A New Logic Synthesis and Optimization Procedure

*Kuo-Hsing Cheng** and *Ven-Chieh Hsieh*

Department of Electrical Engineering, Tamkang University
Tamsui, Taipei Hsien 25137, TAIWAN
E-mail: **cheng@ee.tku.edu.tw***

ABSTRACT

The objective of this work is to develop a new logic circuit synthesis and optimization procedure for arbitrary logic function. Following the procedure, we may get a new high performance logic circuit family, which has low power consumption, low power-delay product, area efficiency and suitable for low supply voltage. The new logic family based upon the proposed design procedures has certain advantage over CMOS, DVL and DPL in most cases.

1. INTRODUCTION

At logic circuit design level, a proper choice of a circuit style for implementing combinational logic is very important. For example, in the NOR gate implementation, the static CMOS logic circuit structure seems the better logic circuit family than the DVL [1], DPL[2] or any other logic circuit families. But when it comes to 2-input XOR logic implementation, the static CMOS logic circuit family becomes the worst choice. This result may confuse someone in logic circuit family selecting.

The objective of this work is to develop a new logic circuit synthesis and optimization procedures for arbitrary logic function implementation. Following the synthesis and optimization design procedures, we may get a new high performance logic circuit family, which has low power consumption, low power-delay product, area efficiency and suitable for low voltage supply. From the example shown below, this new logic family based upon the proposed design procedures has advantage over CMOS, DVL and DPL in most cases. We use 0.35μm CMOS technology to design the circuits and to do the post layout simulation in all comparisons.

Some basic concepts and definitions of the proposed new logic synthesis and optimization procedures are shown in Section II. The detail procedures of the new logic style and a circuit implementation example are given in Section III. Comparisons between Static CMOS, DPL, DVL and the New Logic Style are listed in Section IV. Some conclusions are finally drawn in Section V.

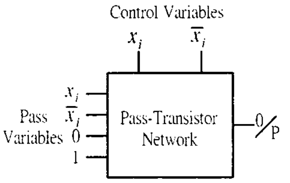

Fig.1. Pass-Transistor Network.

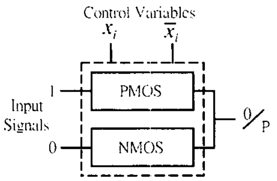

Fig.2. Static CMOS Network.

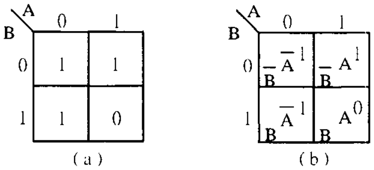

Fig. 3. (a) The K-map of the NAND Function
(b) The Modified K-map of the NAND Function.

2. BASIC COMCEPT AND DEFINITIONS OF THE NEW LOGIC CURCUIT DESIGN

In view the problems shown in previous section, we can implement a arbitrary logic function by using the pass transistor logic network as shown in Fig.1. The input signals Xi and $\overline{\text{Xi}}$ can be used as the control variables or pass variables of the pass-transistor network. The control variables are connected to drive the gate of the MOS transistors. The pass variables are connected to the sources/drains of the MOS transistors. In general, the static CMOS logic circuit structure can be seem as a special case of pass transistor logic network that the pass variables input signals are just "1" and "0", and the input signals Xi and $\overline{\text{Xi}}$ are connected to drive the gate of the MOS transistor as shown in Fig.2. Thus, it is possible to develop a synthesis and optimization procedure of the pass transistor logic circuit for arbitrary logic function and high performance applications.

The proposed new circuit synthesis and optimization procedures are based upon the pass transistor logic circuit implementation. As shown in Fig.4, we use 2-input XOR function as a circuit implementation example. The detail design flow of the circuit will be shown in following. In order to describe the proposed new procedures clearly, some important notations and circuit implementation of the

0-7803-6685-9/01/$10.00©2001 IEEE

synthesis and optimization procedures, *Square*, *Modified K-map, Loop Circling, Selected Set, Implicate Loop* and *Circuit Implementation Methods* are shown as following.

A. Square

The *Karnaugh map (K-map)* of a function specifies the value of the function for every combination of values of the independent variables. The *Square* indicates a function output state on the K-map. As shown in Fig. 3(a), the output state of the Square for {A=B=0}, plotted in the upper left on the K-map, is "1".

B. Modified Karnaugh map (K-map)

The *Modified K-map* is almost the same with the *K-map*, except that not only the power lines ("1"and "0") but also the input variables ("Xi" and "$\overline{X_i}$") are listed in the *Square* to represent the function result as shown in Fig. 3(b). It is straightforward to implement the circuit based on the Static CMOS Logic according to the *K-map*. And the *Modified K-map* provides us the thoughts of implementation of the new logic synthesis and optimization procedures.

