

HIGH-SPEED HYBRID THRESHOLD-BOOLEAN LOGIC COUNTERS AND COMPRESSORS

Marius Padure, Sorin Cotofana, Stamatis Vassiliadis

Computer Engineering Laboratory,
Delft University of Technology,
Mekelweg 4, 2628CD Delft, The Netherlands
{maris,sorin,stamatis}@dutepp0.et.tudelft.nl

ABSTRACT

In this paper we propose high-speed hybrid Threshold-Boolean logic counters and compressors employed in parallel multiplication and multiple operand addition. First, we present a depth-2 hybrid implementation scheme for arbitrary symmetric Boolean functions, based on differential Threshold logic gates as circuit style. Subsequently, we apply the previous general scheme to parallel p/q counters and $p/2$ compressors. Finally, we present hybrid logic designs of a $7/3$ counter and a $7/2$ compressor. The simulation results, suggest that the hybrid $7/3$ counter and $7/2$ compressor designed in $.25\mu\text{m}$ CMOS, achieve between 53% and 61% higher speed when compared with traditional Boolean logic and Threshold logic counterparts, at expense of between 67% and 74% more transistors.

1. INTRODUCTION AND PREVIOUS WORK

Multiplication and multiple operand addition schemes for VLSI implementations have received much attention and have been solved using several approaches [1, 2, 7, 8] for partial product reduction matrix schemes. The design of efficient counters and compressors is a key issue in the design of high-performance parallel multipliers since the reduction schemes employ traditionally p/q counters or $p/2$ compressors [1, 8].

High order counters and compressors have been implemented usually by using logarithmic-depth networks of full adders (i.e., $3/2$ counters) [1, 7, 8]. Currently, other possibilities exist in VLSI for the constant-depth implementation of such counters using custom circuits (e.g., Threshold logic) in CMOS technology [3, 6, 9].

Threshold logic (TL) originally emerged in the early 60's as a generalized theory of logic gates, which includes conventional Boolean logic (BL) as its subset [5]. A Threshold Logic Gate (TLG) is defined as an n -input processing element such that its output performs the following Boolean function¹: $F(X) = \text{sgn} \{ \sum_{i=0}^{n-1} \omega_i \cdot x_i - T \}$, where $X = \{x_0, x_1, \dots, x_{n-1}\}$, $\Omega = \{\omega_0, \omega_1, \dots, \omega_{n-1}\}$ and T are the set of Boolean input variables, the set of fixed signed integer weights associated with data inputs, and the fixed signed integer threshold, respectively [5].

TL is fundamentally more powerful than Boolean logic since the TL gate can perform more complex and wider functions than the usual Boolean CMOS gates (e.g., Nand, Nor, Invert) can. Several recent theoretical investigations [9, 10] have indicated that computer arithmetic building blocks (e.g., adders and multipliers)

can be implemented in TL with smaller number of logic gates and fewer logic stages when compared with traditional Boolean logic counterparts.

In this paper we propose novel high-speed hybrid Threshold-Boolean logic (HTBL) counters and compressors. High-speed design of counters and compressors is addressed in both electrical and logical directions. Electrical improvements are achieved by resorting to a fast differential latch-based TLG [6] design style. Logical improvements are achieved via the introduction of a novel depth-2 hybrid Threshold-Boolean logic implementation for arbitrary symmetric Boolean functions, in general, and for parallel p/q counters and $p/2$ compressors, in particular. The main idea behind the hybrid method is the use of a first TL stage, computing in parallel all necessary Threshold functions over the primary inputs, connected by *dual-rail* signals to a second traditional Boolean logic stage. Due to the dual-rail connection between the Threshold and Boolean logic stages, the second stage achieves *high-speed* operation and *regular routing*. Simulation results, in $.25\mu\text{m}$ CMOS, suggest that the parallel HTBL $7/3$ counters and $7/2$ compressors achieve between 53% and 61% higher speed when compared with both traditional Boolean logic and Threshold logic counterparts, at expense of between 67% and 74% more transistors.

