Low-Power Asynchronous Viterbi Decoder for Wireless Applications

Mohamed Kawokgy C. André T. Salama

Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto, Toronto, Ontario M5S 3G4, Canada
Tel: (416)-978-8658 Fax:(416)978-4516

kawokgy@vrg.utoronto.ca salama@vrg.utoronto.ca

ABSTRACT

This paper describes the implementation of an asynchronous 64-state, 1/2-rate Viterbi decoder using an original architecture and design methodology. The decoder is intended for wireless communications applications, where bit rates over 100 Mb/s and minimum power consumption are sought. The choice of an asynchronous design was predicated by the power and speed advantages of such a methodology. Asynchronous designs are inherently data driven and are active only when doing useful work, enabling considerable savings in power and operating at the average speed of all components. The decoder, implemented in a 0.18 µm CMOS technology, occupies an area of 2 mm^2 and operates above 200 Mb/s while consuming 85 mW: a 55% power reduction when compared to state of the art synchronous design implemented in a 0.25 µm technology.

Categories and Subject Descriptors

B.1.1 [Control Structures and Microprogramming]: Control Design Styles; B.2.1 [Arithmetic and Logic Structures]: Design Styles – *Pipeline*; B.4.3 [Input/Output and Data Communications]: Interconnections (Subsystems) – *Asynchronous/Synchronous Operations*; B.5.1 [Register-Transfer-Level Implementation]: Design – *Arithmetic and Logic Units – Control Design – Data-Path Design – Styles*; B.5.2 [Register-Transfer-Level Implementation]: Design Aids – *Hardware Description Languages*; B.7.1 [Integrated Circuits]: Types and Design Styles – *Algorithms Implemented in Hardware – VLSI*; C.3 [Special-Purpose and Application Based Systems]: Signal Processing Systems.

General Terms

Algorithms, Performance, Design.

Keywords

Viterbi Algorithm, Wireless, Digital signal Processing, Synchronous, Asynchronous, Four-Phase, Speed-Independent, Handshaking Protocol, Bundled-Data, Low-Power, Register Transfer Level, VHDL.

1. INTRODUCTION

Viterbi decoders are widely used in digital transmission and recording systems and are expected to be used in next generation wireless applications. Such portable, battery operated systems, require low-power consumption as well as high processing speeds, over 100 Mb/s, to allow multimedia transmission. The objective of this work is to investigate the design of a CMOS Viterbi decoder for such applications.

Two design styles are available to the implementation of Digital Signal Processors (DSPs): synchronous and asynchronous. Conventional synchronous DSP designs are controlled by a global clock, running throughout the entire system. Asynchronous designs, on the other hand, are locally rather than globally synchronized and use handshaking signals between their components in order to perform the necessary synchronization and sequencing of events. There are many advantages to be gained from migrating to asynchronous designs and giving up the use of a global clock, with all its inherent overhead. In terms of power consumption, synchronous systems involve higher switching activity than asynchronous ones. High switching activity translates into a large amount of wasted power. On the other hand, in data driven asynchronous systems, idle parts consume negligible power and switching activity is associated only with useful work being done: a valuable feature for battery operated systems. Asynchronous systems produce less electromagnetic emissions than their synchronous counterparts which generate spurious signals at the operating clock frequency and all its harmonics. These signals interfere with cellular phones, television and navigation systems. Finally, the speed of synchronous systems is determined by the worst case path delay which determines the maximum clock frequency. However, asynchronous systems run on the average delay of all their components. But asynchronous designs suffer from two major drawbacks. First, there is an overhead associated with the asynchronous handshaking units, in terms of silicon area, speed and power. Second, there is a lack of CAD tools for use in such designs. Both of these drawbacks are addressed in the design of the asynchronous Viterbi decoder described in this work.

ISLPED?04, August 9-11, 2004, Newport Beach, California, USA.
Copyright 2004 ACM 1-58113-929-2/04/0008?$5.00.

2. ASYNCHRONOUS VITERBI DECODER DESIGN

2.1 Viterbi Algorithm

Viterbi decoders [1] are used to decode data encoded using convolutional encoders and transmitted over noisy channels. A message encoded using a convolutional encoder follows what is called a trellis diagram which shows the different states of the encoder as well as the path taken to encode an arbitrary message. Viterbi's algorithm tries to reconstruct this correct path based on the received stream, despite errors in the received stream. This is done by reconstructing the trellis diagram and allocating a weight to each branch and node (i.e. state) of the reconstructed trellis, at each time slot. These weights define the likely branches and nodes used by the encoder. This is referred to as the Maximum Likelihood Probability (MLP) [1]. By tracing back through the reconstructed trellis, the decoder can detect and correct errors in the received stream.

