A New Description of CMOS Circuits at Switch-Level*

Massoud Pedram

Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, CA 90089-2562, U.S.A.
Tel: (213)740-4458
Fax: (213)740-7290
email: massoud@zugros.usc.edu

Xunwei Wu**

Department of Electronic Engineering
Hangzhou University
Hangzhou, Zhejiang 310028, CHINA
Tel: +86-571-8071224
Fax: +86-571-8070107
email: xwu@ms.fudan.sh.cn

Abstract— After analyzing the limitations of the traditional description of CMOS circuits at the gate level, this paper introduces the notions of switching and signal variables for describing the switching states of MOS transistors and signals in CMOS circuits, respectively. Two connection operations for describing the interaction between MOS transistors and signals and a new description for CMOS circuits at the switch level are presented. This new description can be used to express the functional relationship between inputs and the output at the switch level. It can also be used to describe the circuit structure composed of various transistor switches. Based on the new description, the design of CMOS circuits at switch level can be efficiently realized. It is expected that this will provide a basis for techniques for analyzing and optimizing delay and power dissipation of CMOS circuits.

I. Traditional Description of CMOS Circuits at Gate Level

The traditional description of CMOS circuits is based on Boolean algebra. Its elementary points are:

(1) Boolean variables are used to represent signals in circuits. The two values of a variable, 1 and 0, are physically represented by two levels of a signal, for example 5V and 0V.

(2) The basic operations among variables in Boolean algebra are NOT, AND and OR operations. Usually, two composite operations, NAND and NOR, are also introduced. These operations are realized by the corresponding basic circuit units called gates, such as NOT gate (inverter), AND gate, OR gate, NAND gate and NOR gate.

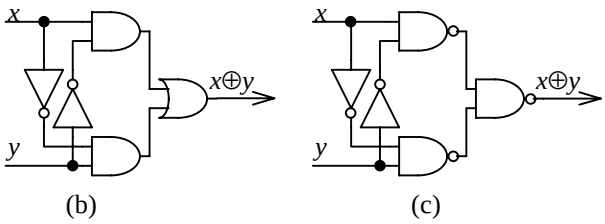
(3) NOT, AND and OR operations form a complete set and can be used to express any functions. Besides, NAND alone (or NOR alone) can form a complete basis by itself. Therefore, as long as we get the function expression we can obtain its corresponding circuit configuration by using gates from the complete set. For example, the Exclusive-OR function is given by its truth table shown in Fig.1(a). Based on the Boolean algebra, we can describe the Exclusive-OR function of x and y by the following expression:

$$x \oplus y = (x \cap \bar{y}) \cup (\bar{x} \cap y) = \overline{\overline{(x \cap \bar{y})} \cap \overline{(\bar{x} \cap y)}},$$

which yields the corresponding circuits in Fig. 1(b) and (c), respectively. This example explains how the description is used for both the function and the circuit structure at the gate level.

(a)

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0



(b) (c)

Fig. 1 Definition and circuits for Exclusive-OR operation (a) truth table, (b) circuit composed of NOT, AND and OR gates, (c) circuit composed of NOT and NAND gates

We should also point out the following limitations of the traditional description.

(1) The internal structure of a gate cannot be described, or derived from the function expression. For example, the internal structures of CMOS inverter and CMOS NAND gate shown in Fig. 2 (a) and (b) cannot be described by Boolean algebra.

* This work was supported in part by DARPA under contract #F33615-95-C-1627 and NNSF of CHINA.
** Xunwei Wu is presently on leave at Univ. of Southern California..

ASP-DAC '97
0-89791-851-7$5.00

(2) A compound gate which also realizes the Exclusive-OR function is shown in Fig. 2 (c). The structure cannot be described by Boolean algebra because it is not composed of gates. Reference [1] introduces the following procedure to derive the structure from its Boolean expression. First, we obtain the inverted expression: $x \oplus y = \overline{(x \cap y) \cup (\bar{x} \cap \bar{y})}$. For the n-side of the CMOS structure, we take the non-inverted expression $(x \cap y) \cup (\bar{x} \cap \bar{y})$. Here the operations $\cap$ and $\cup$ may be considered connections of nMOS transistors in series and in parallel. After having the n-structure, a dual p-structure can be derived and the whole configuration is obtained as shown in Fig. 2 (c). Obviously, the procedure is not included in Boolean algebra.