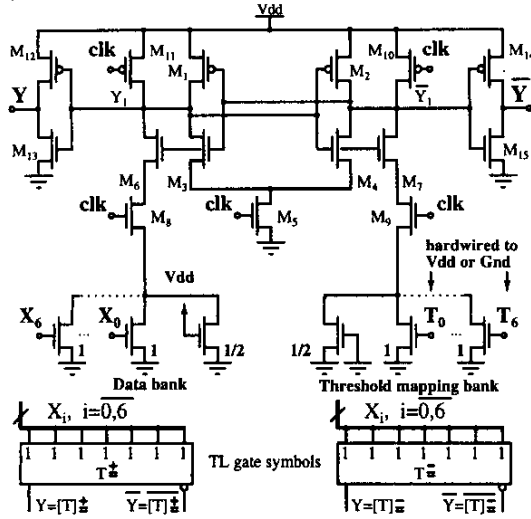
The paper is organized as follows: Section 2 briefly reviews the differential TL gate employed by the hybrid scheme; Section 3 introduces the hybrid depth-2 implementation scheme for arbitrary symmetric Boolean functions and its utilization for the implementation of parallel p/q counters and $p/2$ compressors. Additionally, the hybrid logic designs of a $7/3$ counter and $7/2$ compressor are described; Section 4 presents simulation results and a comparison in terms of delay and estimated area of counters/compressors implemented in traditional Boolean, traditional Threshold, and HTBL. Section 5 presents some concluding remarks.

2. DIFFERENTIAL TL GATE DESCRIPTION

The differential TL gate schematic [6] and its basic symbols utilized further in the paper are depicted in Figure 1. The gate is in principle a clocked differential cascode voltage switch (DCVS) circuit, operated with a single phase clock. It comprises a fast latched comparator and two parallel-connected sets of unit nMOS transistors, referenced herein as input data bank and threshold mapping bank (as its transistors have usually the gates hardwired to ground or power supply). The gate presented in Figure 1 has $\omega_i = 1$ for every input while the same holds true for every threshold mapping input (T_i). The total conductances of the transis-

¹All the operators are algebraic.

Figure 1: 7-input differential latch-based TL gate schematic and its symbols



tor banks are compared each other by the latched comparator and therefore the output Y is logic one if the current generated by the data bank is greater than the current generated by the threshold mapping bank and logic zero otherwise. Please note that, by design, the data bank is prevented to have similar conductance with the threshold mapping bank, when the threshold is reached, since an nMOS transistor with weight 0.5 is always on. This prevents the latch comparator entering in a metastable state.

As described in [6], the TL gate from Figure 1 has several potential advantages over other CMOS TL gates, e.g., [3], which makes it suitable for high-speed hybrid counter and compressor designs, as follows:

- the gate achieves high-speed of operation since the latched comparator is very fast;
- the gate is complementary since it provides both Y and \bar{Y} simultaneously;

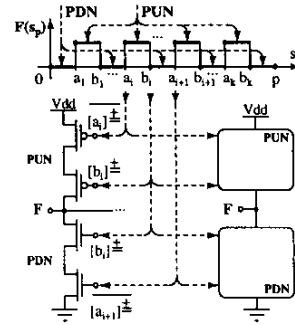
During the next section we use the following notations: p is the total number of primary inputs, generally, for arbitrary symmetric functions or, in particular, for counters and compressors; $s_p = \sum_{i=0}^{p-1} x_i$, the total number of true inputs; $[T]_+^+$, $[T]_-^+$, $T = \bar{0}, p$, (as TL symbols from Figure 1 suggest for $p = 7$) denote $sgn\{s_p - T\}$ and $sgn\{T - s_p\}$ respectively; $'+$ ' and $'\cdot'$ operators are the algebraic sum and product while $'\vee'$ and $'\wedge'$ denote a logical 'OR' and 'AND' operation, respectively.

3. HYBRID COUNTERS AND COMPRESSORS DESIGN

In this Section we present first a depth-2 hybrid implementation scheme for arbitrary symmetric Boolean functions, based on differential Threshold logic gates as circuit style. Subsequently, we apply the previous general scheme to parallel p/q counters and $p/2$ compressors. Finally, we present hybrid logic designs of a $7/3$ counter and a $7/2$ compressor.

