FINAL REPORT TO

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Electrical Engineering Division
Goddard Space Flight Center
on the project entitled

HIGH-SPEED SOFT-DECISION DECODING OF TWO REED-MULLER CODES

Grant Number: NAG 5-2613

Principal Investigator: Shu Lin, Professor
e-mail: slin@spectra.eng.hawaii.edu
Co-Principal Investigator: Gregory T. Uehara, Assistant Professor
e-mail: uehara@spectra.eng.hawaii.edu

Department of Electrical Engineering
University of Hawaii at Manoa
2540 Dole street
483 Holmes Hall
Honolulu, HI 96822

February 20, 1996

INTRODUCTION

In his research, we have proposed the (64, 40, 8) subcode of the third-order Reed-Muller (RM) code to NASA for high-speed satellite communications. This RM subcode can be used either alone or as an inner code of a concatenated coding system with the NASA standard (255, 233, 33) Reed-Solomon (RS) code as the outer code to achieve high performance (or low bit-error rate) with reduced decoding complexity. It can also be used as a component code in a multilevel bandwidth efficient coded modulation system to achieve reliable bandwidth efficient data transmission.

This report will summarize the key progress we have made toward achieving our eventual goal of implementing a decoder system based upon this code.

In the first phase of study, we investigated the complexities of various sectionalized trellis diagrams for the proposed (64, 40, 8) RM subcode. We found a specific 8-trellis diagram for this code which requires the least decoding complexity with a high possibility of achieving a decoding speed of 600 M bits per second (Mbps). The combination of a large number of states and a high data rate will be made possible due to the utilization of a high degree of parallelism throughout the architecture. This trellis diagram will be presented and briefly described. In the second phase of study which was carried out through the past year, we investigated circuit architectures to determine the feasibility of VLSI implementation of a high-speed Viterbi decoder based on this 8-section trellis diagram. We began to examine specific design and implementation approaches to implement a fully custom integrated circuit (IC) which will be a key building block for a decoder system implementation. The key results will be presented in this report.

This report will be divided into three primary sections. First, we will briefly describe the system block diagram in which the proposed decoder is assumed to be operating and present some of the key architectural approaches being used to implement the system at high speed. Second, we will describe details of the 8-trellis diagram we found to best meet the trade-offs between chip and overall system complexity. The chosen approach implements the trellis for the (64, 40, 8) RM subcode with 32 independent sub-trellises. And third, we will describe results of our feasibility study on the implementation of such an IC chip in CMOS technology to implement one of these sub-trellises.

1. Background and Implementation Considerations

We will begin this section with a brief discussion of the system block diagram in which the proposed decoder is assumed to be operating. Next, we will examine advantages of the proposed architectures for implementation of the Viterbi decoder along with design considerations which result. Following this we will present the architecture we have chosen for implementation of the decoder system.

System Block Diagram

A simplified block diagram of a receiver in which the proposed decoder may be used is shown in Fig. 1. The signal enters the receiver via an antenna and is first amplified by a low noise amplifier (LNA) before begin passed to the 2-PSK demodulator. We assume the functions of carrier and timing acquisition and gain control are properly performed in the demodulator. The output of the demodulator is sampled at the correct phase at the symbol rate of 960 MHz. The output of the sampler is converted to the digital domain by the 3-bit analog-to-digital converter (ADC) for decoding by the Viterbi Decoder block which follows. Our discussion will focus exclusively on the implementation of the Viterbi Decoder.

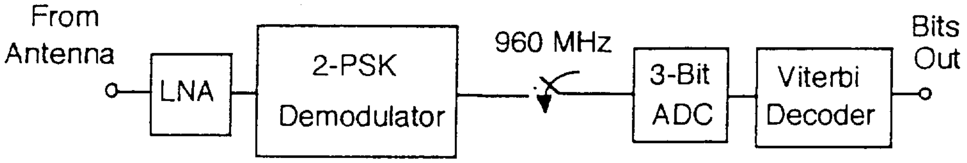


Figure 1 Block diagram of a high speed satellite receiver employing 2-PSK signalling and a Viterbi Decoder.