(3) NOT, AND and OR operations in Boolean algebra form a complete set, but they cannot be used to describe the relationship of the output to the inputs of a CMOS circuit that has a high-impedance state Φ. The output of a simple CMOS transmission gate shown in Fig. 2 (d) is expressed as $f = (c \cap x) \cup (\Phi \cap \bar{x})$ in some textbooks. However, we do not define the operations related to the high-impedance state F in Boolean algebra.

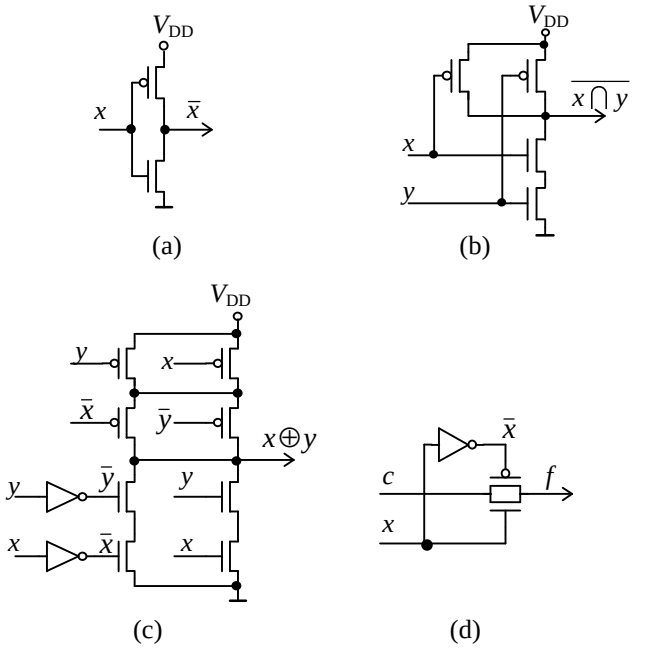


Fig .2. Switch structure of some CMOS gates (a) inverter, (b) NAND gate, (c) composed Exclusive-OR gate, (d) transmission gate

According to the above discussion, Boolean algebra can be used to describe the CMOS circuit structure at the gate level, but cannot be used to describe the switching states of MOS transistors in the circuit and the circuit structure at switch level. The description at switch level is however desired for the switch-level techniques [2,3]. This paper proposes a new description that describes both signal and switching state of transistors in CMOS circuits and reflects the circuit structure at the switch level.

II. Description of CMOS Circuits at Switch Level

In order to describe the CMOS circuit structure with transistors, we introduce an additional variable that describes switches in circuits; we should distinguish the new variable from the variable which is used to describe signals.

(1) Assume α, β … are switching variables that take two values, T and F, which in turn represent the two opposite states of on and off for a MOS transistor. The basic operations related to switching variables are NOT, AND and OR. Their definitions are as follows:

NOT operation

$$\tilde{\alpha} = \begin{cases} \text{T} & \text{if} \quad \alpha = \text{F}, \\ \text{F} & \text{if} \quad \alpha = \text{T}; \end{cases} \tag{1}$$

AND operation

$$\alpha \cdot \beta = \begin{cases} \text{T} & \text{if} \quad \alpha = \beta = \text{T} \\ \text{F} & \text{otherwise} \end{cases} \tag{2}$$

OR operation

$$\alpha + \beta = \begin{cases} \text{F} & \text{if} \quad \alpha = \beta = \text{F} \\ \text{T} & \text{otherwise} \end{cases} \tag{3}$$

Based on the above basic operations, a binary *switching algebra* is established.