Hybrid design of arbitrary symmetric functions: An p -variable Boolean function, $F(x_0, \dots, x_{p-1})$, is called *symmetric* iff (if and only if) it is invariant to any permutation of its inputs or, equivalently, iff F depends only on the sum of its inputs, s_p . Consequently, instead of using a truth table with 2^p entries, a symmetric function can be described by stating the ranges of s_p for

Figure 2: Arbitrary symmetric function implementation using hybrid Threshold-Boolean approach



which the function is true [4, 5]. Any symmetric function can be specified either as a reunion of intervals for which $F = 1$, called symmetric function *on-set*, or as a reunion of intervals for which $F = 0$, called symmetric function *off-set*. We denote by *interval description*, both the on-set and off-set of the symmetric function. The interval description of an arbitrary symmetric function is depicted in the upper part of Figure 2, where by the dot sign, \bullet , we mean a closed interval.

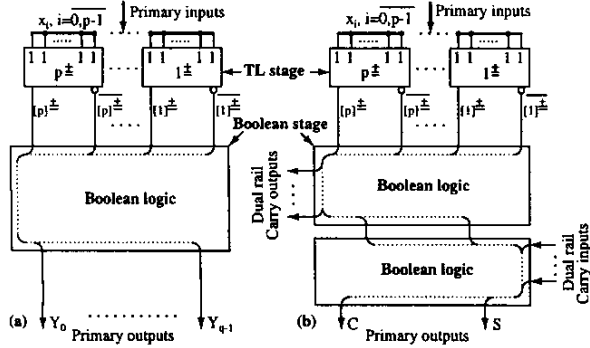
More formally, given a symmetric Boolean function with k intervals in which $F = 1$, as in Figure 2, its on-set can be expressed as $F^1(s_p) = \bigcup_{i=1}^k [a_i, b_i]$, while its off-set as the reunion of complementary intervals $F^0(s_p) = [0, p] - F^1(s_p) = \bigcup_{i=0}^k [b_i, a_{i+1}]$, where $b_0 = 0$ and $a_{k+1} = p$.

Given any arbitrary symmetric Boolean function of p variables, having the interval description as in Figure 2, we propose a general hybrid Threshold-Boolean implementation approach in two stages as follows:

Stage 1: the first stage comprises a total of $2k$ fan-in p differential TL gates, as those presented in Figure 1, computing in parallel $[a_i]_+^+$, $[b_i]_-^+$, $i = \bar{1}, k$. Therefore, k TL gates are responsible with the computation of "rising transitions", $[a_i]_+^+$, while the other k TL gates with the computation of "falling transitions", $[b_i]_-^+$. Note that TL gates computing $[0]_+^+$ and $[p]_-^+$ are not necessary for the computation of off-set intervals $[0, a_1]$ and $[b_k, p]$ as there are no "transitions" in $s_p = 0$ and $s_p = p$, with respect to the interval description depicted in Figure 2. However, there are symmetric Boolean functions having such kind of transitions (e.g., parity). Moreover, since the TL gate employed is differential, both true and complement forms of the output signals are available.

Stage 2: the second Boolean stage receives the dual-rail outputs of the first stage and generates single-rail primary outputs. It comprises a pull-up network (PUN) and a pull-down network (PDN). The PUN is responsible with the implementation of the symmetric function on-set while PDN implements its off-set. Each on-set/off-set interval (and only one) is implemented by 2 serial connected MOS transistors, as suggested in Figure 2, for the intervals $s_p \in [a_i, b_i]$ and $s_p \in [b_i, a_{i+1}]$. Please note, the outer intervals $s_p \in [0, a_1]$ and $s_p \in [b_k, p]$ are implemented each with only one transistor in PDN as the gates $[0]_+^+$ and $[p]_-^+$ are not necessary. Since generally, the interval descriptions are basically reunions of intervals and since each interval is implemented as in Figure 2, PUN and PDN can be extended easily by connecting in parallel such sets of no more than two series transistors. Therefore high-speed operation is achieved.

Figure 3: General hybrid Threshold-Boolean topology: a) p/q counters and b) $p|2$ compressors; p odd



With respect to Figure 2, the PUN and PDN work as follows: assuming that $s_p \in [a_i, b_i]$, it follows that $[a_i]^+ = 0$ and $[b_i]^+ = 0$, therefore both pMOS transistors presented in Figure 2 are on and $F(s_p) = 1$, as it should. Similarly, assuming that $s_p \in [b_i, a_{i+1}]$, it follows that $[b_i]^+ = 1$, $[a_{i+1}]^+ = 1$, both PDN nMOS transistors are on and consequently $F(s_p) = 0$, as it should. Please note that, by construction, no electrical conflict between PUN and PDN may exist as *only one* set of serial connected transistors is active at a time even in PUN or in PDN.