Summary of System Level Architectural Considerations

In our earlier report [1], we describe in detail the different ways in which parallelism can be utilized to decode the (64, 40) RM code. We will briefly present a summary of that description in this section.

There are many diverse issues at different levels of the design requiring consideration for implementation of the (64, 40) RM code at a rate of 600 Mbits/sec. Fig. 2 illustrates the different layers of hierarchy associated with the proposed implementation. First, there are N parallel decoders with each operating on a different independent block of 64 symbols. Given a decoder which can decode a 64-symbol block at a certain rate, using N decoders and having them each operate on a different block of 64 symbols allows a throughput N times greater.

Second, each decoder is implemented with K parallel isomorphic subtrellises. As described in [6], the trellis for an RM code can be decomposed into parallel isomorphic subtrellises that are connected at only the inputs and outputs as shown conceptually in Fig. 2 with K parallel subtrellises. This has a tremendous advantage for IC implementation because it minimizes the amount of routing required within the trellis which would otherwise be unrealizable at high speed for applications requiring large numbers of states. This is the key which makes an implementation using CMOS IC's at such a high rate and complexity possible.

And third, there are a number of parameters associated with the implementation of each of the K subtrellises. The first is the number of sections in the subtrellis denoted as L. Next, is the number of states at the end of each section i ($i = 1, 2, \ldots, L$) denoted as $|S_i|$ which will generally not be the same. Finally, there is the radix of each section denoted as R_i for radix R in section i. As the number of sections L decreases, the complexity of each section and the number of parallel branches per section increases. These trade-offs are discussed in detail in [1].