(2) Assume x, y, … are binary signal variables. They take two values, 1 and 0, which represent the two signal levels, high and low, in a circuit. They have a precise magnitude and can be identified by comparing their magnitude with a threshold value, denoted by 0.5. The basic operations related to binary signal variables are Complement, Minimum and Maximum. Their definitions are as follows:

Complement operation

$$\bar{x} = \begin{cases} 1 & \text{if} \quad x = 0 \\ 0 & \text{if} \quad x = 1 \end{cases} \tag{4}$$

Minimum operation

$$x \cap y = \begin{cases} 1 & \text{if} \quad x = y = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Maximum operation

$$x \cup y = \begin{cases} 0 & \text{if} \quad x = y = 0 \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

Based on the above basic operations, a binary *signal algebra* is formed.

Therefore, we have two kinds of binary algebra systems; it can be shown that the two systems are isomorphic. However,

in the existing literature they are always substituted for one another confused without considering their essential differences and their isomorphism. Therefore, gates that realize Complement, Minimum, and Maximum operations are traditionally named NOT, AND and OR gates.

Taking inverter in Fig. 2(a) as an example to explain two kinds of variables, we can use x, $\bar{x}$, α_p and α_n to express the input and output signal, and the switching states of pMOS and nMOS transistors, respectively. Their relationship is given in Table 1.

TABLE 1

RELATIONSHIP BETWEEN SIGNALS AND SWITCHING STATES IN A CMOS INVERTER

x	α_p	α_n	$\bar{x}$
0 (low level)	T (on)	F (off)	1 (high level)
1 (high level)	F (off)	T (on)	0 (low level)

We can further introduce operations between two kinds of variables for describing the connection between the on-off states of switching elements and the voltage levels of the signals, as shown in Fig. 3. They are:

Connection operation I -- describing the physical process of how the binary signal controls the on-off state of an element.

Connection operation II -- describing the physical process of how the on-off state of an element controls the transmission of the binary signal.

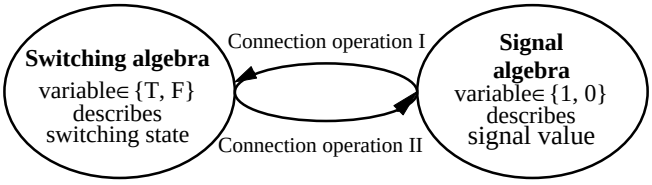


Fig. 3. Connections between binary switching variables and binary signal variables

In a CMOS digital circuit, the on-off state of a MOS transistor is dependent on the comparison between the gate signal and the threshold. Therefore, we can define the connection operation I as follows:

High-threshold comparison operation

$$^{0.5}x = \begin{cases} \mathrm{T} & \text{if} \quad x > 0.5 \\ \mathrm{F} & \text{if} \quad x < 0.5 \end{cases} \tag{7}$$

Low-threshold comparison operation

$$x^{0.5} = \begin{cases} \mathrm{T} & \text{if} \quad x < 0.5 \\ \mathrm{F} & \text{if} \quad x > 0.5 \end{cases} \tag{8}$$

In Eqs.(7) and (8), 0.5 implies that the detection threshold is set in the middle of two logic levels, 1 and 0. These two equations represent the low-active switching characteristic of a pMOS transistor and the high-active switching characteristic of an nMOS transistor, respectively.

The following properties can be easily verified by use of the above definitions:

$$\bar{x}^{0.5} = {}^{0.5}x = x^{0.5} \sim , \tag{9}$$

$${}^{0.5}\bar{x} = x^{0.5} = {}^{0.5}x \sim , \tag{10}$$

which state that the two threshold comparison operations can be transformed by complementing the signal variable.

$${}^{0.5}(x \cap y) = {}^{0.5}x \cdot {}^{0.5}y , \tag{11}$$

$${}^{0.5}(x \cup y) = {}^{0.5}x + {}^{0.5}y . \tag{12}$$

In Eqs.(10)-(12) the corresponding relationships between Complement, Minimum and Maximum in binary signal algebra and NOT, AND and OR in switching algebra are established by use of the high-threshold comparison operations. Furthermore, the high-threshold comparison operation penetrates through a function $f(x, y, \cdots; -, \cap, \cup)$ in binary signal algebra as follows:

$${}^{0.5}f(x, y, \cdots; -, \cap, \cup) = f({}^{0.5}x, {}^{0.5}y, \cdots; \sim, \cdot, +) . \tag{13}$$

On the other hand, the on-off state of a MOS transistor determines whether the source signal is transmitted to the drain or not. Therefore, we may introduce the connection operation II as follows.

