Development and Synthesis Method for Pass-Transistor Logic Family for High-Speed and Low Power CMOS

Vojin G. Oklobdzija, Michael Soderstrand
Electrical and Computer Enginering
University of California
Davis, CA 95616
e-mail: vojin@ece.ucdavis.edu

Benoit Duchêne
Ecole Superieure d'Ingenieurs en Electrotechnique et Electronique
93162 Noisy le Grand CEDEX FRANCE

ABSTRACT

This paper presents a new pass-transistor logic termed DVL which contains fewer transistors and has better performance than other CMOS logic families. A method for synthesis of DVL is also developed and demonstrated. This new logic has advantages over CMOS and is characterized by excellent speed and low power.

I. INTRODUCTION

New CMOS logic families using pass-transistor circuits have recently been proposed with the objective of improving speed and power [4-6,8]. Two of them, simultaneously developed by Hitachi: CPL [4] and DPL [6], are the most notable. The Double Pass-Transistor Logic, developed by Hitachi in 1993 demonstrated an 1.5nS 32-bit ALU in 0.25 μm CMOS technology [6] and a 4.4nS 54×54 bit multiplier [9]. However, DPL has not yet been fully adopted because of its use of p-type transistors. CPL logic has been advantageous because it uses only n-type transistors, however it suffers from the problem of the threshold voltage drop at the output. The objective of this work is to develop a new logic family and the synthesis method for pass-transistor logic which will minimize the number of transistors used in DPL and yet preserve the speed of the logic. We used 1-μm CMOS technology to test our examples and make comparisons with DPL.

II. NEW LOGIC FAMILY: DVL

The new logic family was obtained from DPL by the elimination of the redundant branches and rearrangement of signals [10]. These simplifications still preserve full swing operation of DPL and improve speed. The speed improvement is a direct result of elimination of one branch containing one transistor. This minimizes the capacitive load "seen" by the previous gate by minimizing the number of inputs and of capacitive loads.

The new logic family termed DVL (Dual Value Logic) is achieved in two steps:

(a) elimination of redundant branches in DPL

(b) elimination of branches via signal rearrangement

(c) combination of (a) and (b) using two better halves

The resulting DVL gate contains total of 8 transistors (3 p-transistors and 3 n-transistors) compared to DPL consisting of 4 transistors of each type. There is total of 9 inputs in DVL versus 12 in DPL resulting in a smaller capacitive load of DVL gate. Of those inputs 3 are connected to the transistor source and 6 to the gate (3 to p-type and 3 to n-type). In DPL, four transistors are connected to the source (4 to p-type and 4 to n-type transistors). The total DVL gate (taking resizing into account) is only 5% larger than the DPL gate. The speed advantage is 20% in favor of DVL.

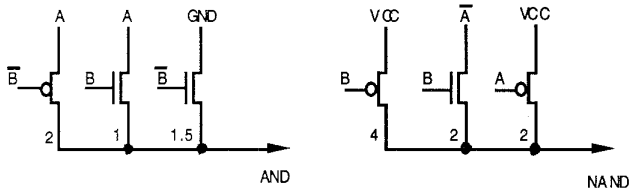

Fig. 1. *Resulting DVL gate*

In summary, the comparison between NAND/AND DPL gate and NAND/AND DVL shows:

- *20% speed improvement, utilizing 75% of the transistors used in DPL.*
- *25% less connections and wires as compared to a DPL gate.*
- *The 4% area increase in comparison to DPL is not found to be substantial.*

Similar methods can be used to build the NOR/OR gates.

III. DVL SYNTHESIS METHOD

In this work we developed a method for synthesis of DVL based on transistors instead of logic gates. In place of conjointly assembling several basic gates, functions are synthesized at the transistor level. In addition, a programming of this method has been developed to prove its efficiency.

The key point of DVL synthesis consists of employing a Karnaugh-Map at the transistor level. Thus, we are not cascading several logic gates in order to implement a given function. Instead, we are building functions by directly using several transistor levels in series. A *"Pseudo Karnaugh-Map"* is used when the number of inputs is less than 8 because its explanation is simple. A *"Pseudo Quine McCluskey"* technique has been programmed instead.

Pseudo Karnaugh Map

Usually, the general Karnaugh-Map covers "0s" or "1s", in such a way that a minimized Sum of Products (or Product

0-7803-2972-4/96$5.00©1996 IEEE

of Sums) is obtained. In our case we allow for four classes of loops to directly synthesize the final circuit. It is necessary to cover all the implicants in our case, as illustrated in Fig. 2.