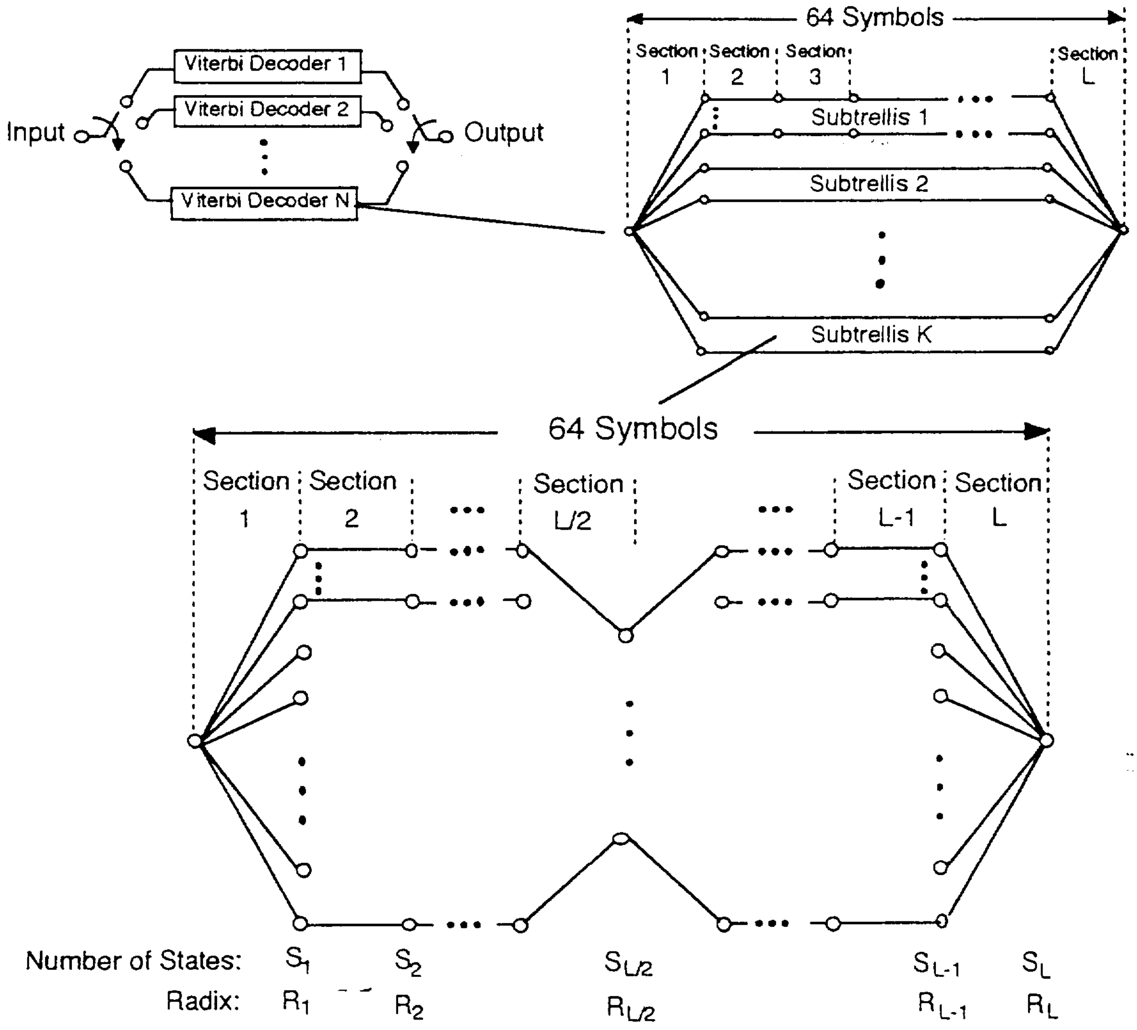


Figure 2 Levels of hierarchy in the proposed Viterbi decoder implementation. (a) Parallel Viterbi decoders operating on different blocks of data. (b) Implementation with K parallel isomorphic subtrellises. (c) Subtrellis implementation.

2. Architecture Chosen for Implementation

In this section, we will present the architecture we chose (over two other candidates) to investigate for implementation of the decoder and present some of the approaches we have developed for implementation of this architecture.

In Fig. 3 is the 8-section trellis which we are investigating for implementation of the decoder. It illustrates the form of two of the parallel isomorphic subtrellis for this chosen architecture. Atop the trellis is the number of subtrellises required to implement the decoder. The numbers inside the subtrellises indicate the number of states in that particular section of the trellis. Below the trellis is the radix at each stage of the trellis.

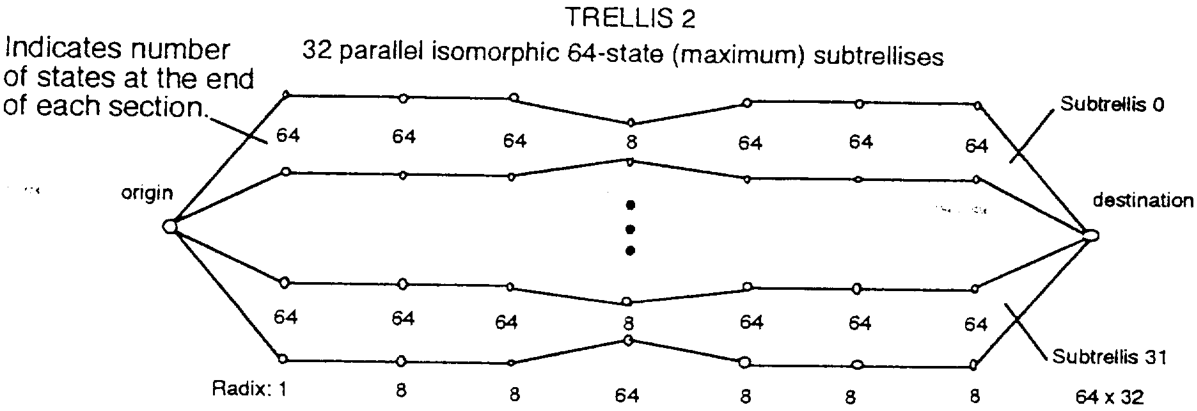

TRELLIS 2
32 parallel isomorphic 64-state (maximum) subtrellises
Indicates number of states at the end of each section.
64
64
64
8
64
64
64
Subtrellis 0
origin
destination
64
64
64
8
64
64
64
Subtrellis 31
Radix: 1
8
8
64
8
8
8
64 x 32

Figure 3 The 8-section architecture we are investigating for implementation of the 600 Mb/sec Viterbi decoder for the (64,40,8) RM subcode.