Transmission operation

$$c * \alpha = \begin{cases} \mathrm{c} & \text{if} \quad \alpha = \mathrm{T} \\ \Phi & \text{if} \quad \alpha = \mathrm{F} \end{cases} \tag{14}$$

where the binary variable c is called transmitted source signal, and a represents the switching state of a transmission switch network. If α = T, signal c is transmitted to the output; if α = F, the switch network is off and its output is in the high-impedance state, denoted by symbol Φ. The switch network is composed of an nMOS transistor, if c = 0, or a pMOS transistor, if c = 1.

To denote the joining of the outputs of two (or more) transmission branches, we define the following operation further.

Union Operation

$$c_1 * \alpha_1 \# c_2 * \alpha_2 = \begin{cases} c_1 * \alpha_1 & \text{if} \quad c_2 * \alpha_2 = \Phi \\ c_2 * \alpha_2 & \text{if} \quad c_1 * \alpha_1 = \Phi \end{cases} \tag{15}$$

In Eq.(15), the transmission operation * takes priority over the Union operation #. Note that 1 # Φ = 1 and 0 # Φ = 0. Furthermore, if $c_1 \neq c_2$ and $\alpha_1 = \alpha_2$ = T, a voltage conflict

arises between sources c_1 and c_2; this condition is not allowed. As an example, we can use the above operations to re-express $\bar{x}$ in Eq.(4) at switch level:

$$\bar{x} = 1 * x^{0.5} \# 0 *^{0.5} x \,. \tag{16}$$

The above expression exactly describes the circuit structure in Fig.2(a) and its working process shown in Table 1.

It can be proved that the following laws related to the transmission operation and union operation hold.

Serial transmission law

$$(c*\alpha_1)*\alpha_2 = c*(\alpha_1 \cdot \alpha_2)\,, \tag{17}$$

Parallel transmission law

$$c*\alpha_1 \# c*\alpha_2 = c*(\alpha_1 + \alpha_2)\,, \tag{18}$$

Commutation law

$$c_1*\alpha_1 \# c_2*\alpha_2 = c_2*\alpha_2 \# c_1*\alpha_1\,, \tag{19}$$

Associative law

$$\begin{aligned}&(c_1*\alpha_1 \# c_2*\alpha_2)\#c_3*\alpha_3 = c_1*\alpha_1 \#(c_2*\alpha_2 \# c_3*\alpha_3)\\ &= c_1*\alpha_1 \# c_2*\alpha_2 \# c_3*\alpha_3\end{aligned} \tag{20}$$

Distributive law

$$(c_1*\alpha_1 \# c_2*\alpha_2)*\alpha_3 = c_1*(\alpha_1 \cdot \alpha_3) \# c_2*(\alpha_2 \cdot \alpha_3)\,, \tag{21}$$

We can use the connection operations to derive a new canonical function form. For example, a two-variable function $f(x,y)$ has the following canonical expansion form at the switch level:

$$\begin{aligned}f(x,y) = {}& f(0,0)*(x^{0.5} \cdot y^{0.5}) \# f(0,1)*(x^{0.5} \cdot^{0.5} y)\\ &\# f(1,0)*({}^{0.5}x \cdot y^{0.5}) \# f(1,1)*({}^{0.5}x \cdot^{0.5} y)\end{aligned} \tag{22}$$

In comparison, the two-variable function $f(x,y)$ has its traditional min-term expansion at the gate level:

$$\begin{aligned}f(x,y) = {}& [f(0,0) \cap \bar{x} \cap \bar{y}] \cup [f(0,1) \cap \bar{x} \cap y]\\ &\cup [f(1,0) \cap x \cap \bar{y}] \cup [f(1,1) \cap x \cap y]\end{aligned} \tag{23}$$

Equation (23) shows how the circuit is realized by using gates, which could be renamed Complement gate (inverter), Minimum gate (AND gate) and Maximum gate (OR gate). However, Eq.(22) explains how four signals, $f(i,j)$, are transmitted to the output through two switches in series. These equations illustrate the difference in philosophy between switch-level and gate-level descriptions.