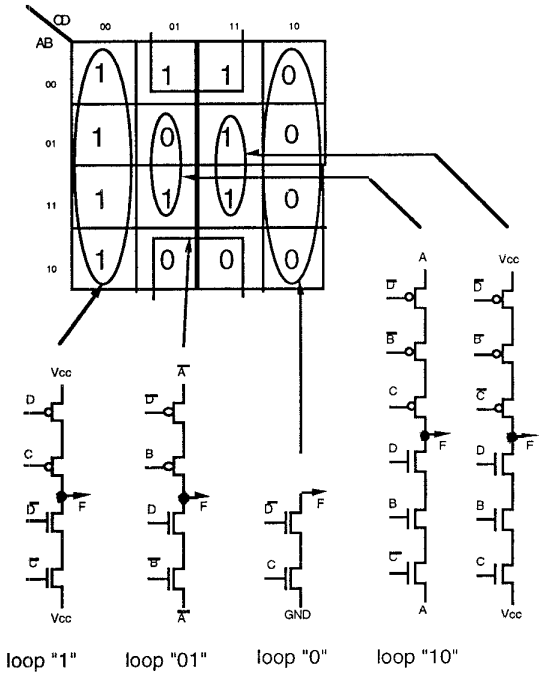

Fig. 2. *Pseudo Karnaugh Map*

The synthesis of circuits from functional expressions is summarized in Fig. 3. There are two important steps in this flowchart which are critical:

1. The first step determines how to cover the Karnaugh-Map at the transistor level for an N-input function.

2. The second step, based on circuit synthesis rules, deals with acquiring suitable sizes for transistors in order to minimize delay. The last phase, reduction of the number of transistors is accomplished in this second step when the list of nodes, components, and nets is automatically generated.

DVL Synthesis Rules

The rules for synthesis of DVL circuits are described here. The covering of the "pseudo Karnaugh Map" has the following meaning:

(a) A loop corresponds to one branch in DVL circuit.

(b) The loop control variables are the inputs to the gates of transistors in the branch.

(c) The values are attached to the branch input and represent:

- GND for a "0" loop.
- VCC for a "1" loop.
- One input (pass signal) for "01" or "10" loop.

(d) For the loop covering "0s" nMOS transistors connected to GND are used.

(e) For the loop covering "1s" , we use pMOS transistors but add a twin nMOS transistor branch, where the control variables are complement of those connected to the gate of pMOS transistors. The purpose of this is to avoid a high speed degradation.

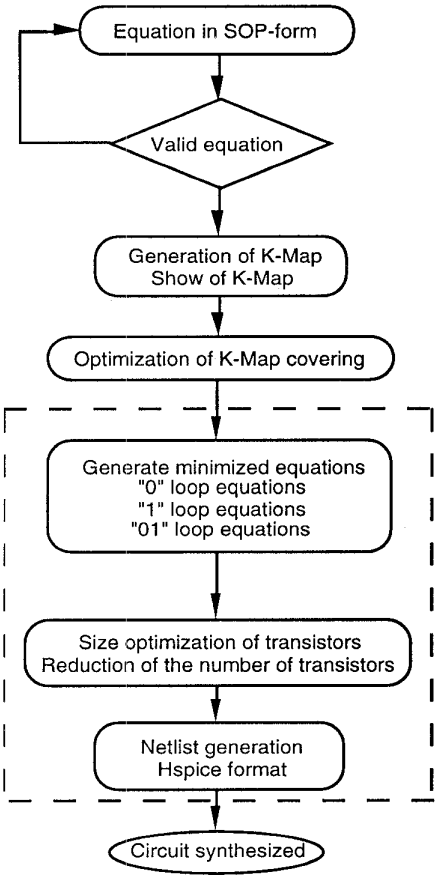

Fig. 3. *General Flowchart*

(f) For "10" or "01" loops, nMOS and pMOS branches are used. They terminate at the variable.

This results in two benefits which are the basics for DPL logic family:

(a) full swing and (b) compensation of speed degradation using nMOS in parallel to pMOS transistor.

Rules for Transistor Sizes

In general, transistor sizes are fixed in standard NAND/AND, NOR/OR, XNOR/XOR gate synthesis. However the transistor size in our case has considerable effect on the delay

due to the existence of different branches connected to the gate output.

