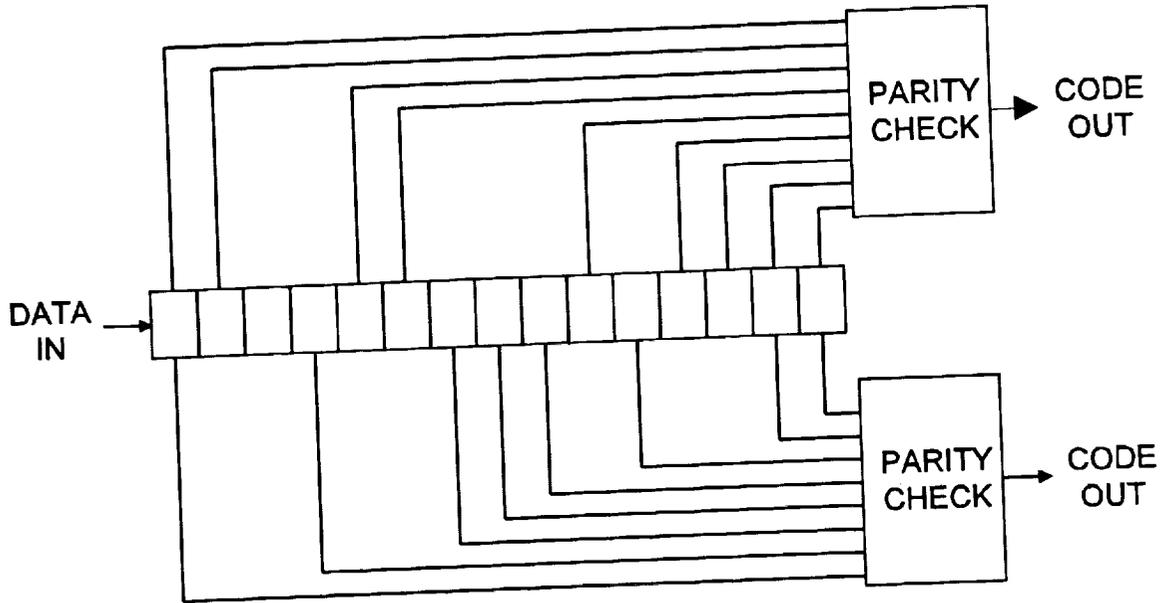


Figure 3. K=15 Convolutional Encoder.

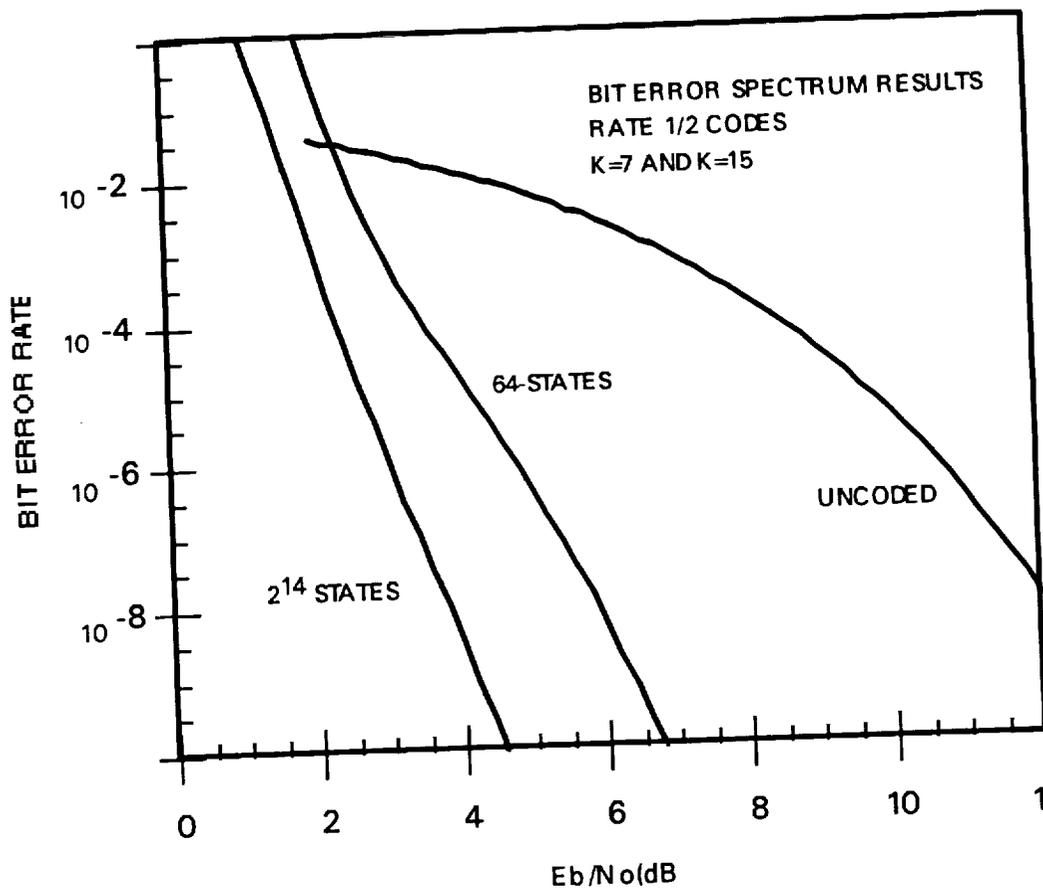


Figure 4. Analytical Calculation of Bit Error Rate of K=15 code.

