HIGH-SPEED ADD-COMPARE-SELECT UNITS USING LOCALLY SELF-RESETTING CMOS

Gunok Jung, Jun Jin Kong, Gerald E. Sobelman and Keshab K. Parhi

Department of Electrical and Computer Engineering
University of Minnesota
Minneapolis, MN 55455, USA
Phone: (612) 625-8041, Fax: (612) 625-4583
e-mail: sobelman@ece.umn.edu

ABSTRACT

This paper presents a new self-resetting CMOS design for an Add-Compare-Select (ACS) unit, which is a key building block in a Viterbi decoder. Static CMOS and two-phase domino CMOS designs have also been implemented for comparison purposes. The simulation results show that, with the SRCMOS technique, the ACS units operate at a data rate of 568 Mbps in a 0.25 micron CMOS technology, as compared to 357 Mbps and 485 Mbps for static and domino CMOS implementations, respectively.

1. INTRODUCTION

The Viterbi algorithm [1] has become a very important decoding scheme for a wide range of applications in digital communications and data storage. Many implementations use a state-parallel architecture to obtain high throughput [2], [3], [4], [5]. Even so, achieving very high performance is still a challenge due to the computational requirements of the add-compare-select (ACS) unit [6].

In this paper, we show that the performance of Viterbi decoders can be enhanced by implementing the critical ACS function using an advanced circuit-level design technique. In particular, we develop an implementation using a locally self-resetting form of self-resetting CMOS (SRCMOS). The improvement in performance is demonstrated by comparing this design with standard designs based on static CMOS and two-phase domino CMOS.

The architecture of the Viterbi decoder we consider is shown in Fig. 1. The decoder consists of three functional blocks, the branch metric calculator (BMC), the ACS/register unit and the path memory (PM) block. In our design, we use a constraint-length, K, of 5 (so that there are a total of 2^{K-1} = 16 states), and a code rate, R, of 1/2.

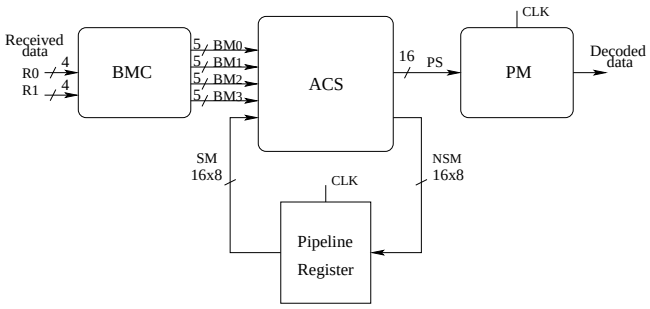

Fig. 1. Functional block diagram for a Viterbi decoder.

2. ACS IMPLEMENTATION

The overall speed of a Viterbi decoder is largely determined by the ACS computation time. In a state-parallel structure, the number of ACS units is the same as the number of states. Accordingly, 16 ACS units are used in our design. Each ACS unit consists of two saturated adder blocks, a comparator block and a selector block, as shown in Fig. 2.

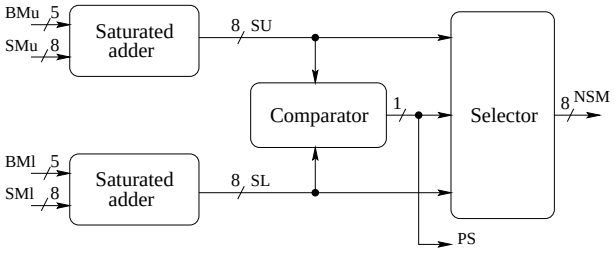

Fig. 2. Block diagram for an Add-Compare-Select (ACS) unit.

