

Implementation of Ternary Circuits with Binary Integrated Circuits

DANIEL ETIEMBLE AND MICHEL ISRAEL

Abstract—A general method for implementing ternary circuits with binary integrated circuits is described. The conditions are given for a particular technology to be able to implement ternary circuits. For TTL, COSMOS, and ECL technologies, the necessary circuits are developed. Formulas are provided to obtain the minimum circuit according to the particular function required. Simplification rules corresponding to each technology are given. Some examples illustrate the method. Some dynamical characteristics are shown. Extensions of these results to other technologies and to n -valued circuits are possible.

Index Terms—Circuit realization of multivalued functions, combinational circuits, COSMOS integrated circuits, ECL integrated circuits, fundamental ternary circuits, multivalued circuits, ternary logic implementation, TTL integrated circuits.

INTRODUCTION

FOR several years, different papers have proposed ternary circuits implemented with binary integrated circuits or compatible with existing integrated circuit technologies [1]–[4].

For each technology, the authors have presented gate circuits able to implement the basic set of operations for a complete algebra and the expansion theorem to implement any arbitrary 3-valued function. But no general method has been given to implement ternary circuits for any technology.

In 1974, Birk and Farmer presented "An algebraic method for designing multivalued logic circuits using principally binary components" [5]. But they did not provide technological data to demonstrate how their method could be used in detail.

In this paper, we present a general method for implementing ternary circuits with IC binary components. In summary,

1) A method similar to Birk and Farmer's to realize the ternary to binary and binary to ternary conversions is presented.

2) The algorithms are given to implement any one of the 27 unary functions or any n -ary functions, with examples of TTL and COSMOS technologies provided. Thus, the conditions to implement multivalued logic circuits with typical IC binary circuits are defined.

Furthermore, the methods are provided for building basic circuits. Some dynamic characteristics are given of TTL and COSMOS technologies.

Manuscript received November 15, 1976; revised July 8, 1977.

D. Etiemble is with the Institut de Programmation, University of Paris, Paris, France.

M. Israel is with Conservatoire National des Arts et Metiers, Paris, France.

USING BINARY COMPONENTS

The difficulties met in implementing ternary logic circuits are due to technological factors. Ideally, physical phenomena selected must have three equiprobable stable states with the distances between each state and the other two being equal to give a uniform minimum switching time. The use of a cyclic physical phenomenon would allow one to obtain an optimum system. But in the field of electrical phenomena, quantified cyclic ones apparently do not exist. Furthermore, it is possible to prove that using nonquantified phenomena would make the circuits unserviceable [6]. They would present all the defects of linear circuits: noise transmission and additivity properties. The resulting circuits, adding and transmitting internal and external noises, would be thereby noniterative.

Thus, noncyclic physical phenomena must be used in some way. A good way is then to use, and combine in a special manner, binary integrated circuits.

The ternary logical circuits resulting must answer to the following requirements: speed, simplicity of circuits, minimum costs, integrability.

METHOD USED

We adopt the following notational conventions.

$E_3 = \{0, 1, 2\}$ set of ternary values.

$E_2 = \{0, 2\}$ set of binary values.

1) Let $\{E_3, +, \bullet\}$ be a distributive lattice, such that $+$, \bullet , be, respectively, the l.u.b. and g.l.b. among variables defined in E_3 , corresponding to the MAX and MIN logic operations.

2) Let $z = f(x, y)$ be a 3-valued function with $x, y, z \in E_3$.

3) Let $z_i = f_i(x, y)$ be a 2-valued function such that $x, y \in E_3$ and $z_i \in E_2$

$$z_i = f_i(x, y) = \begin{cases} 2, & \text{when } z = i, \\ 0, & \text{otherwise.} \end{cases}$$

4) Let x^i be a 2-valued function such that

$$x^i = \begin{cases} 2, & \text{when } x \geq i, \\ 0, & \text{otherwise.} \end{cases}$$

5) Let $z = f_{ijh}(x)$ be the unary 3-valued function such that

TABLE I

z	z ₂	z ₁	z ₂	z ₀	z ₁	z ₂	z ₁	z ₀	z ₀	z ₂	z ₀	z ₁
0	0	0	0	2	0	0	0	2	2	∅	2	∅
1	0	2	0	0	2	∅	2	∅	0	0	0	2
2	2	∅	2	∅	0	2	0	0	0	2	0	0

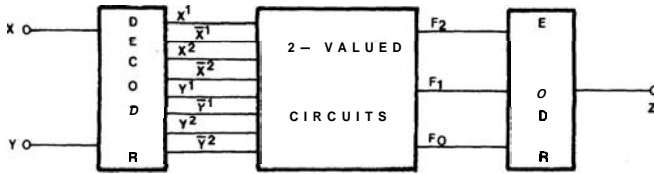


Fig. 1. Implementation of $z = f(x,y)$.

- $z = i,$ when $x = 0,$
- $z = j,$ when $x = 1,$
- $z = h,$ when $x = 2.$

6) Let ϕ be the notational term for a DONT CARE. The general expression of the z function is

$$z = 2 \cdot z_2 + 1 \cdot z_1 + 0 \cdot z_0.$$

But the z function is determined by two of the three z_i functions according to the relations defined in Table I. Each of the 6 pairs z_i, z_j corresponds to a particular ordered set $i > j > h$. We shall choose the particular pair z_i, z_j which allows us to minimize the number of circuits. With the DONT CARE's, the usual covering algorithms may be used to synthesize the z function. For example, if we want to realize the unary function f_{120} , the z_i, z_j functions are as follows.