The hybrid scheme presented for arbitrary symmetric functions can be applied to every p/q parallel counter [1, 2, 8] and $p|2$ parallel compressor [2, 7, 8] as depicted in Figure 3.a and in Figure 3.b respectively, for p having odd value². Given their counting behavior, the first stage have in common p fan-in p TL gates computing in parallel all p -input possible symmetric function “transitions” from their interval description, $[i]^+_{\pm}$, $i = \overline{1, p}$. In contrast with p/q parallel counters, $p|2$ parallel compressors are restricted to compute some of their carry outputs *earlier* and depending *only* from the primary inputs, therefore the Boolean stage will have actually at least one additional Boolean layer, as suggested in Figure 3.b.

7/3 and 7|2 counters and compressors: A 7/3 counter can be described by a interval description depicted in Figure 4.a and equivalently by the following on-set [9] expressed as Boolean equations:

$$Y_2 = [4]^+_{\pm}, Y_1 = [2]^+_{\pm} \wedge [4]^+_{\pm} \vee [6]^+_{\pm}$$

$$Y_0 = [1]^+_{\pm} \wedge [2]^+_{\pm} \vee [3]^+_{\pm} \wedge [4]^+_{\pm} \vee [5]^+_{\pm} \wedge [6]^+_{\pm} \vee [7]^+_{\pm}. \quad (1)$$

Figure 4.a and, equivalently, Equations (1) suggest that, for example, the Y_0 output has to be logic one in 4 intervals ($s_7 = 1, 3, 5, 7$) and logic zero in 4 intervals ($s_7 = 0, 2, 4, 6$). Consequently the Boolean gate implementing the Y_0 output, has 4 sets of pMOS and 4 sets of nMOS transistors in PUN and PDN respectively. More formally, the PUN can be synthesized assuming Y_0 equation, with negated terms and \wedge/\vee operators meaning serial/parallel connection of pMOS transistors, as in the synthesis theory of static CMOS gates. Similarly, PDN can be synthesized taken into account the off-set descriptions. Figure 5 presents the Boolean stage implementation of the 7/3 HTBL counter. Please

²The scheme we propose can be also utilized unrestricted for p even. When assuming p having even value, e.g., 8/4 counter, the interval description has no “transition” in $s_p = p$, therefore the TL gate implementing $[p]^+_{\pm}$ is not necessary.

Figure 4: Interval descriptions: a) 7/3 counter, b) 7|2 compressor, \times sign indicates possible redundant representation; c) interval implementation example

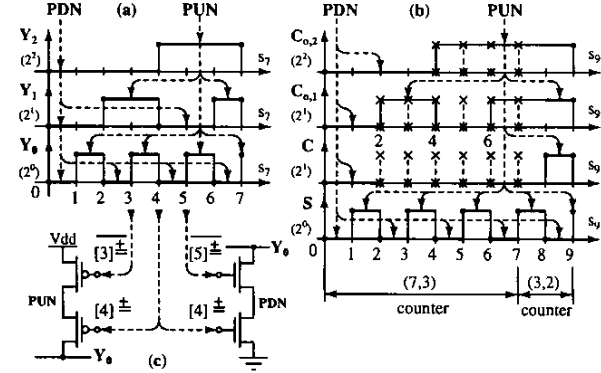
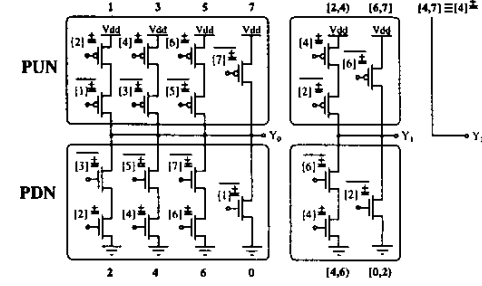


Figure 5: 7/3 HTBL counter—Boolean stage design



note that since the Boolean stage comprises only parallel sets of maximum 2 series MOS transistors, the routing is *regular*. Using the same way of reasoning, higher order hybrid counters e.g., 15/4, 31/5, can be designed.