Each saturated adder has the structure shown in Fig. 3, where the adder portion has been implemented using the high-performance Kogge-Stone architecture [7]. Note that the input operands, A and B, and the sum, S, are all un-

signed numbers. The saturation feature is implemented as follows: If the addition creates a final carry of one, then the active-low c_8 signal will be low. This forces all of the sum output bits, s_7 - s_0, to be high and all of the complementary sum output bits, sn_7 - sn_0, to be low. (Both the true and complement forms of the sum bits are generated because the domino or SRCMOS circuits in succeeding logic stages are constructed using dual-rail implementations.)

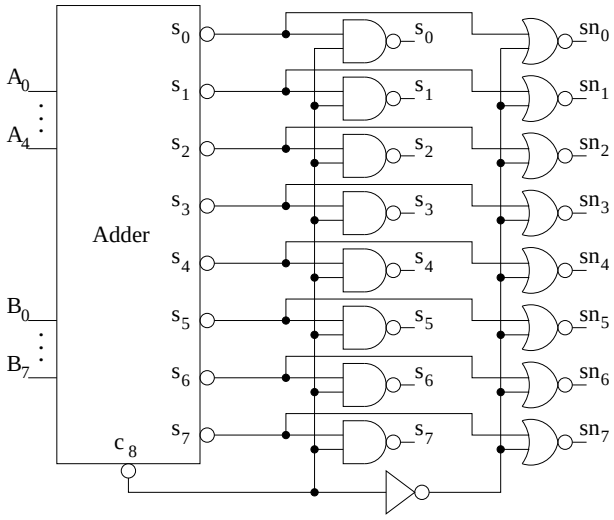

Fig. 3. Circuit structure for a saturated adder.

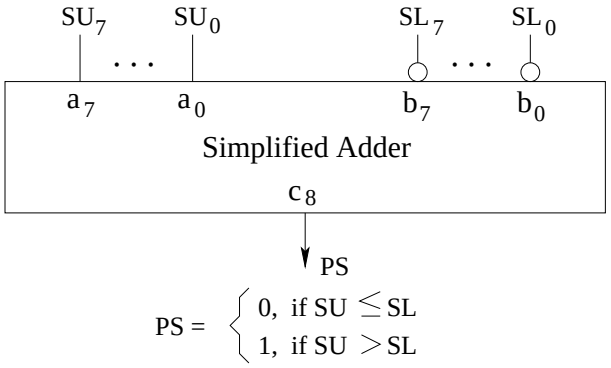

Fig. 4. Block diagram for the comparator.

The comparator is implemented using a simplified adder structure where only the final carry output is needed, as shown in Fig. 4. This unit operates as follows: If SU[7:0] is less than or equal to SL[7:0], then a final carry will not be generated so that the path select (PS) output becomes zero. Otherwise, a final carry will be generated so that PS becomes one.

The selector block is implemented using CMOS transmission gates, and operates as follows: If PS is zero, then the SU[7:0] signals are sent to the NSM output lines. Otherwise, if PS is one, then the SL[7:0] signals are sent to the NSM outputs.

The pipeline register block consists of a set of positive-edge-triggered D-type flip-flops (D-FFs). The total number of D-FFs required is equal to the number of states multiplied by the number of state-metric bits, i.e. $16 \times 8 = 128$.

The D-FF circuits used in the pipeline register and the PM are shown in Fig. 5. Part (a) shows an active-low asynchronous reset D-FF, and part (b) shows an active-low asynchronous preset D-FF. The simulated clock-to-Q delays for these two circuits are found to be 0.2 ns and 0.3 ns, respectively.