C. Loop Circling

The *Loop Circling* is the method to implement the pass transistor circuit. A loop contains one or more squares on the *Modified K-map*. For example, the *Square* {A=B=0} and *Square* {A=1, B=0} combine to form the Loop (i) by looping the corresponding A's on the *Modified K-map* in Fig. 4(a).A loop may contains all squares that are never selected by other loops (Loop(iii)in Fig. 4(a)), or part squares are selected by other loop (*Square* {A=B=0} in Loop(ii) in Fig. 4(a) is selected by Loop(i)).

D. Selected Set

We define a set of controlling and passing variables that ever used for circuit implementation, call *Selected Set*. Which means we can choose the variables in the *Selected Set* for implementing new circuit without extra inverters to generate the newly complementary signals. The initial values in the *Selected Set* are {0, 1, Xi}. For example, the initial variable in the *Selected Set* is {0, 1, A, B} in Fig. 4(a). After every loop circling, we put the new selected passing and controlling variables in the *Selected Set* immediately. For example, the *Selected Set* is {0,1,A, B, $\overline{B}$ } after Loop (i) is circled. The select of variables in the *Selected Set* to implement new circuit is based on the choosing priority "0">"1">"Xi">"$\overline{X_i}$" (in the *Selected Set)* >"$\overline{X_i}$" (not in the *Selected Set).*

E. Implicate Loop

An *Implicate Loop* may include partial or all squares that are already chosen by a selected Loop. For example as in Fig. 4(a), the Loop (ii) can be seen as a *Implicate Loop* to the Loop (i), cause the *Square* {A=0, B=0} is circled again. And due to the Loop (iii)(the *Square*{A=1, B=1}) is not circled by any other selected loops, so it is a *Non-Implicate Loop*.

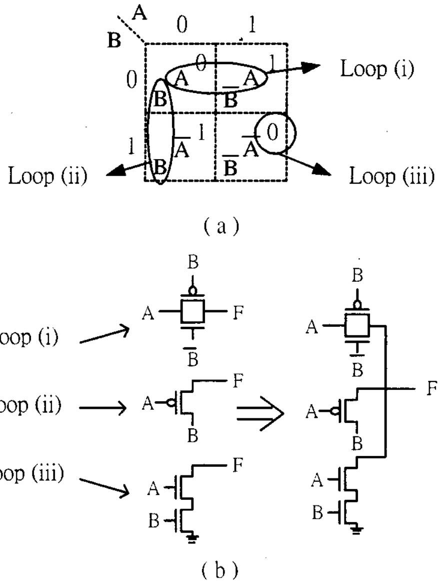

Fig. 4. (a) The Modified K-map and Loop Circling of the 2-Input XOR Function. (b) The Circuit Implementation of the 2-Input XOR Function.

F. Circuit Implementation Methods

For pass transistor circuit implementation, we will only concern those squares are newly choose in the current loop. If a loop contains newly outputs states has both 1's and 0's (Loop (i) in Fig. 4(b)), its pass-transistor circuit switch is implemented by both NMOS and PMOS (a transmission gate is used). The PMOS is used to implement all 1's loop (Loop (ii) in Fig. 4(b), the A=0, B=1 *Square* is only concerned) and the NMOS is used to implement all 0's loop (Loop (iii) in Fig. 4(b)).

3. Circuit Synthesis and Optimazation Procedures

3.1. The Synthesis and Optimization Procedures

The circuit synthesis and optimization procedures, based up the circuit implementation concepts list above are shown in following:

Step 1. Build the *Modified K-map* of the given logic function.

Step 2. Circle The Largest *Non-Implicate Loops*:

Case(a). Only one largest loop: Choose that Loop to implement the pass transistor logic switch circuit, even the passing or controlling variables do not exist in the *Selected Set.* Then repeat the Step2 and the circling subject is changing to the second largest *Non-Implicate Loop.*

Case(b). Many largest loops: The loops selecting is following the priority in the *Selected Set.* But we will skip the loop either the passing variable cannot be found in the *Selected Set* or it is an *Implicate Loop*.

Step 3. Circle The Largest *Implicate Loops* and Second Largest *Non-Implicate Loops*: (The loop will not be chosen if it's passing variable is not found in the *Selected Set.*)

Case(a). The largest *Implicate Loop* can include more amounts of un-chosen squares than second largest *Non-Implicate Loops*: Choose the largest *Implicate Loop* to implement the pass transistor logic switch circuit.