We extend the hybrid logic design to compressors by presenting the equations of a hybrid 7|2 compressor [2, 9]. Logically speaking a 7|2 compressor can be viewed as a 7/3 counter followed by a 3/2 counter. Additionally, a compressor connected at column i has 2 carry inputs, $C_{in,1}$, $C_{in,2}$, and 2 carry outputs, $C_{o,1}$, $C_{o,2}$, connected to compressors from columns $i+1$ and $i+2$ and thus having weights 2^1 and 2^2 . For all possible data inputs, a 7|2 compressor must satisfy the following arithmetic equation [2]:

$$s_7 + (C_{in,1} + C_{in,2}) = s_9 = 2^2 C_{o,2} + 2^1 C_{o,1} + 2^1 C + 2^0 S. \quad (2)$$

with $C_{o,1}$, $C_{o,2}$ symmetric functions depending only on s_7 (the sum of the primary inputs) and C , S symmetric functions depending on s_9 (the sum of both primary and carry inputs). Please note that Equation (2) implies a degree of redundancy (e.g., $s_9 = 4$ can be written as $C_{o,2} = 1$, $C_{o,1} = C = S = 0$, or $C_{o,2} = S = 0$, $C_{o,1} = C = 1$). One possible set of logic equations, describing the on-set, (the interval description depicted graphically in Figure 4.b) is the following:

$$\begin{aligned} C_{o,2,i} &= [4]^+_{\pm,i}, C_{o,1,i} = [2]^+_{\pm,i} \wedge [4]^+_{\pm,i} \vee [6]^+_{\pm,i}, \\ Y_{0,i} &= (1, 3, 5, 7)_i, \overline{Y_{0,i}} = (0, 2, 4, 6)_i, \\ C_i &= Y_{0,i} \wedge [1]^+_{\pm,i-1,i-2} \vee [2]^+_{\pm,i-1,i-2}, \\ S_i &= Y_{0,i} \wedge (0, 2)_{i-1,i-2} \vee \overline{Y_{0,i}} \wedge (1)_{i-1,i-2}. \end{aligned} \quad (3)$$

Table 1: 7/3 counters data (.25 μ m CMOS, typical corner)

Type	Delay [ps]	Norm. delay	Delay [Δ_{FA}]	# of Tran.
BL	726	1.00	2.42	136
TL	460	0.63	1.53	155
HTBL	345	0.47	1.15	237

Table 2: 7|2 compressors data (.25 μ m CMOS, typical corner)

Type	Delay [ps]	Norm. delay	Delay [Δ_{FA}]	# of Tran.
BL	1250	1.00	4.16	170
TL	638	0.51	2.12	415
HTBL	489	0.39	1.63	285

where the subscript distinguish between signals from current and from previous two bit positions and $Y_{0,i}$ and $\bar{Y}_{0,i}$ are dual-rail intermediate parity signals, computed in a similar manner as for 7/3 counter. Please note that $C_{o,1}$ and $C_{o,2}$ have similar equations with the Y_2 and Y_1 outputs of the 7/3 counter and depend only on the primary inputs connected at the counter from bit position i . Due to those dependency restrictions for $C_{o,1}$, $C_{o,2}$, the Boolean stage must comprise two levels of gates: a first level which generates three outputs ($C_{o,1,i}$, $C_{o,2,i}$, $Y_{0,i}$ in dual-rail form) followed by a second stage generating single-rail C_i and S_i outputs.

To evaluate the potential implications of the proposed hybrid counter/compressor implementation scheme, we present in the next Section a comparison between the performance and associated cost for parallel 7/3 counters and 7|2 compressors implemented using Boolean logic (full-adders), TL, and HTBL.

4. SIMULATION RESULTS AND COMPARISONS

We compared the HTBL 7/3 counters and 7|2 compressors, having the logic designs as presented in Section 3, with traditional full-adder based schemes from [2] (pp. 129-130), [7], and with TL schemes proposed in [4, 9], respectively. We choose the full-adder based schemes for our comparisons since they have been proved in literature to be the nearly optimal Boolean approach in terms of logic depth (and therefore speed) for the specific compression ratio taken into account ($7 : 3 = 2.33$, $7 : 2 = 3.5$). A 7-input TL gate, as in Figure 1, was designed and optimized in .25 μ m feature size CMOS technology since such gate was used extensively in both TL and HTBL counters and compressors. It has 250ps worst case delay, with more than 17% faster than a full-adder, in the same technology which has a worst case delay of $\Delta_{FA} \approx 300ps$. The circuits were simulated with HSpice. Correct operation of TL and HTBL counters and compressors was verified by extensive simulations under the process corners. Every simulated counter was loaded with similar counters/compressors in order to have a more clear image of the delay improvement in a real partial product reduction matrix. We would like to stress out that the delay penalty caused by capacitive loading in the succeeding stages is minimized with our TL gate since the capacitive load per input is only half the value of a minimum inverter (see Figure 1).