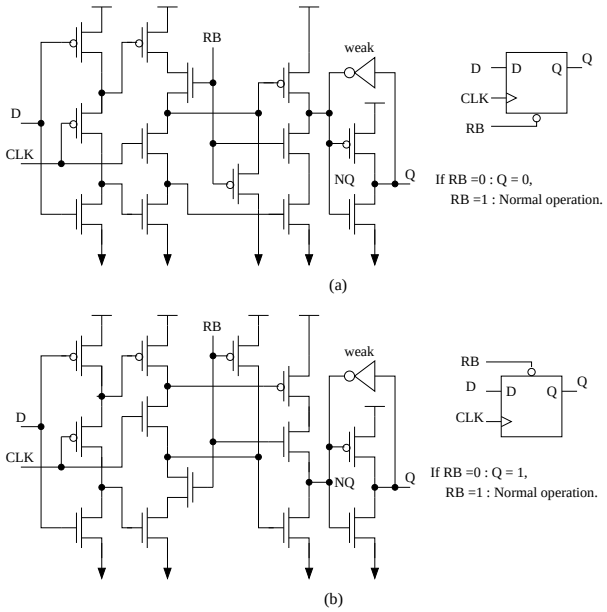

Fig. 5. D-FF circuit designs. (a) Active-low asynchronous reset D-FF (b) Active-low asynchronous set D-FF.

3. SRCMOS DESIGN DETAILS

In SRCMOS, signals are represented as short-duration pulses, and the circuits are reset (i.e., precharged) immediately after they evaluate. Thus, there is no need for a separate precharge phase. In addition, the evaluate or "foot" devices can be eliminated, since the input pulses return to zero before the reset operation occurs.

Various forms of SRCMOS have been proposed in recent years. In globally self-resetting CMOS [8], the reset signal for each stage is generated by a separate timing chain which provides a parallel worst-case delay path. On the other hand, in locally self-resetting CMOS [9], the reset signal for each stage is generated by a mechanism internal to that stage. Most previous implementations of SRCMOS have required great care in device sizing in order to ensure that the reset pulses will occur at the correct times. This is

especially difficult to achieve over a wide range of process, voltage and temperature variations. However, the form of SRCMOS used in this paper is much more robust [10].

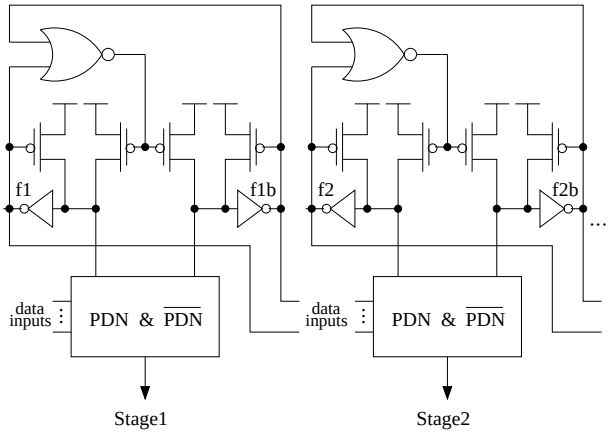

Fig. 6. Circuit structure for a robust form of locally self-resetting CMOS.

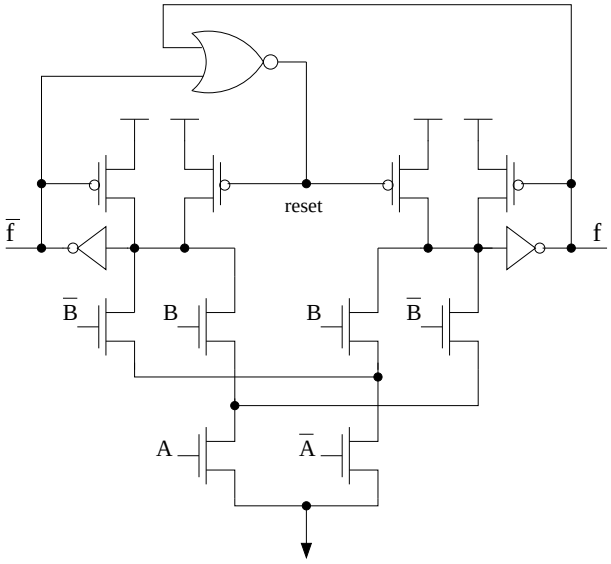

Fig. 7. SRCMOS implementation of the first-level propagate signal.