III. Applications of The Description at Switch Level

Since in Eq.(22) the expansion coefficient $f(i,j) \in \{0,1\}$, we can factor coefficients 1 and 0, respectively, and obtain the following form:

$$f = 1 *^{0.5} f \# 0 *^{0.5} f^{0.5}\,, \tag{24}$$

where ${}^{0.5}f$ and $f^{0.5}$ are complementary. They are the switching functions of source 1 and source 0, respectively. By using Eq.(9) the above equation can be rewritten as

$$f = 1 *^{0.5} f \# 0 *^{0.5} \bar{f}\,. \tag{25}$$

If we have the simplified function expression $\overline{f(x,y,\cdots;-,\cap,\cup)}$ in traditional binary signal algebra, we can easily derived its corresponding switch-level expression ${}^{0.5}\overline{f(x,y,\cdots;-,\cap,\cup)}$ by using Eq.(13). The latter shows how to use serial and parallel nMOS switch connections for controlling the transmission of source 0. According to ${}^{0.5}\bar{f} = {}^{0.5}f$ in Eq.(25) and De Morgan's Law, the two switch-level expressions, ${}^{0.5}f$ and ${}^{0.5}\bar{f}$, are dual. And that is the principle of the design procedure presented in [1] .

Taking $f_1 = \overline{x \cap y}$ and $f_2 = x \oplus y$ as examples, we have $\bar{f}_1 = x \cap y$ and $\bar{f}_2 = (x \cap y) \cup (\bar{x} \cap \bar{y})$. Then the following switch-level expressions can be obtained:

$${}^{0.5}\bar{f}_1 = {}^{0.5}x \cdot^{0.5} y \quad \text{and} \quad {}^{0.5}\bar{f}_2 = {}^{0.5}x \cdot^{0.5} y + {}^{0.5}\bar{x} \cdot^{0.5} \bar{y}\,.$$

The above two expressions describe the n-branches in Fig.2 (b) and (c). By De Morgan's Law, we have the following dual expressions for describing the corresponding p-branches:

$${}^{0.5}f_1 = x^{0.5} + y^{0.5} \quad \text{and} \quad {}^{0.5}f_2 = (x^{0.5} + y^{0.5}) \cdot (\bar{x}^{0.5} + \bar{y}^{0.5})\,.$$

In fact, the duality between p-part and n-part in a CMOS circuit is unnecessary. Because $f_2 = x \oplus y$ also can be expressed as $f_2 = (x \cap \bar{y}) \cup (\bar{x} \cap y)$, we have the following expression instead:

$${}^{0.5}f_2 = {}^{0.5}x \cdot^{0.5} \bar{y} + {}^{0.5}\bar{x} \cdot^{0.5} y = \bar{x}^{0.5} \cdot y^{0.5} + x^{0.5} \cdot \bar{y}^{0.5}\,.$$

The above expression will guide a new pMOS connection model, which would eliminate the internal connection in the p-branch in Fig. 2(c).