## 2 BVD Specifications.

The BVD is to decode a rate 1/2 constraint length 15 ( $2^{14}$  state) convolutional code using the full Viterbi algorithm as opposed to sequential decoding. The BVD will be programmable to allow for different polynomials but it is the best known generator polynomials are:

$$G(d) = \begin{aligned} & [1+d^2+d^3+d^4+d^6+d^7+d^8+d^{10}+d^{14}], \\ & [1+d+d^5+d^7+d^8+d^{11}+d^{13}+d^{14}] \end{aligned}$$

Additionally, the BVD is to use 8-level soft decisions and a traceback depth of 70.

The decoder can be constructed from 128 0.8 micron CMOS chips with system clock of 35-45 MHz. Given a faster clock, the system could be built with fewer chips, each chip having more processors but the same number of I/O's. Increasing the number of I/O's is not recommended, as this would make the chip too large. Input data, representing 8-level received symbols will be clocked into the decoder at a rate of 20 MHz, for a decoded data rate of 10 Mbps.

### 2.1. Processor Specifications.

The specifications for the processor chips have been drafted by the MRC, and are as follows:

A radix-16 node processor scheme was chosen to implement the basic metric calculation. Each processor operates in a bit-serial manner, forward propagating the 16 computed metrics along 16 separate bit serial lines to the appropriate 16 destination processors, as shown in figure 5. Each IC is to contain 8 independent radix-16 processors thereby supporting 128 nodes, and requiring 128 such chips to implement the entire BVD. A radix 16 implementation was chosen because it allows the metric information to be updated once every 4 information symbols. This significantly reduces the data rate of the metric communication channels and allows this information to be transferred bit serially between processors. A radix larger than 16 was not chosen because the chip area required to implement a radix 32 processor is too large. The number of node processors that can be implemented on a single chip is still limited by the number of I/O pins that can be reasonably placed around a single die in the radix-16 configuration. The limiting factor becomes the actual computation circuitry when considering higher radix processors, for an only marginal improvement in data flow.

The I/O pins for the processor chip are detailed in the appendix.

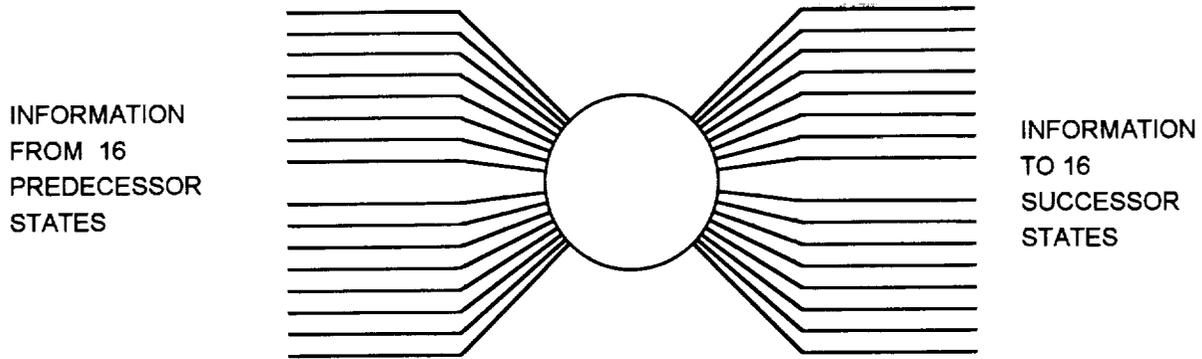


Figure 5. Radix-16 processor.

