Proceedings of the 1995 International Symposium on VLSI Technology
May 31 - June 2, 1995, Taipei, TAIWAN

Pass-Transistor Dual Value Logic for Low-Power CMOS

Vojin G. Oklobdzija, B. Duchêne*
Advanced Computer System Engineering Laboratory,
Electrical and Computer Engineering Department
University of California Davis
(916) 752-5634
vojin@ece.ucdavis.edu
*Ecole Superieure d'Ingenieurs en Electrotechnique et Electronique
93162 Noisy le Grand CEDEX FRANCE

ABSTRACT

This paper presents new pass-transistor logic termed DVL which contains fewer transistors than its counterpart DPL yet maintaing comparable performance. A method for synthesis of such networks is also developed and demonstrated in this paper. The new logic is characterized by good speed and low power. The simulations and tests were performed using 1-μm CMOS.

I. INTRODUCTION

New logic CMOS families using pass-transistor circuit techniques have recently been proposed with the objective of improving speed and power consumption [1-6]. This logic (in most cases) passes the charge between the nodes rather than charging the nodes from V_{CC} and then discharging them to GND. This feature contributes to less power being used as compared to the regular CMOS. The Double Pass-Transistor Logic (DPL), developed by Hitachi demonstrated an 1.5nS 32-bit ALU and 4.4nS 54-bit multiplier in 0.25 μm technology [4,5]. However, DPL has not yet been fully adopted because of its high transistor count. The objective of the new logic gates and the synthesis method developed for pass-transistor logic is to minimize the number of transistors used in DPL and preserve the speed of the logic.

II. NEW LOGIC GATES

The new logic gate represents an improvement over DPL family achieved by the elimination of the redundant branches and rearrangement of signals. This simplification, illustrated in Fig. 1, 2 and 3, preserves the advantages of DPL gates which are:

a) Compensation of speed degradation due to the use of pMOS transistors.

b) Straightforward full swing operation.

This simplification is achieved by in three steps:

A. *Elimination of the redundant branches*

This simplification is achieved by eliminating the redunant branches (shown in shaded area) from DPL. Most of the pull up and pull down transition times, in the resulting configuration, surpass those of the DPL gates. However, the improved gate has some undesirable input configurations in which the current path is supplied by a single transistor instead of a double pass-transistor (as in the case in DPL), making this transition time worse. To avoid degradation of delay due to the use of just one pMOS transistor, the particular transistor width is increased. The elimination of redundant branches is illustrated in Fig. 1. The resulting two halves (which constitute the gate) are not of the same speed. The faster half is NAND (60pS) and the slower is AND (70pS), which is still being faster than DPL (75pS).

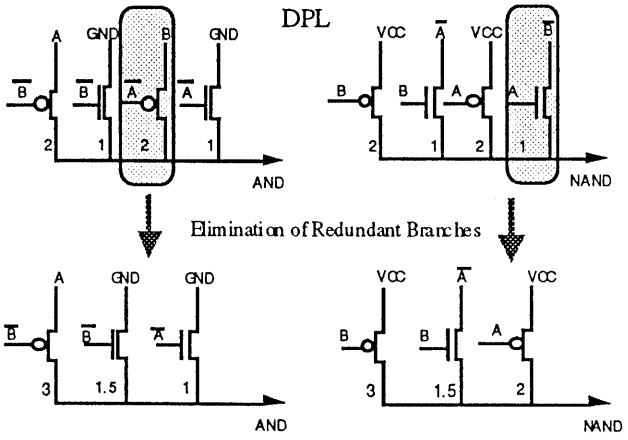


Fig. 1. Elimination of redundant branches

B. *SiMrrangement*

The use of two parallel pMOS transistors is avoided by simple signal re-arrangement because the two pMOS transistors contribute more to the delay than one pMOS transistor in parallel with one nMOS transistor. This is especially true in a pull up operation. The current is always provided by one nMOS transistor alone or by one nMOS and one pMOS in parallel. The AND/NOR DPL gate (in Fig. 2.) is obtained from NOR/OR DPL configuration whose inputs are simply inverted. Signal rearrangement applied to AND/NOR DPL gate results

in an AND gate configuration which is faster than DPL (60pS vs. 75pS), where AND is a faster half.