Another example is given in design of a combinational adder as follows. For a circuit with output inverting buffers, we can design a circuit with inverse outputs, $\overline{C_+}$ and $\bar{S}$, first. From Eq.(25) we have

$$\begin{aligned}\overline{C_+} &= 1 *^{0.5} \overline{C_+} \# 0 *^{0.5} C_+\,,\\ \bar{S} &= 1 *^{0.5} \bar{S} \# 0 *^{0.5} S\,.\end{aligned}$$

Based on the traditional Boolean algebra the following expressions are derived:

$$\begin{aligned}C_+ &= (A \cap B) \cup [(A \cup B) \cap C]\,,\\ \overline{C_+} &= (\bar{A} \cap \bar{B}) \cup [(\bar{A} \cup \bar{B}) \cap \bar{C}]\,,\\ S &= [(A \cup B \cup C) \cap \overline{C_+}] \cup (A \cap B \cap C)\,,\\ \bar{S} &= [(\bar{A} \cup \bar{B} \cup \bar{C}) \cap C_+] \cup (\bar{A} \cap \bar{B} \cap \bar{C})\,.\end{aligned}$$

Note that in the above expressions C_+ and $\overline{C_+}$, S and $\overline{S}$ are symmetric rather than dual with each other. By using Eq.(13), their corresponding expressions at switch-level are:

$$^{0.5}\overline{C_+} = A^{0.5}\cdot B^{0.5} + (A^{0.5} + B^{0.5})\cdot C^{0.5}$$

$$^{0.5}C_+ = {}^{0.5}A\cdot{}^{0.5}B + ({}^{0.5}A + {}^{0.5}B)\cdot{}^{0.5}C$$

$$^{0.5}\overline{S} = (A^{0.5} + B^{0.5} + C^{0.5})\cdot \overline{C_+}^{0.5} + A^{0.5}\cdot B^{0.5}\cdot C^{0.5}$$

$$^{0.5}S = ({}^{0.5}A + {}^{0.5}B + {}^{0.5}C)\cdot{}^{0.5}\overline{C_+} + {}^{0.5}A\cdot{}^{0.5}B\cdot{}^{0.5}C$$

Therefore we obtain the corresponding circuit design at switch level, as shown in Fig. 4 [1]. Obviously, the p-branch and n-branch in the optimized schematic are symmetric.

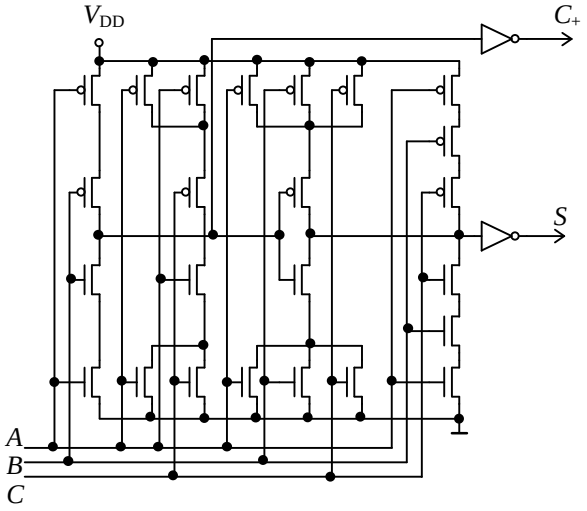


Fig. 4. Circuit design of a combinational 1-bit full adder

In addition to transformation from the traditional Boolean expression, the switch-level expression can be also derived directly from truth table or Karnaugh map. Figure 5 shows three Karnaugh maps for outputs of the NAND gate, the transmission gate, and the Exclusive-OR gate. From the mapping synthesis shown in Fig.5(a), we directly obtain the switch-level expression:

$$\overline{x\cap y} = 1*(x^{0.5} + y^{0.5})\#0*({}^{0.5}x\cdot{}^{0.5}y)\,.$$

The above expression describes how the source 1 (V_{DD}) transmitted through two p-transistors in parallel to the output, and the source 0 (Ground) transmitted through two n-transistors in series to the output, as shown in Fig. 5(a).

According to the mapping in Fig. 5(b), we use the variable c as its transmitted source and obtain the switch-level expression:

$$f = c*{}^{0.5}x\,.$$

The above expression shows that the source c is transmitted by an n-transistor. To avoid the poor transmission for $c = 1$ we rewrite the expression as:

$$f = c*({}^{0.5}x + \bar{x}^{0.5})\,,$$

where two terms, ${}^{0.5}x$ and $\bar{x}^{0.5}$, are equal based on Eq. (9). However, they describe that the variable source are transmitted by a complementary MOS construction, as shown in Fig. 5(b).