Tables 1 and 2 present the 7/3 counters and 7|2 compressors characteristics, in terms of absolute delay (@ 27°C, $V_{dd} = 2.5V$), normalized delay, and total number of transistors. For TL and HTBL counters and compressors "delay" signifies "latency" (data-output delay) as a clock is needed for their operation.

Table 1 suggests that 7/3 HTBL counters, when compared with Boolean, are with 53% faster at the expense of 74% more transistors. Table 2 presents the delay data of 7|2 compressors. As ex-

pected, HTBL 7|2 compressors are slower than similar 7/3 counters (1.15 vs. 1.63 equivalent FA delay) since they have two gates in the Boolean stage. However, 7|2 HTBL compressors are 61% faster when compared with Boolean compressors, at the expense of 67% more transistors. TL 7/3 counters and 7|2 compressors [4, 9] are with more than 30% slower than HTBL designs. HTBL counters and compressors compare favorably in terms of speed with either full-adder based Boolean and Threshold logic counterparts since they take the advantages of both Threshold logic (fast parallel processing of large number of inputs in the first stage) and Boolean logic (high-speed implementation of And-Or like Boolean functions in the second stage).

5. CONCLUSIONS

In this paper we proposed high-speed hybrid Threshold-Boolean logic counters and compressors employed in parallel multiplication and multiple operand addition. First, we presented a depth-2 hybrid implementation scheme for arbitrary symmetric Boolean functions, based on differential Threshold logic gates as circuit style. Subsequently, we applied the previous general scheme to parallel p/q counters and $p|2$ compressors. Finally, we presented hybrid logic designs of a 7/3 counter and a 7|2 compressor. The simulation results, suggest that the hybrid 7/3 counter and the 7|2 compressor designed in .25 μ m CMOS, achieve between 53% and 61% higher speed when compared with traditional Boolean logic and Threshold logic counterparts, at expense of between 67% and 74% more transistors.

6. REFERENCES

- [1] L. Dadda. Some schemes for parallel multipliers. *Alta Frequenza*, 34:349-356, 1965.
- [2] I. Koren. *Computer arithmetic algorithms*. A. K. Peters, 2002.
- [3] Y. Leblebici, H. Ozdemir, A. Kepkep, and U. Cilingiroglu. A compact high-speed (31,5) parallel counter circuit based on capacitive Threshold-logic gates. *IEEE Journal of Solid-State Circuits*, 31(8):1177-1183, August 1996.
- [4] R. Minnick. Linear-input logic. *IRE Transaction on Electronic Computers*, EC-10:6-16, March 1961.
- [5] S. Muroga. *Threshold logic and its applications*. Wiley and Sons Inc., 1971.
- [6] M. Padure, S. Cotofana, C. Dan, M. Bodea, and S. Vassiliadis. A new latch-based Threshold logic family. *Proceedings of International Semiconductor Conference, CAS2001*, 2:531-534, October 2001.
- [7] M. Santoro and M. Horowitz. SPIM: a pipelined 64X64-bit iterative multiplier. *IEEE Journal of Solid-State Circuits*, 24(2):487-493, April 1989.
- [8] P. Song and G De Micheli. Circuit and architecture trade-offs for high-speed multiplication. *IEEE Journal of Solid-State Circuits*, 26(9):1184-1198, September 1991.
- [9] S. Vassiliadis and S. Cotofana. 7|2 counters and multiplication with Threshold logic. *Proceedings of 30th Asilomar Conference on Signals, Systems and Computers*, pages 192-196, November 1996.
- [10] S. Vassiliadis, S. Cotofana, and K. Bertels. 2-1 addition and related operations with Threshold logic. *IEEE Transactions on Computers*, 45(9):1062-1068, September 1996.