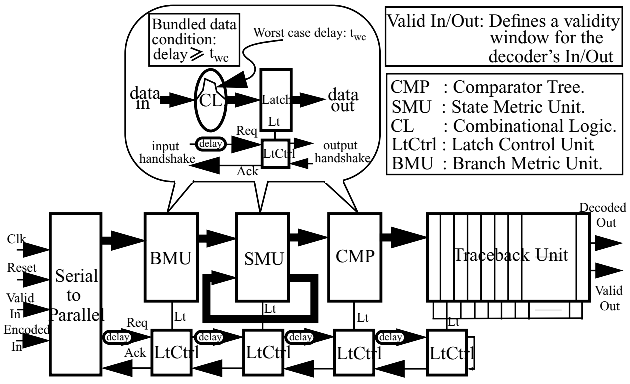
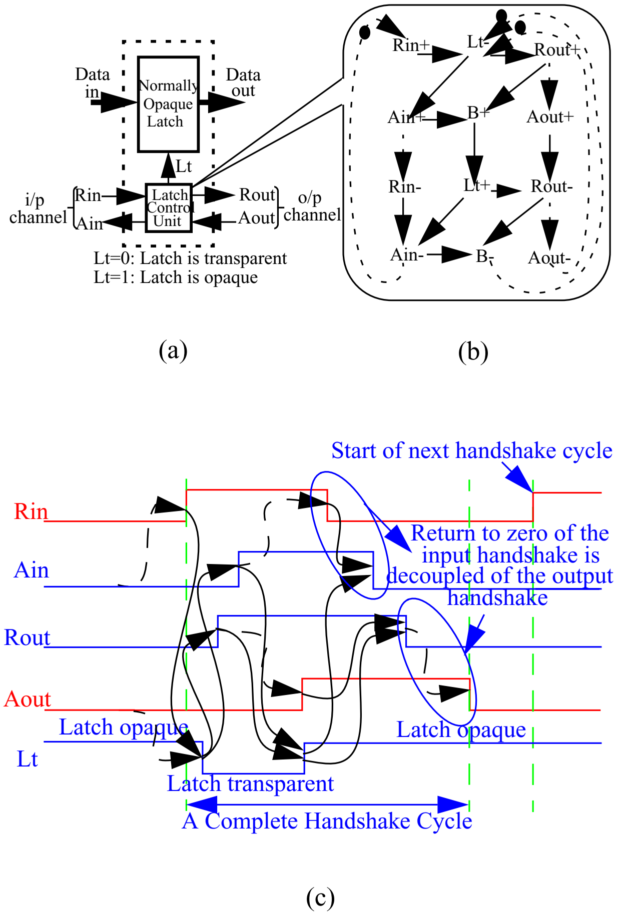


Figure 1. Proposed asynchronous Viterbi architecture.

2.2 Asynchronous Viterbi Architecture

The proposed circuit is a hard-decision, 64-state, 1/2- rate Viterbi decoder. The generator polynomials (G) used are the industrial standards (171_8,133_8). The basic building blocks of the decoder are shown in Figure 1. The asynchronous decoder is assumed to be integrated in a larger synchronous system using a serial to parallel interface which, continuously, receives 2-bit symbols, parallelizes them, then passes them to the Branch Metric Unit (BMU) along with a request signal. The BMU computes the weight of each branch of the trellis. Each branch weight is, recursively, added to the corresponding state weight, in the State Metric Unit (SMU), to generate the new state weight. The comparator tree (CMP) selects the state with minimum accumulated weight. This represents the state most likely reached by the trellis diagram at a given time stage. Once the trellis diagram is reconstructed, tracing back through the trellis is performed in the Traceback Unit (TBU) and errors, in the received stream, are detected and corrected.

The asynchronous control section of the decoder was designed, using Petrify tool [2], as a set of speed independent latch control units using 4-Phase handshaking protocol and single rail bundled-data encoding scheme. For the sake of speed improvement, these latch controllers were designed as fully decoupled [3,4]. Whereas, for the sake of power reduction all over the system, normally opaque latches were used [4]. Figure 2 (a), (b) and (c) show the used latch control circuit, its corresponding Signal Transition Graph (STG) [4] and timing diagram, respectively. In order to satisfy the bundled data condition associated with the design of single rail asynchronous circuits, delay elements are inserted in the handshaking channel of the latch controllers to mimic the worst case combinational logic path delay.



(a) (b)

(c)

Figure 2. (a) The used latch control unit. (b) Corresponding signal transition graph (STG). (c) Corresponding timing diagram.

3. ASYNCHRONOUS VITERBI IMPLEMENTATION

This section discusses the performance of the proposed asynchronous Viterbi decoder architecture described in Section 2.2. A comparison to a synchronously designed version as well as previously reported state of the art designs [5,6] is also presented.