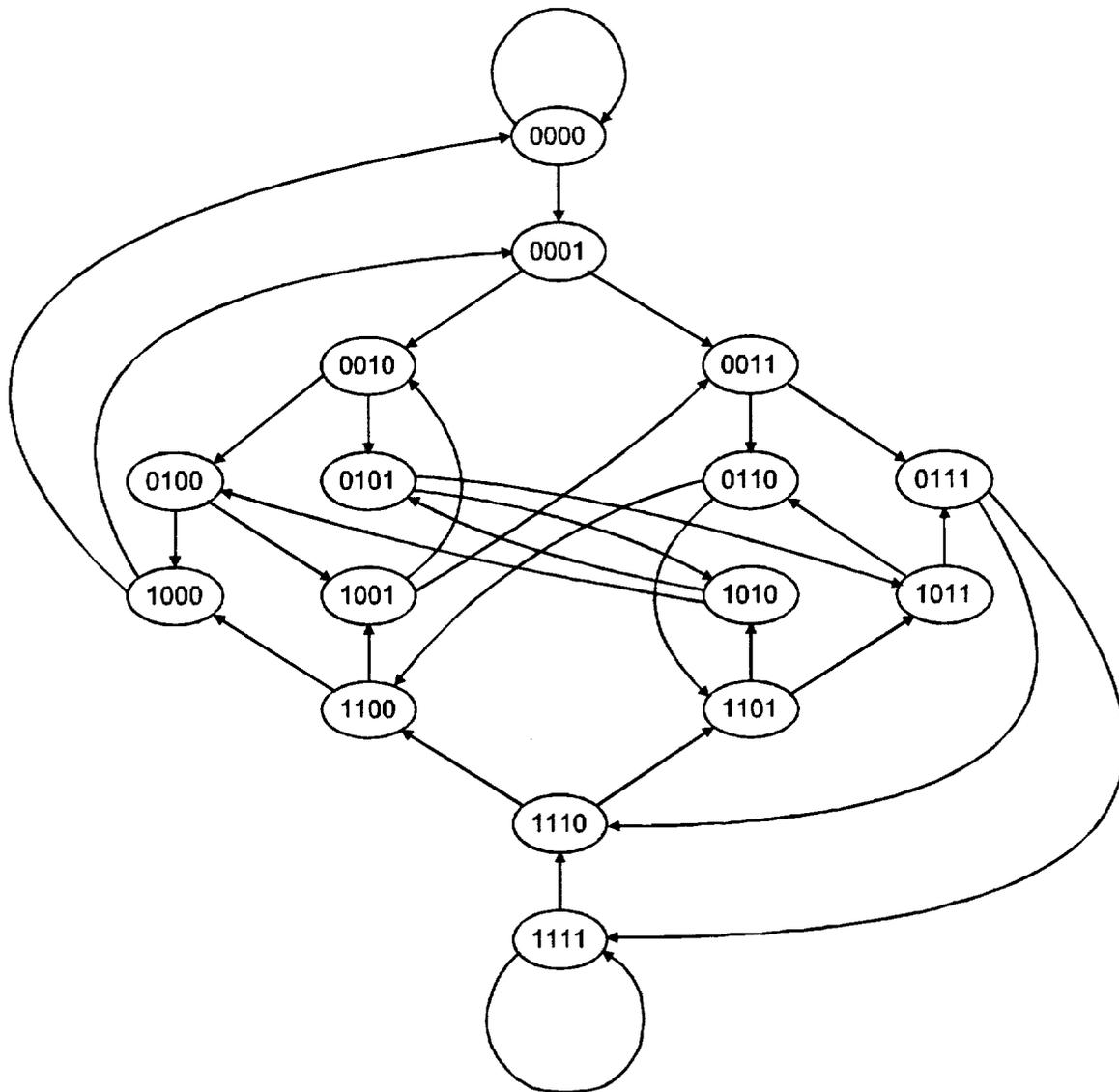
## 2.2. The Interconnection Problem

The function of the Viterbi Algorithm is to determine the most likely state history of the convolutional encoder. To do this, the decoder must maintain a "metric" function, which indicates the likelihood that the decoder is in a given state at a given time. Each state of the code will have a processor dedicated to the task of continuously updating the metric function for that state. Updating the state metric requires that each processor receive metric information concerning predecessor states, and pass on information to processors responsible for successor states. The consequence of this is that the architecture of the Viterbi decoder is dictated by the state diagram of the convolutional encoder. Thus, if we look at a typical state diagram, each node (state) represents a processor, and each line represents a connection between processors.

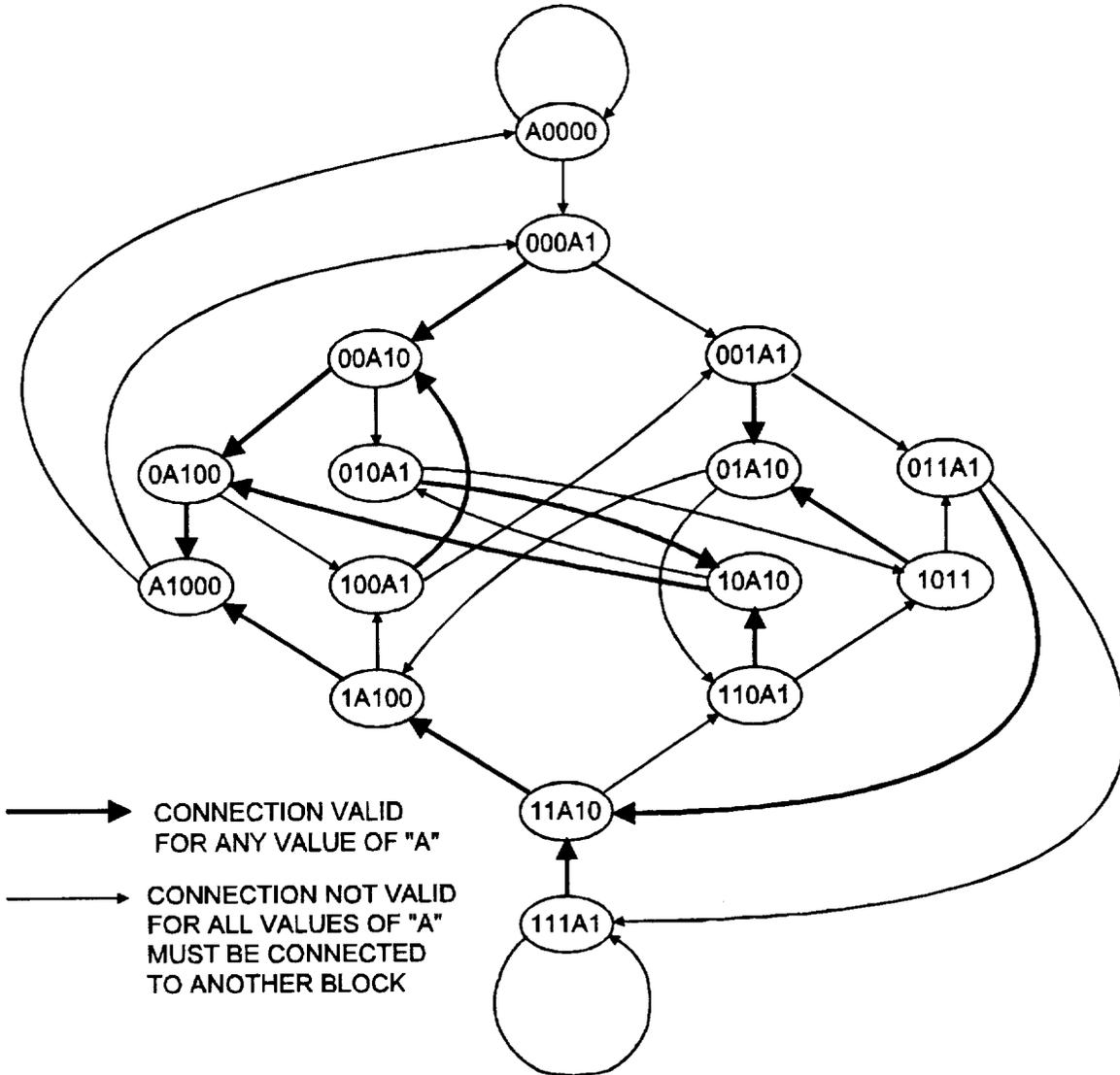
The state register of the convolutional encoder is in fact a shift register, and therefore, the state transitions are generated by shifts of the shift register. Figure 6 shows a state diagram for a 16-state convolutional encoder. The states are labeled with binary numbers, and a state transition is generated by making left shift. The leftmost bit is shifted out, and a "0" or "1" is shifted into the right-most position.