The main idea is shown in Fig. 6, where the locally-generated reset signal is obtained from the dual-rail output lines of a stage of logic. After the stage has completed its evaluation, one of the dual-rail outputs is guaranteed to go high. This will cause the NOR gate output to go low, which turns on the pair of PMOS precharge transistors. (Since the data inputs are short-duration pulses, they would have already returned to zero by this time, so there will not be any fighting between the precharge transistors and the merged pull-down network.) This, in turn, will force both outputs to become low, so that the NOR gate output goes high, turning off the precharge transistors. At this point, the stage is ready to begin another evaluation cycle. Note that the sequencing of all of these operations is enforced by the logical structure of the circuit, so the timing will automatically adjust to accommodate process, voltage and temperature variations.

As an example, the SRCMOS circuit for computing the first-level propagate signals in the adder is shown in Fig. 7. In this and other SRCMOS and domino circuits, we have used skewed static gates at the outputs in order to improve the evaluation times. In addition, we have used secondary PMOS precharge transistors (not shown) connected to high-capacitance internal nodes in the pull-down network to eliminate charge sharing and weak PMOS keepers for increased robustness against noise.

4. SIMULATION RESULTS AND PERFORMANCE COMPARISONS

Our circuit designs were evaluated using the 0.25 micron CMOS process of TSMC that is available from MOSIS. For purposes of comparison, we designed three different circuit-level implementations of the ACS unit. In addition to the locally self-resetting CMOS design, we have constructed standard implementations using static CMOS and two-phase domino CMOS.

A timing diagram for the Viterbi decoder is shown in Fig. 8. The delay through the BMC does not affect the clock cycle time since it is not part of the critical path, which is through the ACS unit and the pipeline register.

The simulation results for the three different implementations of the ACS unit are given in Table 1. The ACS propagation delay for the SRCMOS design is 42% less than for the static CMOS implementation, although the energy consumption is much larger. However, it must be emphasized that the ACS unit is only a part of the entire Viterbi decoder. Therefore, for the whole system, the relative increase in power due to using SRCMOS in the ACS unit would be a smaller amount.

The throughput for each of the three designs are given in Fig. 9. These values are obtained as 1/(ACS propagation delay + D-FF delay).

5. CONCLUSIONS

We have presented design and simulation results for a high-speed implementation of the ACS unit of a state-parallel Viterbi decoder. A robust form of locally-resetting SRCMOS is used to obtain significantly higher throughput than can be obtained using standard static and domino CMOS designs, at the expense of a higher power dissipation. This design style should therefore be of interest for those applications that have very high decoding speed requirements.

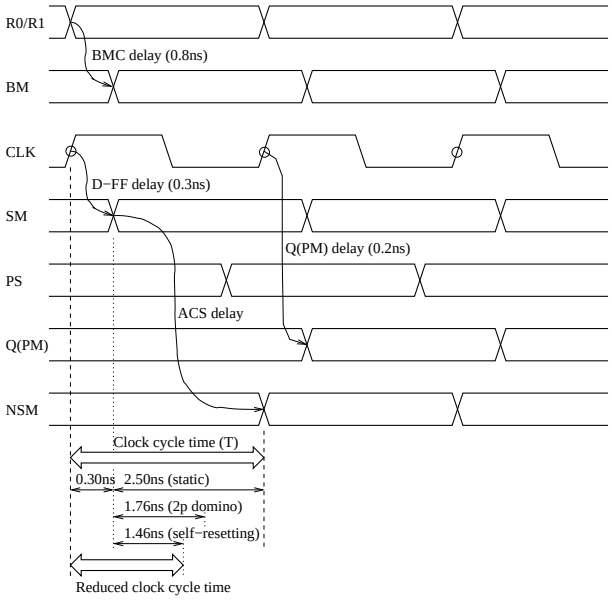

Fig. 8. Timing diagram for the Viterbi decoder.