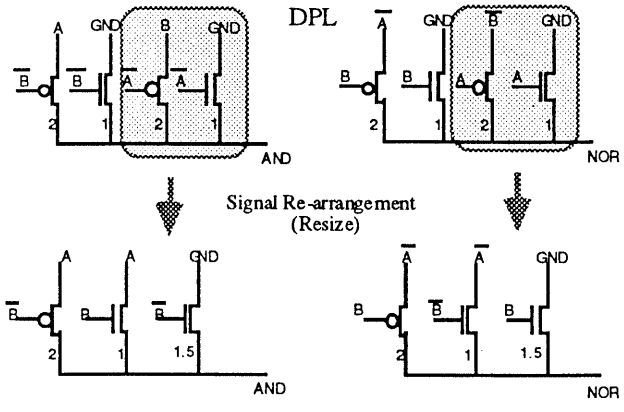


Fig.2. Signal Re-arrangement

C. *Selection of the faster halves*

Finally we take a faster half from Fig.1. and form Fig.2. The resulting AND/NAND complementary logic gate (shown in Fig.3.) is obtained by elimination of the redundant branches for the NAND and rearrangement of signals for the AND gates respectively. We named this logic: DVL (Dual Value Logic). The resulting AND/NAND DVL gate contains a total of 6 transistors as compared to DPL consisting of 4 transistors of each type. There is a total of 9 inputs in DVL versus 12 in DPL resulting in a smaller capacitive load of DVL gate. In DVL 3 inputs are connected to the transistor source and 6 to the gate (3 to p-type and 3 to n-type). In DPL 4 inputs are connected to the source and 8 to the gate (4 to p-type and 4 to n-type transistors).

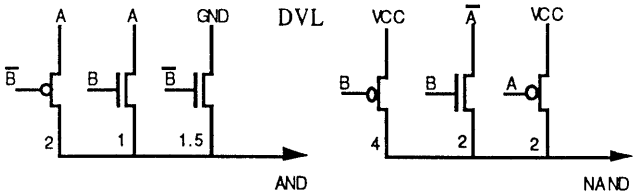


Fig. 3. Resulting DVL Gate

The comparison between NAND/AND DPL gate and DVL shows:

- *20% speed improvement, using 75% of transistors used in DPL gate.*
- *25% less connections and wires than in DPL gate. The 4% area penalty comparison to DPL is quite negligible.*

Similar arguments can be used to build the NOR/OR gates from DPL gates.

III. SYNTHESIS METHOD

The synthesis method for DVL is based on the method used to create the logic gates described before. At present, there is no known algorithm to find minimal multi-stage logic circuits. The new proposed method for synthesis of DVL, is based on transistors instead of logic gates. In place of conjointly assembling several basic gates, functions are synthesized at the transistor level. In addition, the programming of this method has been developed to prove the efficiency of the theory presented.

The key point of DVL synthesis consists of employing Karnaugh-Map at the transistor level. Thus, we are not cascading several logic gate levels (NAND/AND or NOR/OR) but building functions by directly using several transistor levels in series. However, the choice of *pseudo Karnaugh-Map* in the programming for less than 8 inputs was done because its explanation is simple, but a *pseudo Quine McCluskey* technique could have been adopted instead.