When the number of states becomes so large that not all of the processors will fit onto one VLSI chip, the state diagram must be subdivided. Oliver Collins of JPL has derived a method for partitioning large state diagrams and applied it to the construction of a K=15 decoder at JPL [3,4]. According to Collins' method, one can construct an M-state state diagram using M/N identical N-state building blocks derived from the state diagram of an N-state machine. To do this, an additional symbol "A" is inserted into each state label of the N-state diagram, according to certain rules. The symbol "A" represents any number, k, of bits,

and allows the N-state diagram to become one of  $2^k$  building blocks in a  $2^k \cdot N$  state diagram. When the symbol "A" is inserted, some of the connections in the state diagram will be valid for all values of "A", and may be hardwired on the building block, while other connections are not valid for all values of "A", and represent connections which must be made to other building blocks. Figure 7 shows a 16-state state diagram modified according to Collins' method. Connections valid for all values of "A" are shown with bold lines. As can be seen, this partitioning allows slightly less than half of the connections to be hardwired on the building block, the remaining connections must be made between blocks. Collins' method allows greater efficiencies to be obtained for building blocks of greater numbers of nodes.



**Figure 6.** State Diagram for 16-state convolutional encoder.



**Figure 7.** 16-state state diagram with symbol "A" inserted.

The method for inserting the symbol "A" into the state labels is as follows:

- 1) A set of sub-strings, referred to as a cover, is chosen such that every state label in the N-state state diagram includes an element from the cover.
- 2) The symbol "A" is inserted immediately before the last occurrence of an element from the cover.
- 3) Collins has shown that the inclusion of an s-bit element in the cover causes the fraction  $2^{-s}$  of the number of connections in the N-state state diagram to be non-applicable to the universal building block. The objective

is to find the cover which maximizes the number of universal connections.

In the example of figure 7, the cover is {1,0000}. The symbol "A" is inserted immediately before the last occurrence of "1" or "0000", and the fraction of connections which are valid for the universal building block is equal to  $1 - (1/2) - (1/16)$  or  $7/16$ .

Collins' method was applied to the construction of a radix-2 K=15 decoder at JPL. For the radix-16 implementation, Collins' method is applicable, but less powerful. This is because the number of required connections is much larger, while the fraction of connections which are applicable to the universal building block is much smaller. The term radix-k simply means that each state has k successors and k predecessors, so that the simplest implementation is radix-2. Radix 4, 8, and 16 implementations are formed by letting the state diagram show transitions made in 2, 3, or for steps, respectively. This implementation results in a speed advantage, in that the decoder can process more than one symbol per cycle (4 in the case of radix 16), but increases the interconnection complexity. The effect of radix greater than two is shown by figures 8 and 9, which show 8-state state diagrams of radix 2 and 4 respectively. As can be seen, even radix 4 doubles the interconnection density over radix 2.

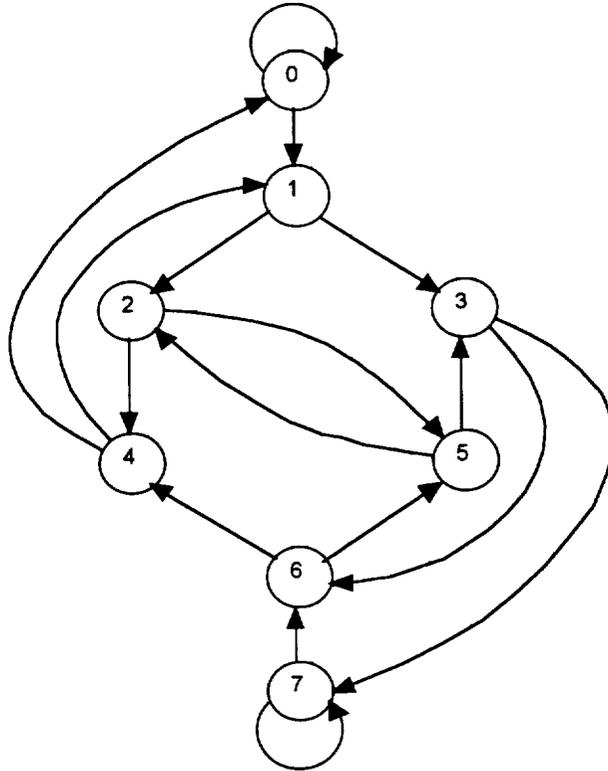
To apply Collins' method to a radix-16 implementation, either of two approaches could be used:

- 1) Replace the binary digits with hexadecimal digits, and extend the involved operations to the hexadecimal numbering system

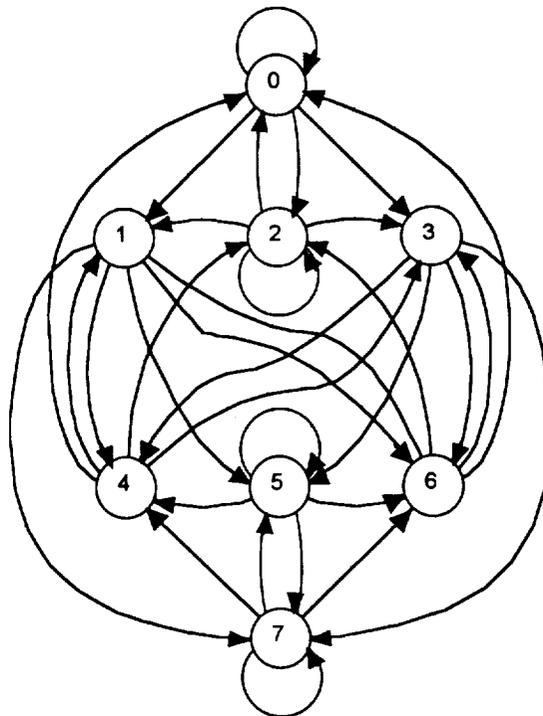
or

- 2) Keep the binary system and treat the radix 16 branch as a sequence of four binary branches.

In the first approach we apply hexadecimal equivalents of the binary operations used in Collins original formulation. This means that each s-digit element included in the cover would cause the fraction  $16^{-s}$  connections to be non-universal, yet more elements would be required in the cover, to the extent that the attainable efficiency is reduced. Furthermore, in partitioning the radix-2 state diagram, Collins first obtains an immediate two-fold reduction by combining the states into pairs of states having the same predecessors. For the radix-16 implementation, the analogous step has already been taken, as the MRC specification calls for processors designed to



**Figure 8.** Radix-2 8-state state diagram.



**Figure 9.** 8-state radix-4 state diagram.

handle 16-nodes. These processors, of which there are 1024, are the elements to be used in the subdivision.

NMSU has accomplished the generalization of Collins method to radix 4, 8, and 16, and developed a program which finds the covers for each of these cases. Table I shows the results found by the program. In the table, a "unit" refers to a collection of nodes between which it is desired to minimize the inter-connections. This could be an integrated circuit chip, or also a board consisting of IC's. As mentioned, Collins' method requires that the nodes must first be grouped into processors (In the radix-2 implementation, Collins refers to them as Butterflies) each of which handles a number of nodes equal to the radix of implementation. Furthermore it is plain from the method of labeling the processors, that the number of processors per unit must be a power of the radix. The wiring requirement of each processor is one incoming wire and one outgoing wire per node. It is in terms of this baseline that the reduction obtained by the algorithm is expressed.

STATES PER UNIT	RADIX			
	2	4	8	16
8	0.25			
16	0.375	0		
32	0.4375			
64	0.5625	0.25		
128			0	
256		0.40625		0
512			0.25	
1024				
2048				
4096				0.25

**Table I.** Efficiencies obtained by applying Collins' reduction method to radix  $\geq 2$ .

As an example, if we are operating in radix-4 we have a processor representing four states. If we place four such processors in a group, the group represents 16 states, but no reduction is possible. With 16 such processors, the processor labeling, in base four, is 0, 1, 2, 3, 10, ...33, and we obtain a reduction of 0.25 using a cover of {0, 1, 22, 23, 32, 33}. In radix 16 implementation, we obtain 256 states using 16 processors, but no reduction is possible. Using 256 processors per unit, we obtain 4096 states per unit, and 4 units are needed for the required 16,384 states.

The algorithm allows a reduction of 0.25, meaning that 2048 I/O wires are required for the metric information, and additionally, 21 control wires.