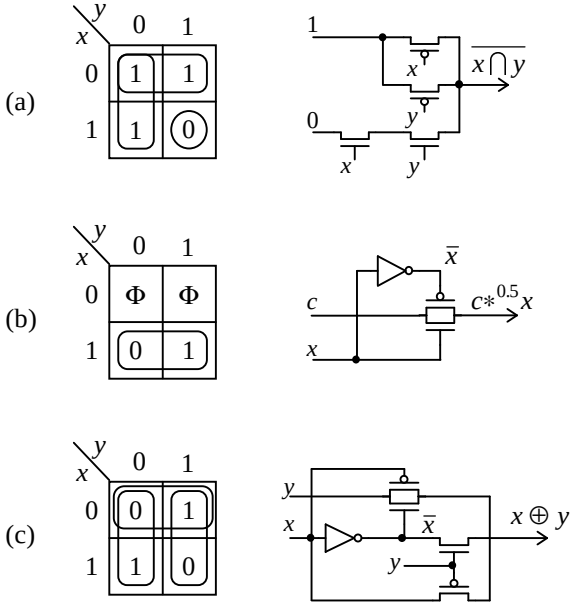


Fig. 5. Mapping synthesis of some CMOS circuits at switch-level (a) NAND gate, (b) Transmission gate, (c) Exclusive-OR gate

Based on the previous example, we can synthesize Exclusive-OR in its Karnaugh map, as shown in Fig. 5(c), and get

$$x \oplus y = y*(x^{0.5} + {}^{0.5}\bar{x})\#x(1)*y^{0.5}\#\bar{x}(0)*{}^{0.5}y\,.$$

Notice that the parts overlapped in Karnaugh map have been realized by $y*(x^{0.5} + {}^{0.5}\bar{x})$, and therefore the branch with source x (term $x(1)*y^{0.5}$) only has to transmit a 1 and never a 0, and the branch with source $\bar{x}$ (term $\bar{x}(0)*{}^{0.5}y$) only has to transmit a 0 and never a 1. Hence each of these transmission branches can be realized with a single MOS transistor as shown in Fig. 5(c). This simple circuit realization previously has been considered as something that "… does not follow from any systematic (design) method" [4].

Comparing all three designs of Exclusive-OR gate in Fig. 1(c), Fig. 2(c) and Fig. 5(c) we find that the number of transistors in the circuits are 16, 12 and 6, respectively. Besides, the numbers of internal nodes in circuits are 5, 3 and 2, respectively. This means that the design based on switch-level description may lead to a circuit with a simpler structure as well as higher quality (delay and power). A schematic diagram of a transmission gate adder, which has two output buffers and equal delay for the sum (S) and carry-out (C_+), and is designed from our switch-level specification directly, is shown in Fig. 6. Only 20 MOS transistors are used in the

circuit. In comparison, a similar design given in [1] needs 26 transistors, and the standard design in Fig. 4 needs 28 transistors..