3.1 Decoder Implementation

The decoder design was carried out using VHDL Hardware Description Language and mapped to a conventional 0.18 μm CMOS standard cell library. A simple asynchronous design methodology, that resembles synchronous design styles, was developed in this work and allowed the use of commercially available simulation tools as well as standard cell libraries rather than requiring the development of custom libraries. This was achieved by separating the logic from the asynchronous control in a pipelined fashion. Figure 3 summarizes the design flow used to synthesize the asynchronous decoder as well as its synchronous counterpart, respectively. The design methodology used was the same for both the synchronous and the asynchronous designs except for the use of the Petrify tool to synthesize the asynchronous control units from the Signal Transition Graph (STG) [4].

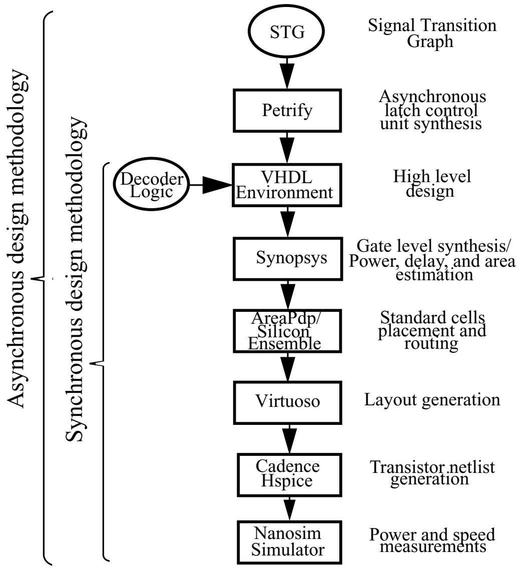

Asynchronous design methodology
Synchronous design methodology
STG
Signal Transition Graph
Petrify
Asynchronous latch control unit synthesis
Decoder Logic
VHDL Environment
High level design
Synopsys
Gate level synthesis/ Power, delay, and area estimation
AreaPdp/ Silicon Ensemble
Standard cells placement and routing
Virtuoso
Layout generation
Cadence Hspice
Transistor netlist generation
Nanosim Simulator
Power and speed measurements

Figure 3. Asynchronous/Synchronous design Flow.

3.2 Simulation Results

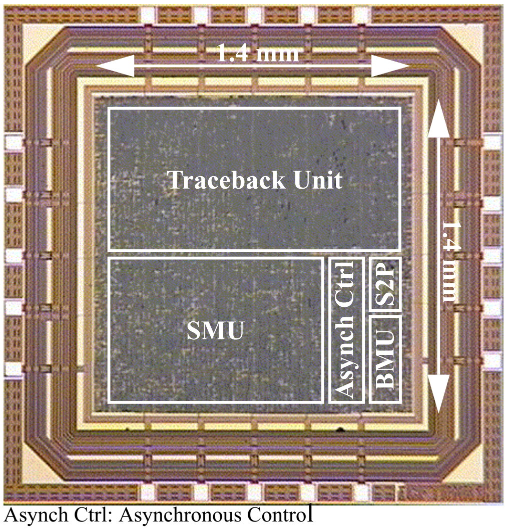
Postlayout simulation for the complete chip was performed using the Nanosim simulator. Simulation results showed that the maximum input stream decoded by the asynchronous design was 230 Mb/s at an average power consumption of 90 mW. The chip core occupied a total silicon area of 1.4 x 1.4 mm^2. Table 1 summarizes the postlayout simulation results of the designed decoder.

Table 1. Asynchronous Viterbi decoder postlayout simulation results.

Results	Postlayout Simulation
Design	Asynch. Design This Work
Technology	CMOS 0.18 μm
No of States	64
Rate (r)	1/2
Core Area	2 mm^2
Maximum Speed	230 Mb/s
Average Power @Max. Speed	90 mW @230 Mb/s
Average Power @100 Mb/s	39 mW

3.3 Experimental Results

The micrograph of the implemented asynchronous decoder chip is shown in Figure 4. The chip core has an area of 1.4 x 1.4 mm^2. The main sections of the decoder are shown on the micrograph. The decoder has a 400K transistor count.



Asynch Ctrl: Asynchronous Control
S2P: Serial to Parallel

Figure 4. Micrograph of the asynchronous Viterbi decoder.

Experimentally, the asynchronous decoder was found to operate at a maximum speed of 213 Mb/s while dissipating 85 mW. Figure 5 shows the experimental Bit Error Rate (BER) performance of the asynchronous decoder as compared to postlayout simulation presented in Section 3.2. The former was measured using Agilent 81200 Data Generator/Analyzer test equipment while the latter was calculated using Matlab. In both cases, the BER performance of the decoder was obtained by applying a long encoded data stream and comparing the output decoded message to the original one. In general, good agreement with simulations was observed.