Implementing one of the 32 subtrellises on a single chip at such a high speed will not be trivial and will require full custom circuit design. From a yield/cost standpoint, the die size of an IC should be kept on the order of 10 mm on each side (100 mm^2). This and other factors were considered in choosing Trellis 2 for further investigation.

The detailed structure of one of the subtrellises for Trellis 2 is shown in Fig. 4. As can be seen in the figure, the 8-way ACS is a critical building block for implementation of this subtrellis. As described in [1], the approach we are examining is based upon a customized 8-way ACS block which is used with comparators to implement the radix-64 section in Section 4 of the subtrellis.

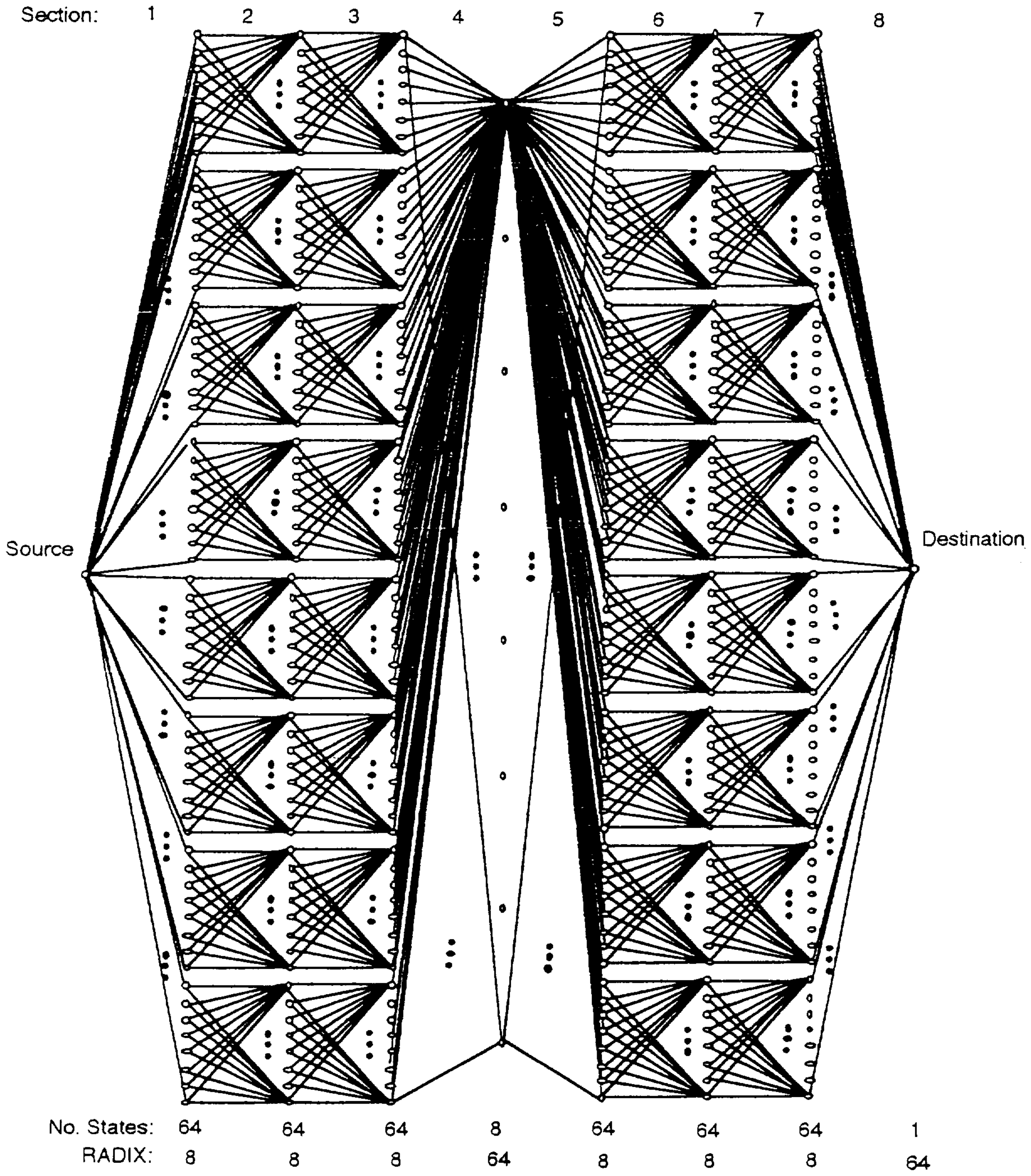


Figure 4 Detailed subtrellis structure for Trellis 2.