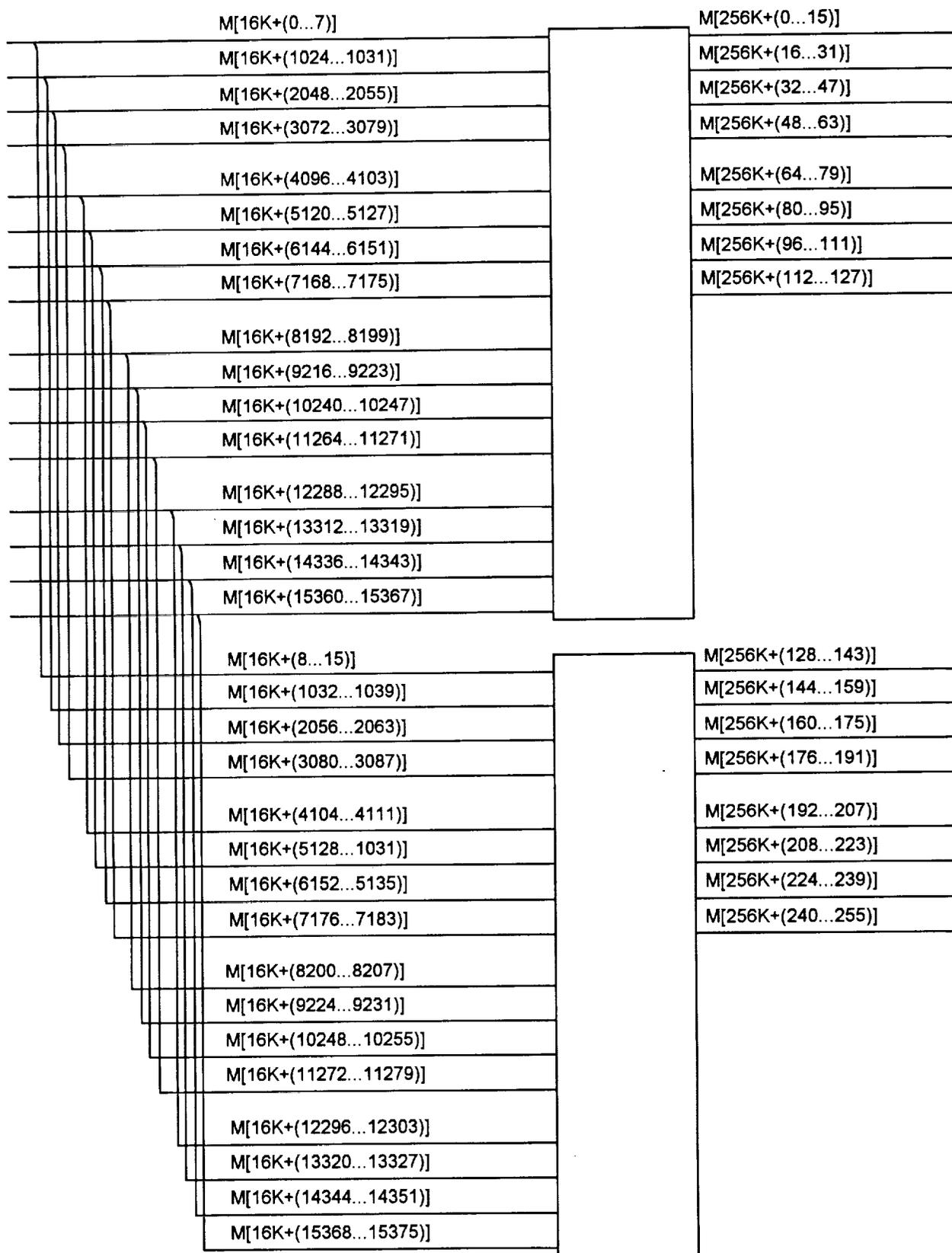
If the other approach is taken, each hexadecimal branch is looked at as composed of four binary branches. In this case, the hexadecimal branch is non-universal if one or more of the binary branches is non universal. Collins defines the efficiency  $E$  as the fraction of branches which are universal. If a given binary cover results in an efficiency of  $E$ , the same cover used in the hexadecimal system results in an efficiency of  $E^4$ . With 4096 states in a building block, there are 256 radix-16 processors. For 256 processors, the best cover found by Collins gives an efficiency of 0.719 in radix 2, which translates to 0.267 in radix 16, a slight gain for using the more exacting approach. However, with the BVD system consisting of 4 modules, 25% of the connections will be on the same module anyway.

### 3 Packaging Considerations

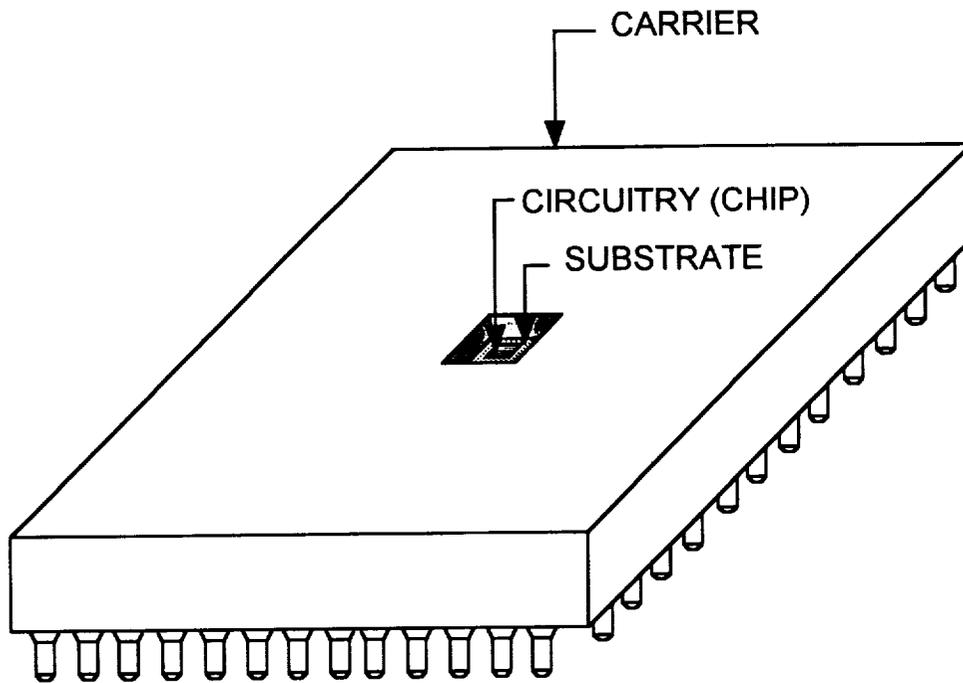
Using the chip specifications of UNM, a single chip provides 128 processors, each handling the operations associated with 16 nodes. If we pair off the chips, each pair consists of 16 processors, and 256 metric lines leave the pair, while 256 arrive at the pair. The UNM chip can be programmed to select the processors on the chip. If the nodes are selected appropriately, it is possible to bus the wires in groups of 16. Although this does not reduce the physical density of the BVD system, it does simplify the layout.