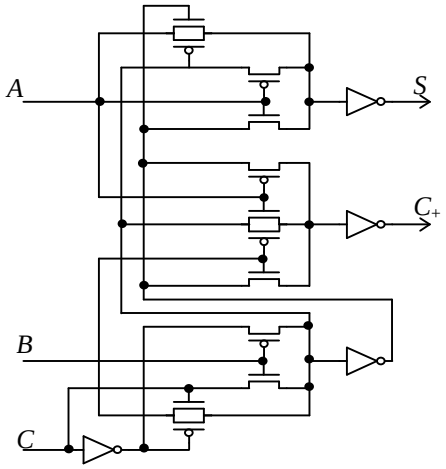


Fig. 6. Transmission gate 1-bit full adder designed at switch level

It should be pointed out that the description of CMOS circuits at switch-level also offers a new method for analyzing and optimizing circuits. However, since there exist physical capacitance and resistance for MOS transistors, the permutation of the inputs to a series chain of transistors will lead to different input pin loads and pin dependent delays. It is well known that the signal to pin assignment in a CMOS logic gate has a sizable impact on the propagation delay through the gate [5]. Besides, researchers of power optimization also indicate that the assignments of input signal with different probability of assuming a controlling value (zero for nMOS and one for pMOS), or input signal with different switching activity when all other inputs are set to their non-controlling values (one for nMOS and zero for pMOS in series-connected transistors) must be considered for power reduction [6,7]. Therefore, we have to describe the circuit more accurately at switch level. It means that each transistor in series-connected structure should be located exactly. In fact, by weakening the communication law of ANDed switching variables, those two switching variables in Eq. (17) cannot be permuted. Thus, the order of ANDed switching variables will represent corresponding location accurately in the series-connected structure. Taking the Exclusive-OR gate shown in Fig. 2(c) as the example, its output can be expressed as

$$x \oplus y = 1*[(x^{0.5} + y^{0.5}) \cdot (\bar{x}^{0.5} + \bar{y}^{0.5})]\#0*[^{0.5}x \cdot^{0.5} y +^{0.5}\bar{x} \cdot^{0.5}\bar{y}].$$

The above expression indicates that the two nMOS transistors which are controlled by signals x and $\bar{x}$ are close to Ground since their corresponding switching variables are near the source 0 in the above expression. Obviously, the other two nMOS transistors are close to the output terminal. Therefore, the proposed description of CMOS circuits is modified to locate each transistor in the circuit. It is expected that the new switch-level description will provide a basis for analyzing and optimizing delay and power dissipation by using the new description.

IV. Conclusion

The traditional description of CMOS circuits is based on Boolean algebra, where three basic operations, NOT, AND and OR, are used to describe functional relationship between inputs and the output, and to describe the circuit structure composed of gates. However, it cannot be used to describe the internal structure of MOS transistor switches. Besides, there exist some problems with Boolean algebra when describing complex gates or gates with high-impedance state. This paper introduced another variable to describe the switching state of transistors in addition to original variable, which describes signal in the circuit. The two variables have their own independent operations. Since there exists a mutual relationship between the on-off states of switch elements and the signals, we proposed two connection operations for describing their interaction, whereby a new description for CMOS circuits at the switch level is presented. Based on the new description the design of CMOS circuits at switch level can be realized. For CMOS circuits the traditional inverting-logic stage design and pass-transistor design have been considered to be two different design methods [8,9]. However, the new description proposed in this paper can unite the two and can overcome other difficulties in the traditional theory. Besides, it is expected that it can provide a basis for techniques for analyzing and optimizing delay and power dissipation.

References

[1] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, 2nd Edition, Addison-Wesly Publishing Company, New York, Ch.1, pp.15-16; Ch.8, p.525, 1993.

[2] J. P. Shen and S. Hirschhorn, "Switch-level techniques," *IEEE Design & Test of Computers*, Vol.4, No.4, pp.15-16, 1987.

[3] P. Hares, "An introduction to switch-level modeling," *IEEE Design & Test of Computers*, Vol.4, No.4, pp.18-25, 1987.

[4] A. Mukherjee, *Introduction to nMOS and CMOS VLSI System Design*, Englewood Cliffs, NJ: Prentice-Hall, 1986.

[5] S. M. Kangand and Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*, McGraw-Hill Companies, Inc. 1996.

[6] M. Pedram, "Power estimation and optimization at the logic level," *Int. J. of High speed Electronics and Systems*, Vol.5, pp.179-202, 1994.

[7] S. C. Prasad and K. Roy, "Circuit optimization of power consumption under delay constraint," in *Proceedings of the 1994 International Workshop on Low Power Design*, pp.15-20, April 1994.

[8] C. Mead and L. Conway, *Introduction to VLSI Systems*, Addison-Wesley, Reading, MA, 1980.

[9] D. Radhakrishnan, S. R. Whitaker and G. K. Maki, "Formal design procedures for pass transistor switching circuits," *IEEE J. Solid-state Circuits*, Vol.SC-20, pp.531-536, 1985.