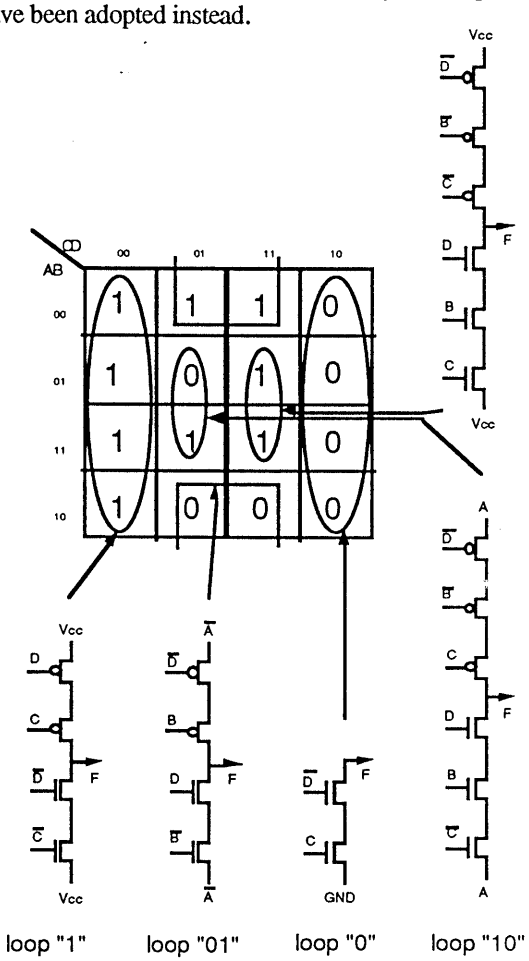


Fig. 4. Loops allowed for 4 inputs

Usually, the general Karnaugh-Map is covered by loops of "0" or "1", in such a way that a minimized Sum of Products form is obtained. In our case, four classes of loops are allowed (as illustrated in Fig. 4) to directly synthesize a part of the final circuit. Accumulating all loops necessary to cover the Karnaugh-Map yields the resulting circuit.

IV. RESULTS

The best way to compare efficiency of the presented algorithm is via synthesizing and simulating circuits obtained using our automated algorithm, and comparing it to circuits produced by CPL and DPL using the concept of logic gates.

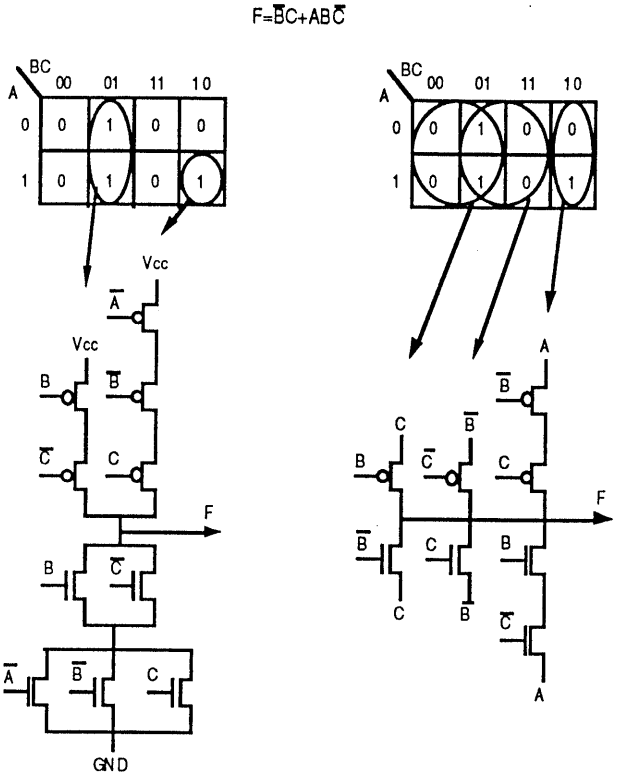


Fig. 5. Example showing implementation of the function F in DVL and conventinal CMOS

The DVL synthesis was compared not only to the Conventional CMOS, but also to DPL circuits [4] and CPL circuits using lean cells [6]. An example of DVL synthesis versus Conventional CMOS synthesis (given in Fig. 5.) shows the improvement in global size, the number of transistors and the delay of the circuit. In each circuit, the global size of DVL is smaller compared to other circuits.