Figure 10 illustrates how this is accomplished. Here,  $K$  is an integer between 1 and 256, which uniquely identifies the chip pair. The nodes assigned to the first chip of the pair should be nodes  $256K$  through  $256K+127$  and the nodes assigned to the second pair should be  $256K+128$  through  $256K + 255$ .  $M[x]$  would be a wire which carries the metric information from state  $x$ .  $M[x_0...x_n]$  represents a bus of such metrics.

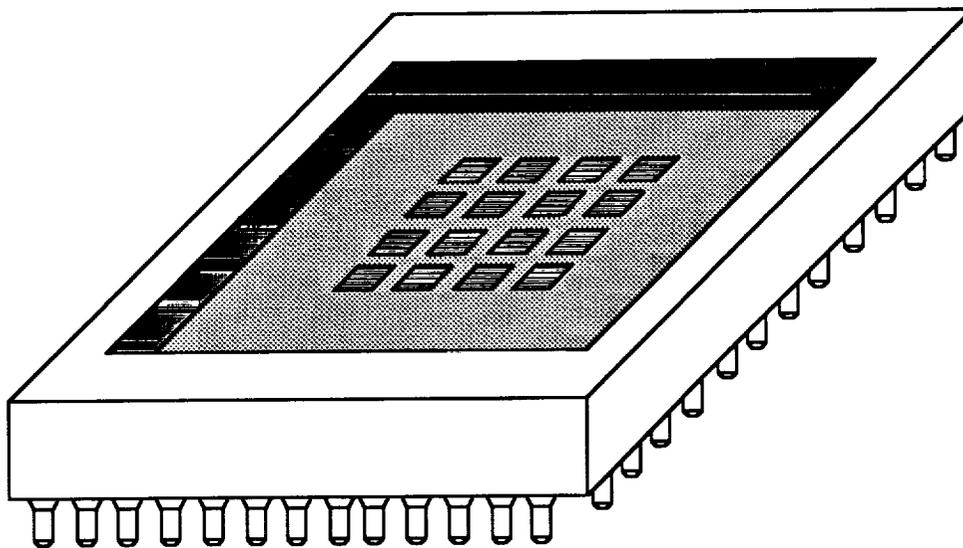
With 256 such pairs in the entire system, the wiring density of the BVD system becomes extreme. It could be stated that the area required for the logic itself is



**Figure 10.** Proposed Grouping of Processors.



**Figure 11.** *Conventional VLSI module.*



**Figure 12.** *Multi-Chip Module.*

relatively insignificant compared to the volume required for wiring. The size of the chip is in fact dictated by the I/O requirements. The system requirements of the BVD will stress the limits of conventional printed circuit board technology, and it is strongly recommended that multi-chip module technology be used for the BVD. The report from Cincinnati Electronics presents two feasible packaging

options, both of which call for the most advanced technology available.

The difference between multi-chip and conventional technology is shown by figures 11 and 12. In conventional technology, the actual circuitry occupies a very small percentage of the volume of the carrier. In multi-chip module technology, multiple chips are installed in a single carrier. One thing which is certain is that the technology chosen for the BVD must allow for the installation of custom designed chips. Packages built for microprocessors will probably not meet the speed requirements of this project.

### **3.1 Capacitance and Power Estimates**

When the architectural requirements of the BVD were determined, Cincinnati Electronics (CE), an engineering firm with experience in electronic packaging was consulted to determine feasible packaging options for the system. CE has identified two options, which are described in a separate report. To specify the project, NMSU developed a set of schematics (which are still on file) showing the chip pairs and the 16-wire busses described in the previous section. The schematics were sent to CE with the following description:

The Big Viterbi Decoder consists of 128 chips. Each chip consists of 276 pins, of which 20 are global control functions, which are connected in parallel to all 128 chips, and the remaining 256 pins are data lines which must be interconnected between the chips in a very specific manner.

The control connections are not shown on the schematics as they are not the main concern. However, we are very concerned about the feasibility of making the data connections. To make the data connections, the 128 Viterbi Decoder Chips have been grouped into 64 pairs. The 64 pairs are interconnected by busses of 16 lines each.

The schematic consists of 5 sheets. The first sheet shows how two chips are interconnected into a pair. The remaining four make up the top level diagram, which shows the 64 pairs interconnected by 16 line busses.

Given the schematics, CE estimated the dimensions of the BVD device, and provided an order-of-magnitude, worst case estimate of the capacitance that the pins of the BVD

chips would be required to drive. This is a significant and important problem for the following reasons:

1) When the processor chips are mounted, and connected in the way that the constraint length 15 code requires, the length of the wires becomes extremely long by IC standards, approximately 10". This means that the trace length capacitance can by no means be neglected, and will impact the ability of the device to meet its speed requirements (40 MHz system clock)

2) Trace length capacitance is needed to determine the power requirements of the processor chips, which affects the size of the line drivers.

3) The trace length and power requirements are mutually compounding problems. That is, longer wires require more power, which would require larger line drivers, and hence larger chips. The larger chips would in turn increase the physical dimensions of the system, and demand longer lines.

For the capacitance estimate, a worst-case chip size estimate of 0.5" by 0.5" was used. This is the requirement for peripheral interconnects, and could possibly be reduced by the use of pin-grid array interconnects. Additionally, CE has stipulated the following assumptions for the capacitance estimate:

1) Trace-to-trace capacitances are assumed to be negligible.

2) Traces are assumed to be over a continuous ground plane. (Could be metal package, etc.)

3) A trace width of 10 mil. Note, smaller trace widths will probably be required for this project, however, this will provide a worst case area.