3. Chip Plan and Key Results from the Feasibility Study

The key to the implementation of a (64, 40) RM decoder will be the successful implementation of an IC implementing the subtrellis described in the previous section. In this section, we will present some of the key results from the feasibility study of the past year in which we examined the issues associated with such an implementation.

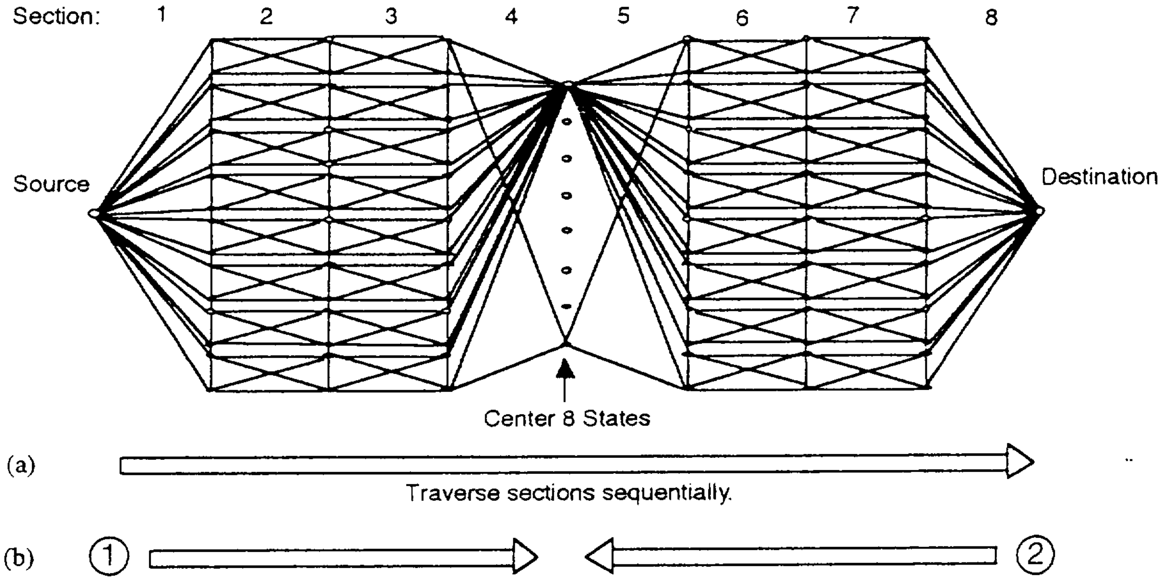
The key objectives of the subtrellis IC implementation are to:

1. Maximize the efficiency as measured by maximizing the utilization of the hardware (in other words, attempt to minimize the time the majority of the hardware is not being used).
2. Use a chip plan which minimizes the area used for routing (routing area is simply an overhead which should be minimized).
3. In whatever the available technology, attempt to approach the speed of 600 Mbits/sec with the minimum number of parallel decoders (in other words, attempt to attain the highest possible speed in a given technology subject to the constraints in the next objective).
4. Consider reliability and robustness issues. In particular, use the lowest speed system clock possible which allows high speed operation in order to reduce the number of issues which can limit the performance (which in this case would be clock skew between chips or race conditions both within and between the different ICs.
5. Consider the board design and the numbers of inputs and outputs to each chip to facilitate implementation of the final decoder system.
6. Keep the size of the IC on the order of 10 mm per side to facilitate its implementation and yield for testing.
7. Utilize the most aggressive IC technology available to our design team at the time of the design.