The comparison with AND/NAND DPL gate shows 25% less transistors resulting in 25% less connections and wires in an equivalent DVL gate, keeping the total transistor area constant. Similar methods can be used to build the NOR/OR gates.

A. *Comparison with CMOS*

A comparison between DVL and conventional CMOS is given in Table I, while the simulation results (for the given function $F2 = \overline{BC + ABC}$) are shown in Fig. 6.

TABLE I.
COMPARISON BETWEEN DVL AND CONVENTIONAL CMOS

Function F2	CMOS	DVL	Savings
No. of Transistors	10nMOS 10pMOS	8pMOS 8nMOS	20%
No. of Levels	3 gate levels	2 transistor levels	
Global size	44	36	18%
Delay @ 50% of Vout	430pS	245pS	43%
Transitor ratio	Wp/Wn= 2	Wp/Wn=2	

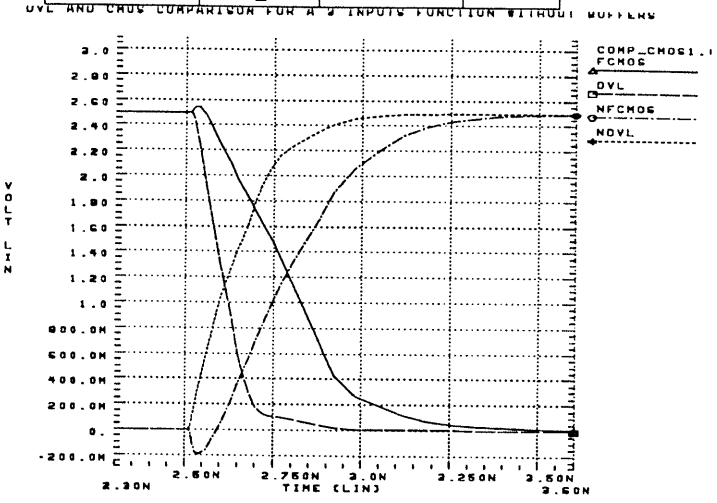


Fig. 6. Delay Comparison between DVL and Conventional CMOS for a 3 inputs function F2

B. *Comparison with DPL*

The function used for comparison is a three variable function $F_2(A,B,C)$, where $F2 = \overline{\overline{BC} + ABC}$ and $\overline{F2} = \overline{BC} + ABC$. This function was implemented using 4 DPL gates in two logic levels. Afterwards this circuit was built using DVL. The load applied to the output is a standard load of two gate inputs. The comparison results are shown in Table II and timing simulation for the function F_2 are shown in Fig. 7.

TABLE II.
COMPARISON BETWEEN DPL AND DVL FOR A 3 INPUTS FUNCTION F2

Function F2	DPL	DVL	Savings
No. of Transistors	16pMOS 16nMOS	8pMOS 8nMOS	50%
No. of Levels	2 gate levels	2 transistor levels	
Global size	48	30	37.5%
Delay @50%	290pS	120pS	58.6%
@80%	350pS	240pS	31.4%
Transistor ratio	Wp/Wn=2	Wp/Wn=2 2 nMOS=1.5	