4) Dielectric constant (K) of 9, this is consistent with MCM-C technology, and probably worst case.

Based on these assumptions and modeling the traces on the hybrid as a rectangular parallel plate capacitor yields 101.2 pF for a maximum trace length of 10".

Given the line capacitance, the MRC has provided estimate for the processor chip, using the fact that the energy required for a change in logic state is:

$$E = \frac{1}{2}CV^2$$

and a clock rate of 40 MHz. The worst case value would be 12.8W. Using the fact that approximately half of the lines will actually change state on any given cycle, and that the average trace length is approximately 1/3 the maximum trace length, a typical power estimate would be 2.112W. Using 3V logic, the power could be further reduced to 0.760W. This is the power for the line drivers. Using 2.4W for the power of the chip logic itself, the power comes to:

$$P(3V \text{ supply}) = 3.16W$$

$$P(\text{typical}) = 4.51W$$

### **3.2 Description of the Packaging Options**

Cincinnati Electronics has recommended 2 packaging options:

- 1) Planar High Density Interconnect Thin Film Module Approach
- 2) 3-D Chip on Board Vertical Integration

Approach #1 offers the advantage of a more compact package, whereas approach number 2 offers the advantage of easier troubleshooting. This is because the first approach requires distributed array chip interconnects, while the second allows peripheral interconnects. Both approaches use 4 multi-chip modules (MCM) of 32 chips each. The first approach uses Bump-Bonded Flip Chips, close mounted heat sink, and ball grid array intermodule connections. Approach #2 uses tape automated bonding (TAB) circuit cards. The cards are stacked, and each card provides vias for inter-card connections. With this approach the circuit would be cooled by circulating air between the cards. With the first approach the dimensions of the MCM would be 10" by 5", with the second, 13.850" by 7.140". Both use nearly all of the area of the MCM for module interconnects. The packaging options are described in more detail in the report by Cincinnati Electronics.

### **4 Conclusion**

While the construction of the 10 Mbps Big Viterbi Decoder is feasible, it is likely to require sophisticated industrial facilities and the most advanced packaging technology available at this time. The technology required to build a 10 Mbps BVD was not available when JPL built the 1 Mbps BVD. The JPL BVD pushed the limits of the technology available at that time, and would have been impossible to build had it not been for Collins' reduction method. The 10 Mbps BVD calls for radix-16 implementation, which increases the interconnection density by a factor of 4, and reduces the gains which can be obtained by the reduction method. It

is essential that the 128 processor chips be identical and individually testable. Testing and trouble shooting of the system will be laborious tasks, and the project will be expensive. However, MCM technology is advancing rapidly, and it is possible that the project may become less expensive in the future. Furthermore, the construction of one Big Viterbi Decoder to be installed at the ground terminal would increase the link margin of all current and potential TDRSS users.

#### REFERENCES

- [1] M. Cedervall and R. Johanneson, "A Fast Algorithm for Computing Distance Spectrum of Convolutional Codes," *IEEE Transactions on Information Theory*, vol. IT-35, p. 1146-1159, November 1989.
- [2] Zehavi, Ephraim, and Jack K. Wolf, "On the Performance Evaluation of Trellis Codes," *IEEE Transactions on Information Theory*, vol. IT-33, no. 2, March 1987.
- [3] Oliver Collins, "The Subtleties and Intricacies of Building a Constraint Length 15 Convolutional Decoder," *IEEE Transactions on Communications*, vol. 40, no. 12 pp. 1810-1819, December 1992.
- [4] O. Collins, R.J. McEliec, S. Dolinar and F. Poltera, "The VLSI Decomposition of the DeBruijn graph," *J. Assoc. Comput. Machinery*, vol. 39, no. 4, pp. 931-948, Oct. 1992.

#### APPENDIX

Pins for Processor Chips, Information furnished by MRC

<u>Pin Name</u>	<u>Count</u>	<u>Description</u>
-----------------	--------------	--------------------

M	256	These pins transfer metric and traceback information to and from the individual radix-16 metric processors in bit-serial format. They are point to point communication lines. The wiring pattern is determined solely by partitioning of metric evaluation processors into individual integrated circuits. UNM has currently done no work to determine optimal chip placement or node processor partitioning. Max data rate approx 40 MHz.
---	-----	--

The envisioned format of the data flowing on these bit serial lines is as follows: 1) The bit serial busses each contain 16 time slices per metric evaluation. Currently 10 to 12 of these time slices are envisioned as being used

to transfer metric information. The remaining will be used for traceback information (propagating in the opposite direction, of course.)

If a detailed analysis shows that fewer bits are actually required for the metric data, then the total number of slots may be reduced accordingly.

S 10 Winning metric select pins. Used to determine which was the smallest metric in a given metric evaluation iteration. Used to renormalize on going metric calculations. Max Data rate approx 40 MHz.

Six of these pins are wired as a balanced binary tree (two 2-bit busses in and one 2-bit bus out). Two are global output pins and two are global input pins. The actual chips may be connected arbitrarily, as long as a binary tree is formed. The circuit implements a binary divide and conquer compare scheme.

Clock 1 System Clock, global

Q 2 Data Out, Tri-State, Global, One pin indicates output data is a one, the other that the output data is a zero. If both are active, the output is ambiguous. Max rate approx 10 Mhz.

ResetN 1 Global Reset

D 3 Input Data. Max data rate 20 MHz. Global Connections.

NoLock 1 No Data Sync. Status Output. Only the pin associated with the chip corresponding to the root of the binary tree built by the S pins contains useful information.

Config 2 Metric processor initialization, serially shifted from chip to chip during system initialization.