Case(b). The largest *Implicate Loop* can include equal amounts of un-chosen squares than second largest *Non-Implicate Loops*: The selecting will follow the rules of: 0's(largest *Implicate Loop*) > 1's(largest *Implicate Loop*) > 0's(second largest *Non-Implicate Loop*) > Xi's (largest *Implicate Loop*) > 1's(second largest *Non-Implicate Loop*) > $\overline{\text{Xi}}$'s (largest *Implicate Loop*) > Xi's (second largest *Non-Implicate Loop*) > $\overline{\text{Xi}}$'s (second largest *Non-Implicate Loop*) to implement the pass transistor logic switch circuit.

Case(c). The largest *Implicate Loop* includes fewer amounts of un-chosen squares than second largest *Non-Implicate Loops*: For implementation of the pass transistor logic switch circuit, we choose the second largest one.

Step 4. Repeat the similar process in **Step 2~3** to circle the rest second largest *Implicate Loop* and third largest *Non-Implicate Loop* circling ... until all squares in the *Modified K-map* are selected.

Step 5. Implement the new circuit according to the concept of the *Circuit Implementing Methods* and share transistor to reduce the number of transistors.

3.2 An Implementation Example (2-Input XOR)

Step 1: The *Modified K-map* of the function already shown in Fig. 5(a). The *Selected Set* is { 0, 1, A, B } in this step.

Step 2 *(case(b)):* Four different largest loops which has square amounts of two are found in Fig. 5(a). According to the *Selected Set*, the loop(a) and loop(b) have higher priority than the loop(c) and loop(d), so the loop(a) is chosen first as Loop(i) for circuit implementation. Next, the loop(b) and loop(d) will be skipped due to they are *Implicate Loops*. The loop(c) will also be skipped, cause the pass variable “$\overline{A}$ ’” is not existed in the *Selected Set*. The Selected Set is {0, 1, A, B, $\overline{B}$ } after Loop(i) is circled.

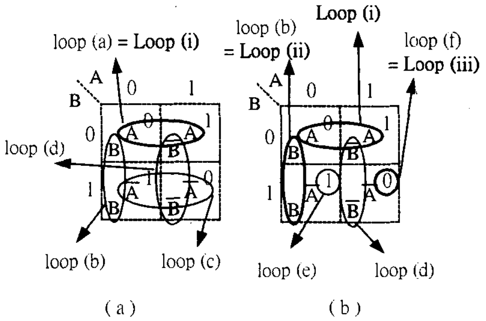

Fig. 5. (a), (b) The circling procedures of the 2-input XOR Modified K-map.

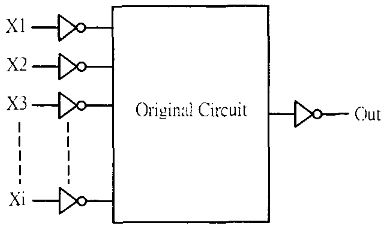

Fig. 6. Circuit arrangement for the simulation.

Step 3 *(case(b)):* The loop(b) and loop(e) both have one un-chosen *Square*, so the loop(b) with higher priority is chose as Loop(ii) to form the circuit. Similarly, the loop(f) will be chose as Loop(iii) due to the higher priority than the loop(d) as shown in Fig. 5(b). The loop(c) will always be skipped, cause the pass variable “$\overline{A}$ ’” is not existed in the Selected Set. The *Selected Set* is {0, 1, A, B, $\overline{B}$ } after Loop(ii) and Loop(iii) are circled.

Step 4: Implement the 2-input XOR gate circuit according to the selected loops. The circuit composition of each selected Loop is shown in Fig. 4.

4. COMPARISONS

Comparisons of the DVL, DPL, CMOS and new logic family through 3 different logic functions are listed in Table.1 ~ 3. The comparisons are based on 0.35μm CMOS technology and post layout simulation for supply voltage at 1.5V. Possible transition combinations are simulated, and the time taken of the worst-case signal transition from input (50% level) to output (50% level) worst-case gate delay is applied as delay value. Power-delay product is calculated as a quality measure for power efficiency.

The measure circuit for delay-time and average power dissipation simulation is shown in Fig.6. Due to the pass transistor circuits using the passive MOS switches to implement a given logic function, in order to measure the average power dissipation of the original circuit, some inverters are added in front of the input of the original

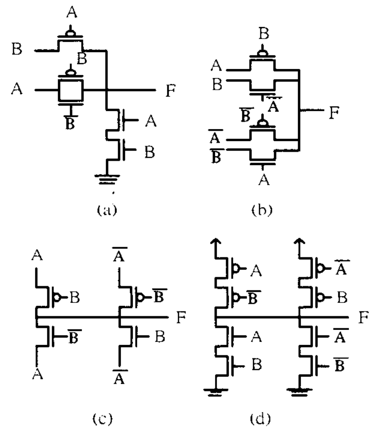

Fig.7. The circuit of 2-input XOR function in
(a) The New Logic Style (b) The DPL structure
(c) The DVL structure (d) The Static CMOS structure.