Because of undesirable input configurations, there exists a slow path which can not be avoided. Therefore, transistor sizes need to be enlarged to match the speed of the faster paths. By choosing bigger transistors, the input capacitance and the output capacitance of those transistors leads to an increased load because the size of these capacitances is increased. Thus, a trade-off occurs in determining optimal transistor sizes for the different branches of the logic network. Those rules are used to determine the width to length W/L ratio as described in Fig. 4.

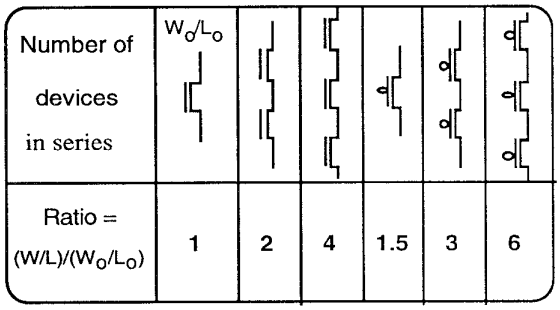

Fig. 4. *Determination of transistor sizes*

Reduction of the number of transistors

A reduction in the number of transistors is automatically accomplished during the netlist generation by the synthesis program. At this step, we can eliminate redundant transistors by examining the resulting netlist. This process involves the merging of two pMOS or two nMOS transistors with the same control variable that belong to a parallel branch, as shown in Fig. 5.

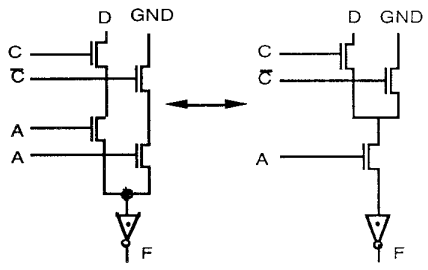

Fig. 5. *Circuit simplification*

IV. RESULTS

A comparison between DVL and conventional CMOS is given in Table 1. The improvement in the global size, the number of transistors and the delay of the circuit is shown.

The simulation results are shown in Fig. 6. showing the output waveforms at the output of the given function F.

Table 1: Comparison between DVL and CMOS

Function F	CMOS	DVL	Savings
Transistors	10 nMOS 10 pMOS	8 pMOS 8 nMOS	20%
Levels	3 gate	2 transistor	
Global size	44	36	18%
Delay	430pS	245pS	43%

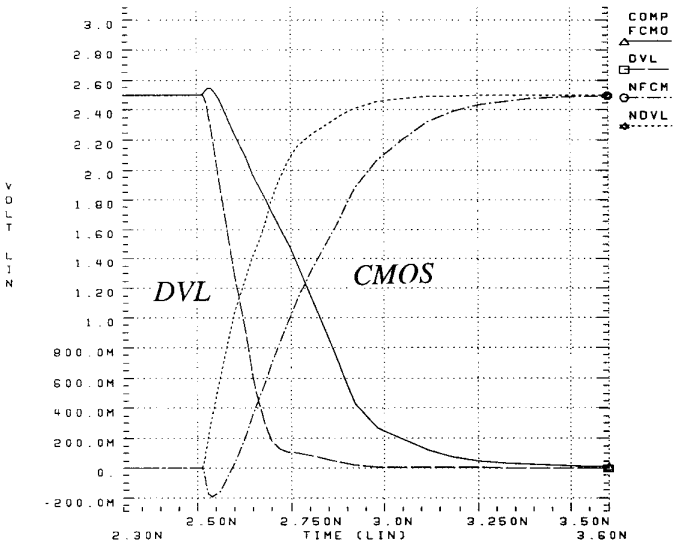

Fig. 6. Delay Comparison between DVL and CMOS for a 3 inputs function $F = \overline{B}C + AB\overline{C}$

Comparison with DPL

The comparison results are shown in Table 2.

Table 2. Comparison between DPL and DVL

Function	DPL	DVL	Savings
Transistors	16 pMOS 16 nMOS	8 pMOS 8 nMOS	50%
Levels	2 gate	2 transistor	
Size	48	30	37.5%
Delay: @50%	290pS	120pS	58.6%
@80%	350pS	240pS	31.4%

The worse case analysis of the signal delay is given by the simulation diagrams shown in Fig. 7. The DVL implementation of the function $F = \overline{B}C + AB\overline{C}$ results in faster circuit than DPL realization.