In this section, we will results for 3 key aspects of the design including the sequence to be used to decode the 8 sections of the subtrellis, the overall chip plan, and some of the details associated with the design of the 8-way ACS.

Decoding Sequence

Due to the inherent nature of block codes, they can be decoded either sequentially or out of order as shown in Fig. 6. The arrow in Fig. 6a indicates how a trellis is typically decoded sequentially, starting with Section 1 and on through to Section 8. In Fig. 6b is another approach where, first, Sections 1 through 4 are decoded sequentially and path information corresponding to the most likely paths into the center 8 states which are the destination states in Section 4 are stored. Next, Sections 5 through 8 are decoded starting from Section 8 and moving back through to Section 5. The path metrics corresponding to the most likely paths into the 8 destination states at the end of Section 5 (moving right to left) are then added to those which were found into those states from the first 4 sections. The two paths (entering the center 8 states) with the largest path metric sum comprise the most likely path through the trellis.



① Resolve first 4 sections;
Store largest path metrics into the center 8 states.

② Resolve second set of 4 sections (starting from Section 8 through Section 5).

Sum the largest path metrics into the center 8 states from both sides.

Find largest path metric through the subtrellis.

Figure 5 Two possible decode paths for the subtrellis. (a) Traverse all sections sequentially. (b) Traverse in two sections.

The approach we have adopted is a third approach which we call the *modified concurrent bi-directional execution sequence*. This approach exploits the use of pipelining in the ACS implementation and the mirror symmetry of the subtrellis about the center axis (the 8 center states) and results in potential advantages in terms of both speed and structural regularity. Sections are decoding starting from Section 1 and then Section 8, Section 2 and then Section 7, and on down the line until the center is reached and the entire path is resolved as in approach (b) illustrated in Fig. 5.

Sequence for Decoding time →

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

Figure 6 Sequence for decoding using the modified concurrent bi-directional execution sequence.

Chip Plan

An outline of the overall chip plan illustrating the major blocks is shown in Fig. 7a. The *Clock Generation and Control* block will generate the necessary clock phases to clock the chip. Input data will enter the *Branch Metric Unit* (*BMU*) which will generate the branch metrics for the *Add-Compare-Select Unit* (*ACSU*). The outputs of the ACS Unit include the winning path metrics and the winning branch labels. These are input to the *Decoder* which determines the most likely path through the subtrellis for the 64-symbol block.

Pipelining is used extensively within the BMU, ACSU, and the Decoder. Preliminary circuit design suggests that to achieve a 600 Mbits/sec decode rate in a 0.6 μm CMOS process, 2 decoders operating in an interleaved manner will be required. As a result, each will be required to operate at a 300 Mbits/sec rate. The symbols will enter the chip at a 300 Mbits/sec x (64/40) = 480 Msymbols/sec rate. The incoming symbols will be separated into groups of 8 3-bit symbols and enter the chip at a 480 M/8 = 60 MHz rate. We currently plan to have the input clock to the chip clock at this 60 MHz rate.

A tentative design for the BMU employs pipelining and takes 3 cycles of the input clock to generate the branch metrics for one section of the trellis. This is indicated in the timing diagram in Fig. 7b with a 3 clock cycle delay from the instant that input data is latched to the time at which branch metrics for a section are output. Each of the stages are shown with the movement of data corresponding to Section 1 indicated with a darkened timing bubble. The outputs of the BMU are input to the ACSU which after 3 cycles of the clock generates outputs for the first section which are passed to the decoder. With each subsequent clock, the ACSU outputs path metrics and branch labels in the order presented in Fig. 6. After the outputs for Section 5 are generated, the decoder then has all the information it needs to determine the most likely path through the subtrellis. Extensive simulations were performed examining different circuit and architectural approaches for implementation of the ACSU. Since this block is potentially the bottleneck to high speed performance and will consume the majority of chip area, much time was spent investigating various permutations of pipelining and parallelism and algorithmic approaches until settling on one which we believe to best meet the various design considerations.