Table1: Comparison of the ACS circuit implementations.

Circuit style	Propagation delay time	Average power	Maximum power	Energy consumption	Number of transistors
Static CMOS	2.50 ns	12.6 mW	44.9 mW	31.5 pJ	952
Two−phase domino	1.76 ns	49.8 mW	260.7 mW	87.6 pJ	1,768
Self−resetting CMOS	1.46 ns	130.6 mW	225.2 mW	190.7 pJ	2,243

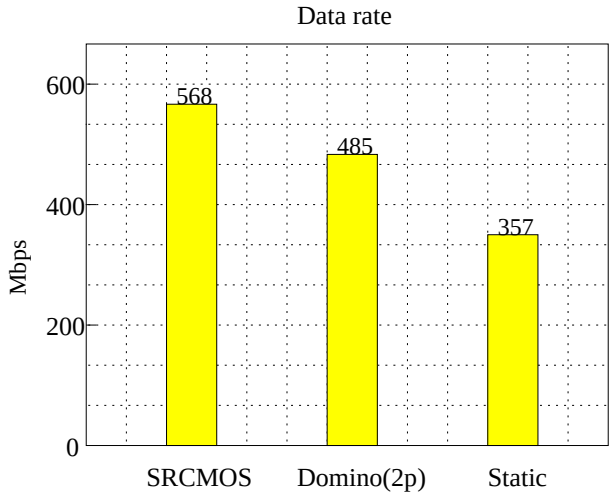

Fig. 9. Decoder throughput for 3 ACS circuit styles.

6. ACKNOWLEDGMENTS

This research was support in part by the National Science Foundation under grant number CCR-9988262.

7. REFERENCES

[1] G. D. Forney, Jr., "The Viterbi Algorithm," *Proc. IEEE,* Vol. 61, pp. 268-278, 1973.

[2] Sung-Han Choi and Jun-Jin Kong, "State Parallel Viterbi Decoder Soft IP and Its Applications," *IEEE TENCON,* Vol. 1, pp. 355-358, 2001.

[3] K. He and G. Cauwenberghs, "Integrated 64-State Parallel Analog Viterbi Decoder," *IEEE International Symposium on Circuits and Systems,* Vol. 4, pp. 761-764, 2000.

[4] K. He and G. Cauwenberghs, "An Area-Efficient Analog VLSI Architecture for State-Parallel Viterbi Decoding" *IEEE International Symposium on Circuits and Systems,* Vol. 2, pp. 432-435, 1999.

[5] G. Fettweis and H. Meyr, "High-Speed Parallel Viterbi Decoding: Algorithm and VLSI-Architecture," *IEEE Communications Magazine,* Vol. 29, No. 5, pp. 46-55, May, 1991.

[6] Jun Jin Kong et al, "A Fast Serial Viterbi Decoder ASIC for CDMA Cellular Base Station, *IEEE TENCON,* pp. 333-337, 1996.

[7] P. M. Kogge and H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," *IEEE Trans. on Computers,* Vol. C-22, No. 8, pp. 786-793, 1973.

[8] W. Hwang et al, "Implementation of a Self-Resetting CMOS 64-Bit Parallel Adder with Enhanced Testability," *IEEE Journal of Solid-State Circuits,* Vol. 34, No. 8, pp. 1108-1117, August 1999.

[9] A. E. Dooply and K. Y. Yun, "Optimal Clocking and Enhanced Testability for High-Performance Self-Resetting Domino Pipelines," *Proceedings, 20th Conf. on Advanced Research in VLSI,* pp. 200-214, 1999.

[10] Gunok Jung, V. Sundarajan and Gerald E. Sobelman, "A Robust Self-Resetting CMOS 32-bit Parallel Adder," submitted for publication.