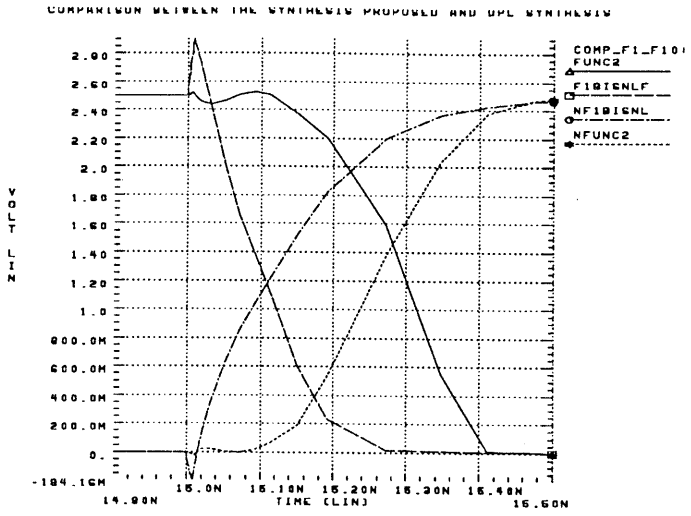
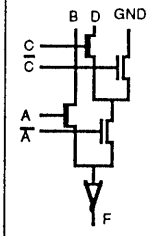


Fig. 7. Simulated delays for the 3 inputs DPL function F2 and DVL implementation of F2

C. *Comparison with CPL*

The function F published by Yano in [6] was synthesized for DVL and compared to CPL which uses lean cells and special inverters [6]. Delays were measured for 1 cell, 2 cells and 3 cells cascaded. DVL circuit is made with conventional inverters. The comparisons were made using the output load of 15 FF in both cases.

TABLE III.
COMPARISON BETWEEN CPL AND DVL FOR A 4 INPUTS FUNCTION.

	1 cell	2 cells	3 cells	Cell size
	375pS (100%)	760pS (100%)	1,150pS (100%)	105µ (100%)
Circuit F in DVL	380pS (101%)	660pS (86%)	950pS (82%)	108µ (103%)

The comparison between CPL and DVL is shown in Table III and the delays of the two cascaded CPL and DVL cells are compared in Fig. 8.

CONCLUSION

DVL logic family has been developed which has advantages over standard CMOS as well as new pass-transistor families such as DPL and CPL. However, the exact speed improvement is dependent on each particular circuit. The power consumption is also reduced for 30-50% over conventional CMOS. Generation of DVL is supported by an automated synthesis tool based on the algorithm developed in the course of this work.

ACKNOWLEDGMENT

The authors acknowledge the contributions of P. Lee, W. Wong and H. Yu.

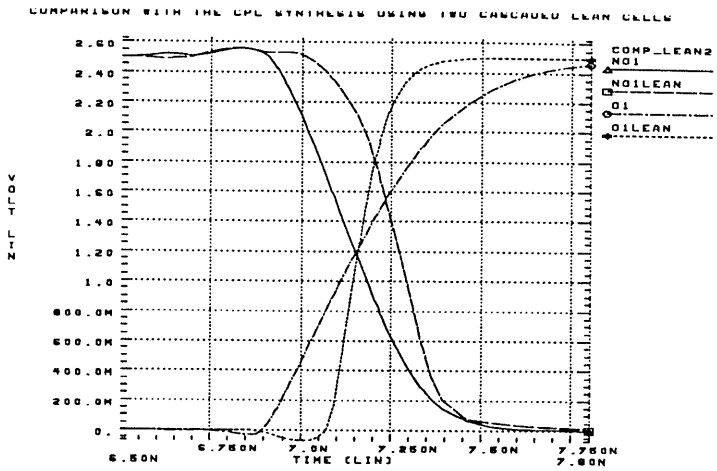


Fig. 8. Delays comparison between 2 cascaded DVL and CPL cell.

REFERENCES

[1] 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.

[2] Yano, K, et al, "A 3.8 ns CMOS 16X16-b Multiplier Using Complementary Pass-Transistor Logic", *IEEE J. Solid State Circuits*, vol 25, p388-395, April 1990.

[3] Akilesh Parameswar, et al, "A Swing Restored Pass-Transistor Logic Based Multiply and Accumulate Circuit for Multimedia Applications", *Proceedings of the IEEE 1994 Custom Integrated Circuit Conference, San Diego, California, May 1-4, 1994.*

[4] 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.

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

[6] 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, San Diego, California, May 1-4, 1994.*