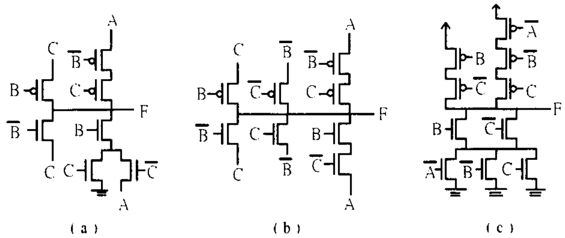

Fig. 8. The F = $\overline{B}C + AB\overline{C}$ function logic gate in
(a) New Logic Style structure. (b) The DVL structure.
(c) The Static CMOS structure.

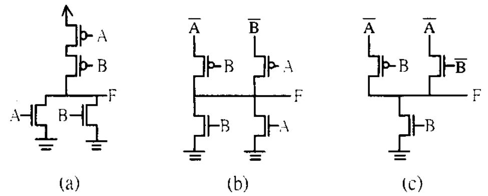

Fig.9. The circuits of NOR function in
(a) The New Logic Style. (b) The DPL structure.
(c) The DVL structure.

circuits. The inverter after the output of the original circuit is used as a unit fan out loading and to check the output waveform and delay-time.

The compared circuits include a simple NOR function, a special 2-input XOR function and a complex combination function for various requirements. For a special 2-input XOR function in Fig.7, the new circuit shown in Fig.7(a) and also proven in literature[4], has advantages over DVL, DPL and static CMOS logic families in power, power-delay product and area. For a complex combination function [3], the circuits are shown in Fig.8, the new circuit based upon the proposed procedures as shown in Table.2. It has the best performance in all aspects. For a simple 2-input NOR function as shown in Fig.9, the proposed new logic design procedure produces exactly the same circuit with static CMOS structure which has the best performance for NOR function.

5. CONCLUSIONS

In this paper, a new logic circuit synthesis and optimization procedure for arbitrary logic function is proposed. The new proposed logic family proves to be superior to DVL, DPL and CMOS in all aspects with only a few exceptions. The advantages of the propose logic family are low power consumption, low power-delay product and area efficiency. It's robustness against transistor downsizing and voltage scaling makes it good for deep sub-micron VLSI usage.

6. REFFERENCE

[1] V. G. Oklobdzija, B. Duchene, "Development and Synthesis Method for Pass-Transistor Logic Family for High-Speed and Low Power CMOS" *Proceedings of the 38th Midwest Symposium* on Volume: 1, 1996, Page(s): 298-301 VOL.1.

[2] M. Suzuki, N. Ohkubo, T. Shinbo, T. Yamanaka et. al."A1.5 32-b CMOS ALU in Double PASS-Transistor Logic," *IEEE J. Solid-State Circuits*, vol. 28, no. 11, pp. 1145-1151, November 1993.

[3] G. Oklobdzija, B. Duchene, "Pass-Transistor Dual Value Logic For Low–Power CMOS" *Proceedings of the 1995 International Symposium on VLSI Technology*, Taipei, Taiwan, 1995

[4] Hanho Lee and Gerald E. Sobelman "New Low-Voltage Circuit for XOR and XNOR", *Southeastcon '97 Engineering New Century, Proceeding IEEE*, 1997, Page(s):225~229.

[5] J.Wang,S.Fang and W.Feng, "New Efficient Designs for XOR and XNOR Function on the Transistor Level", *IEEE Journal of Solid-State Circuit*, 29(7):780-786, July 1995.

Table.1. Various circuits comparison results of the XOR-2 function

	Delay-time (ns)	Power (μW)	Normalize power-delay product	Global size of transistors.
Fig.7a	0.446	7.946	1.00	11
Fig.7b	0.366	11.35	1.17	16
Fig.7c	0.432	10.55	1.79	13
Fig.7d	0.643	15.05	2.72	24

Table.2. Various circuits comparison results of F=$\overline{B}C + AB\overline{C}$ function.

	Delay-time (ns)	Power (μW)	Normalize power-delay product	Global size of transistors.
Fig.8a	0.458	8.389	1.00	15
Fig.8b	0.509	11.94	1.58	23
Fig.8c	0.852	12.95	2.86	28

Table.3. Various circuits comparison results of the NOR function

	Delay-time (ns)	Power (μW)	Normalize power-delay product	Global size of transistors.
Fig.9a	0.364	7.040	1.00	8
Fig.9b	0.346	12.31	1.66	16
Fig.9c	0.359	9.593	1.34	11.5