V. CONCLUSION

In this work DVL logic family has been developed. This logic has advantages over standard CMOS as well as new pass-transistor families such as DPL and CPL. The improvement over DPL is in 25% less transistors with the advantages of decreasing the number of connections and wires. The second advantage involves the method developed to automatically build DVL logic. Instead of using common method in which circuits are synthesized at the gate level, DVL synthesis generates circuits directly at the transistor level. The advantages of DVL synthesis lay in the three main domains : Area, Speed and Power consumption. In each case, the global size of transistors used in DVL was smaller compared to other circuits. In comparison with conventional CMOS DVL shows performance improvement of up to 43% while the improvement in size ranges around 20%. In comparison with DPL circuits, there is 15% to 50% improvement in speed. However, the exact speed improvement is dependent on each particular circuit. As far as the power consumption is concerned comparison between DVL and the Conventional CMOS shows a 30% to 50% savings in favor of DVL. Generation of DVL is supported by an automated synthesis tool based on the algorithm developed in the course of this work.

REFERENCES

[1] LG Heller, WR Griffin, et al, "Cascode Voltage Switch Logic : A Differential CMOS Logic Family" *1984 IEEE International Solid-State Circuits Conference*, vol 27, pp16-17 February 1984.

[2] KM Chu and DL Pulfrey, "A Comparison of CMOS Circuit Techniques : Differential Cascode Voltage Switch logic Versus Conventional Logic" *IEEE J.Solid State Circuits*, vol sc 22, pp528-532, 1987.

[3] F.S. Lai and W. Hang, "Differential Cascode Voltage Switch with Pass Gate Logic Tree for High Performance CMOS Digital Systems", *1993 International Symposium on VLSI Technology, Systems and Applications*, pp358-362, May 1993.

[4] Yano, K, et al, "A 3.8 ns CMOS 16 *16 b Multiplier Using Complementary Pass-Transistor Logic", *IEEE J. Solid State Circuits*, vol 25, p388-395, April 1990.

[5] Akilesh Parameswar, et al, "A Swing Restored Pass-Transistor Logic Based Multiply and Accumulate Circuit for Multimedia Applications", *IEEE 1994 Custom Integrated Circuit Conference*, pp278-281.

[6] Makoto Suzuki, et al, "A 1.5 ns 32 b CMOS ALU in Double Pass-Transistor Logic", *1993 ISSCC Dig. Tech. Papers*, pp90-91, February 1993.

[7] KM Chu and DL Pulfrey, "Design Procedures for Differential Cascode Voltage Switch Circuits", *IEEE J.Solid State Circuits*, vol sc 21, no 6, December 1986.

[8] Yano, K, et al, "Lean Integration : Achieving a Quantum Leap in Performance and Cost of Logic LSIs", *Proceedings of the IEEE 1994 Custom Integrated Circuit Conference*, May 1-4, 1994, San Diego, California, p.603-606.

[9] N. Ohkubo, et al, "A 4.4nS CMOS 54x54-b Multiplier Using Pass-Transistor Multiplexer",*Proceedings of the IEEE 1994 Custom Integrated Circuit Conference*, May 1-4, 1994, San Diego, California, p.599-602.

[10] V.G. Oklobdzija, B. Duchene, "Pass-Transistor Dual Value Logic For Low-Power CMOS," *Proceedings of the 1995 International Symposium on VLSI Technology*, Taipei, Taiwan, May 31-June 2nd, 1995.

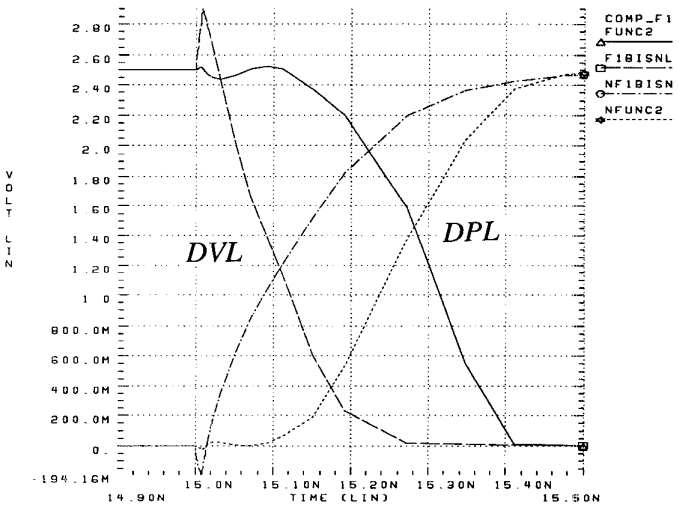

Fig. 7. *Delay of the 3 inputs function F: DPL vs. DVL*