The final decode function is not a trivial one due to the size and amount of data output from the ACSU. During its operation, the ACSU finds the most likely paths from the start of the subtrellis to each of the 8 states at the end of Section 4 and the end of the trellis traversing back through Sections 8-5 to the same location. The decoder must then combine these most likely paths and determine the most likely path from the start to the end of the subtrellis. It must do so while keeping track of the winning branch labels of the partial paths in order to output this information along with the winning path metric to the off-chip post processing which follows. The off-chip processing then determines the path most likely among the most likely from each of the 32 subtrellis. The functions which comprise the decode function are also pipelined although this is not indicated explicitly in the figure.

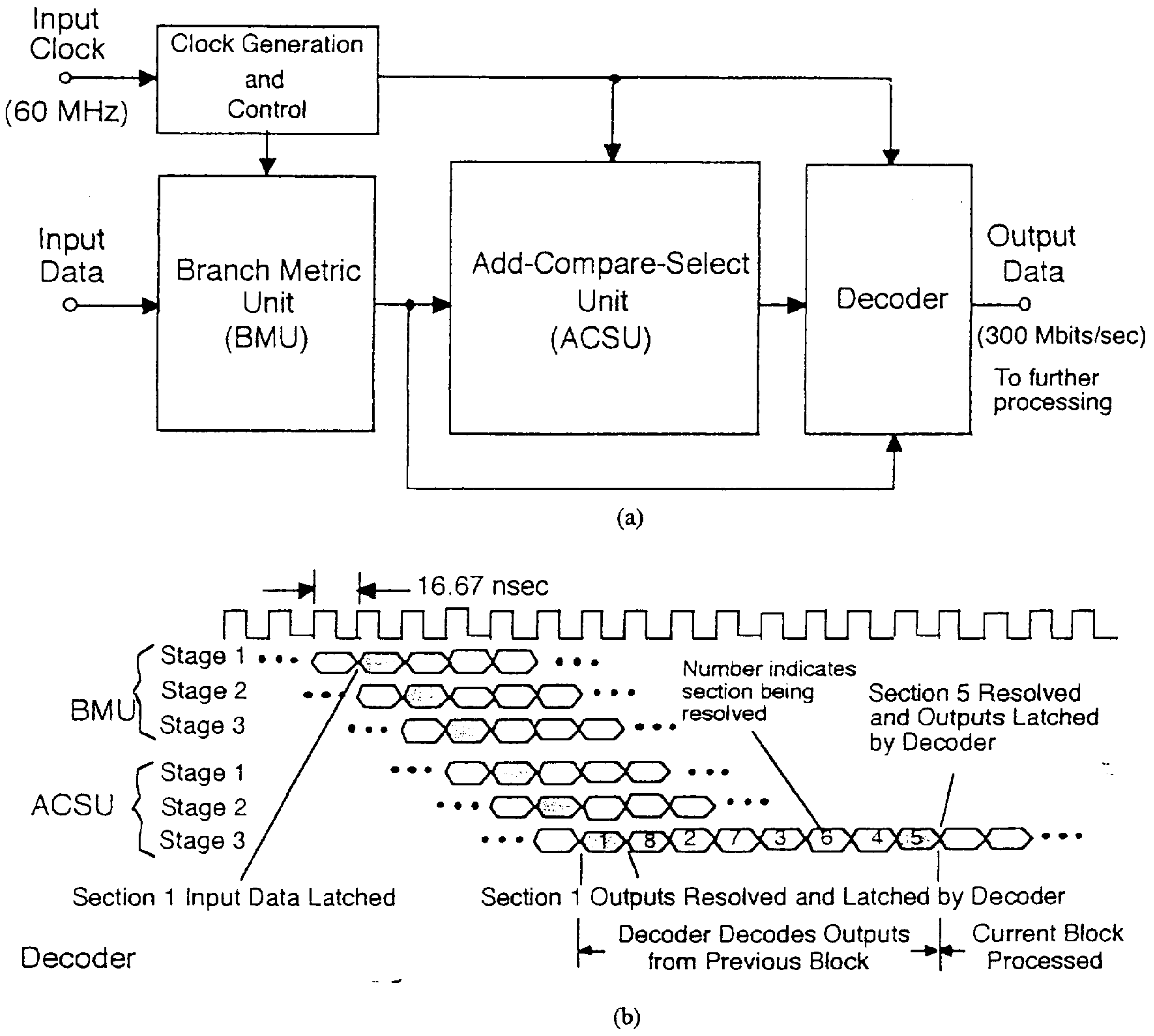

Input Clock
(60 MHz)
Clock Generation and Control
Input Data
Branch Metric Unit (BMU)
Add-Compare-Select Unit (ACSU)
Decoder
Output Data
(300 Mbits/sec)
To further processing
(a)
16.67 nsec
BMU
Stage 1
Stage 2
Stage 3
ACSU
Stage 1
Stage 2
Stage 3
Number indicates section being resolved
Section 5 Resolved and Outputs Latched by Decoder
1
8
2
7
3
6
4
5
Section 1 Input Data Latched
Section 1 Outputs Resolved and Latched by Decoder
Decoder
Decoder Decodes Outputs from Previous Block
Current Block Processed
(b)

Figure 7 (a) Block diagram of the IC being developed to implement a subtrellis. (b) Basic high level timing diagram.

4. Summary and Future Work

Research Summary

In the first phase of study, we investigated the complexities of various sectionalized trellis diagrams for the proposed (64, 40, 8) RM subcode. We found a specific 8-trellis diagram for this code which requires the least decoding complexity with a high possibility of achieving a decoding speed of 600 M bits per second (Mbps). In the second phase of study which was carried out through the past year, we investigated circuit architectures to determine the feasibility of VLSI implementation of a high-speed Viterbi decoder based on this 8-section trellis diagram. We began to examine specific design and implementation approaches to implement a fully custom integrated circuit (IC) which will be a key building block for a decoder system implementation. This examination was performed in order to study the feasibility of implementing such a decoder at such high speed using primarily CMOS technology.