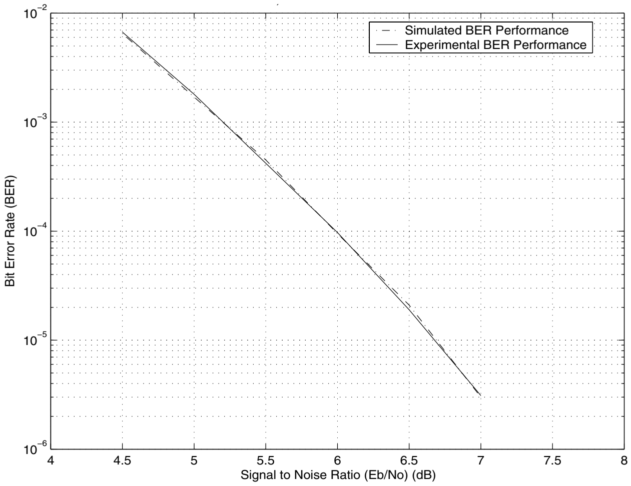


Figure 5. Experimental and simulation Bit Error Rate (BER) performance comparison.

Table 2 compares the experimental results of the implemented asynchronous Viterbi to state of the art decoders [5,6]. In terms of power, the present design considerably outperforms previous designs with a slight improvement in the processing speed. At 200 Mb/s, the asynchronous decoder exhibits a 55% reduction in power consumption as compared to You's synchronous design [5] implemented in 0.25 μm CMOS technology. At 90 Mb/s, the present design outperforms a previously reported asynchronous design [4,6] by a factor of 15 in terms of power. In both cases, while some of the improvement in performance of the present design can be attributed to the smaller design feature size, the majority of this improvement is due to the asynchronous architecture and design methodology used. This was confirmed by comparing postlayout simulation results obtained on the asynchronous decoder, presented in this work, to that of a synchronously designed counterpart implemented using the same 0.18 μm technology. The asynchronous design featured a 45% reduction in power as compared to the synchronous one.

Table 2. Comparison between the asynchronous design and state of the art Viterbi decoders.

Results	Experimental Results		
Design	Synch. Design You [5]	Async. Design Riocreux [4,6]	Async. Design This Work
Technology	CMOS 0.25 μm	CMOS 0.35 μm	CMOS 0.18 μm
Core Area	2.16 mm^2	N/A	2 mm^2
No of States	64	64	64
Rate (r)	1/2	1/2	1/2
Maximum Speed	200 Mb/s	90 Mb/s	213 Mb/s
Average Power @Max. Speed	183 mW @200 Mb/s	1333 mW @90 Mb/s	85 mW @213 Mb/s
Average Power @100 Mb/s	90 mW	N/A	40 mW

4. CONCLUSIONS

This paper described the implementation of a fully asynchronous, 64-state, 1/2-rate Viterbi decoder, suitable for wireless applications. The asynchronous Viterbi decoder uses a pipelined architecture with asynchronous control units and normally opaque latches. The latch control units were designed as speed independent asynchronous circuits using 4-phase bundled-data protocol. The complete design was carried out using VHDL Hardware Description Language and implemented in a 0.18 μm CMOS using a standard cell library. A simple asynchronous design methodology was developed in this work and allowed the use of commercially available simulation tools as well as standard cell libraries rather than requiring the development of custom libraries geared to asynchronous designs.

The circuit was characterized in terms of power, speed and area. Experimental results showed that the implemented asynchronous Viterbi chip was operational up to 213 Mb/s while consuming 85 mW: a 55% power reduction over previously reported synchronous designs.

5. Acknowledgments

This work was supported by NSERC, Micronet, Gennum, Nortel Networks and PMC-Sierra. The implementation was carried out through the Canadian Microelectronics Corporation.

6. REFERENCES

[1] B. P. Lathi, "Modern Digital and Analog Communication Systems," *Oxford University Press*, New York, 1998.

[2] http://www.lsi.upc.es/~jordic/petrify/petrify.html

[3] S. B. Furber, and P. Day, "Four-Phase Micropipeline Latch Control Circuits," *International Symposium on Advanced Research in Asynchronous Circuits and Systems,* Proceedings, pp. 15-24, 2001.

[4] J. Sparso, and S. Furber (EDs.), "Principles of Asynchronous Circuit Design," *Kluwer Academic Publishers,* Boston, 2001.

[5] Y. You, J. Wang, F. Lai, and Y. Ye, "VLSI Design and Implementation of High-Speed Viterbi Decoder," *IEEE International Conference on Communications,* vol. 1, pp. 64-68, 2002.

[6] P. A. Riocreux, L. E. M. Brackenbury, M. Cumpstey, and S. B. Furber, "A Low-Power Self-Timed Viterbi Decoder," *International Symposium on Advanced Research in Asynchronous Circuits and Systems,* Proceedings, pp. 15 - 24, 2001.