The results of our feasibility study indicate that it is feasible to implement such an IC meeting the objectives outlined at the beginning of Section 3 in a somewhat optimum manner assuming the use of a 0.65 μm CMOS process which is currently available to us. In this technology, current data suggests that the 600 Mbits/sec speed should be attainable using 2 parallel decoders ($N = 2$ in the Section 1 discussion).

The key results upon which we base this conclusion include:

1. Development of the optimum sequence with which sections of the trellis should be decoded in order to meet the objectives outlined above.
2. Development of an overall chip plan.
3. Circuit design and layout of the ACS unit. This includes scheduling of the data inside the ACS block which has many considerations and a large amount of data in transit
4. Scheduling of the inputs and outputs to and from the chip and between the major blocks of the chip.
5. Die size in this technology may exceed the 10 mm per side target by up to 20% per side. This target will be easily met in a state-of-the-art technology (0.25 μm CMOS) which in principle should allow the 600 Mbits/sec speed to be implemented with $N = 1$.
6. Preliminary gate level circuit design of over 80% of the major blocks.

Much work still remains in the circuit design, layout, and simulation of the chip.

Future Work

We will be continuing the development of a decoder system, focusing our current efforts on continuing the development of a full custom CMOS IC to implement a subtrellis which will be the key building block for the system.

The long term goal of this project is to demonstrate performance and implementation advantages of Reed-Muller codes for very high speed, bandwidth efficient communication.

REFERENCES

[1] H. T. Moorthy, S. Lin, and G. T. Uehara, "On the trellis structure of a (64,40,8) subcode of the (64,42,8) third-order Reed-Muller code," NASA Report, NAG 5-931, Report No. 95-001, March 1, 1995.

[2] 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.

[3] P. Black and T. Meng, "A 140 Mb/s 32-state radix-4 Viterbi decoder," *ISSCC 1995 Digest of Technical Papers,* San Francisco, CA, Feb. 1992.

[4] P. Black, "Algorithms and architectures for high speed Viterbi decoding," *Ph.D. Thesis*, Stanford University, May 1993.

[5] G. Fettweis and H. Meyr, "High-speed parallel Viterbi decoding algorithm and VLSI-architecture," *IEEE Communications Magazine*, Vol. 29, May 1991.

[6] 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. E76-A, No. 9, pp. 1411-1421, September 1993.