7) With ordered set $0 < 1 < 2$

$$f_2 = x^1 \cdot \bar{x}^2$$

$$f_1 \equiv \bar{x}^1.$$

8) With ordered set $0 < 2 < 1$

$$f_1 = \bar{x}^1$$

$$f_2 = \bar{x}^2.$$

(1 may be considered as a DONT CARE.)

The implementation is easier when using the pair f_1, f_2 instead of the pair f_2, f_1 . The general scheme to realize the z function is given in Fig. 1.

The method used is similar to the method presented by Birk and Farmer [5]. The ternary to binary conversion is realized with threshold detectors (decoder circuit). For the sum (maximum) of product terms, Birk and Farmer use level restoring binary AND gates (at the next to final level) and finally a MAX gate at the output. We use a special circuit corresponding to the technology used (encoder circuit).

To implement ternary functions with a given technology, we must specify

1) the encoder circuit, which has 2-valued inputs and

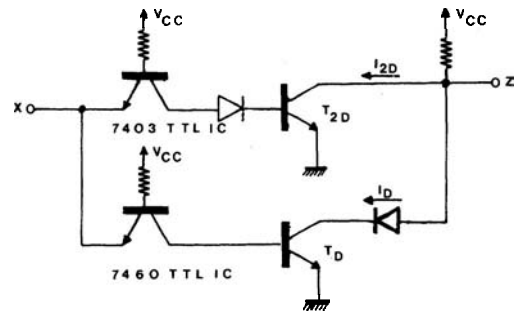


Fig. 2. Ternary TTL inverter.

TABLE II

x	X	i _D	Z _D	z _D
0	$X < V_D$	= 0	High	2
1	$X > V_D$	≠ 0	Low	0
2	$X > V_D$	≠ 0	Low	0

a 3-valued output. This circuit defines the 3 different output levels.

2) the decoder circuit, which has 3-valued inputs and a 2-valued output. This circuit realizes 2 separate threshold detectors. It implements the x^i functions.

TTL TECHNOLOGY

The authors presented in [3] a TTL ternary inverter. Another version is shown in Fig. 2. The TTL ternary inverter has one 7460 gate, one 7403 gate, one diode and one resistor.

The 7460 TTL IC [8] is a dual 4-input expander. When the output transistor emitter is grounded, this circuit is a dual 4-input positive NAND gates with open collector outputs. The multiemitter transistor implements the MIN function of the different inputs. The output transistor implements the logical negation. The expander has an input threshold of one diode drop. The gate's operation is shown in Table II, where X and Z are the physical input and output, x and z are the logical input and output, i_D is the collector current of the output transistor, and V_D is the switching threshold corresponding to one diode drop. The unused inputs are connected to the supply voltage. We use the symbol shown in Fig. 3 for the 7460 gate.

The 7403 TTL IC [8] is a quadruple 2-input positive NAND gate. It has a switching threshold of about two diode drops. The 7403 gate's operation is shown in Table III, where i_{2D} is the collector current of the output transistor, and V_{2D} is the switching threshold corresponding to two diode drops. Unused inputs are connected to supply voltage. We use the symbol shown in Fig. 4 for this gate.

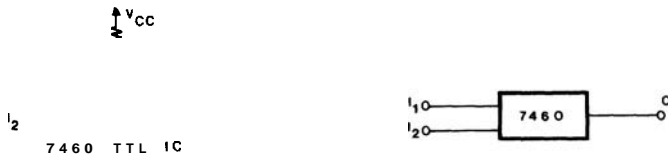


Fig. 3. 7460 TTL IC symbol.

The ternary inverter function is shown in Table IV, where the operating signal levels are given in the following.

$$\begin{aligned} \text{High} &= V_{CC} \\ \text{Middle} &= V_{CE}(\text{ON}) + V_D(\text{ON}) \\ \text{Low} &= V_{CE}(\text{ON}). \end{aligned}$$

The 3 levels are respectively 5 V, 0.9 V, 0.1 V, for $V_{CC} = 5$ v.

With the version presented in [3], the operating levels are 1.6 V, 0.9 V, and 0.1 V.

The TTL encoder and decoder use 7403 gates and 7460 gates. The encoder is shown in Fig. 5. Its operation is shown in Table V. We may use either 7403 or 7460 gates. Table VI summarizes the logical function of the encoder. Table I and Table VI allow us to determine the relations between a and b (encoder inputs) and the 6 pairs z_i, z_j as given in Table VII.

The decoder implements the \bar{x}^i functions. The \bar{x}^1 function is implemented by the 7460 gate (Fig. 6) and the \bar{x}^2 function is implemented by the 7403 gate (Fig. 6) with the unused inputs connected to the supply voltage.

The following relations are available with x^i functions.

$$\bar{x}^1(x^1) = \bar{x}^1(x) \tag{1}$$

$$\bar{x}^1(x^2) = \bar{x}^2(x) \tag{2}$$

$$\bar{x}^2(x^1) = \bar{x}^1(x) \tag{3}$$

$$\bar{x}^2(x^2) = \bar{x}^2(x). \tag{4}$$

These relations are demonstrated by Table VIII.

The MIN functions needed to synthesize the z_i functions may be realized either using the multiemitter inputs or by the well-known wired-AND capabilities of open collector gates. The 2-valued circuits used are either 7460 or 7403 gates with wired-AND capabilities or any TTL gate without wired-AND capabilities. Awareness of these remarks will allow us some simplification in implementations of ternary functions.

COSMOS TECHNOLOGY

Mouftah and Jordan presented in [4] the COSMOS ternary inverter which is shown in Fig. 7.

The operation of the ternary COSMOS inverter is given in Table IX, where V_p and V_n are the threshold levels for the p and n channel transistors. When $V_{SS} = -V_{DD}$, the operating levels are V_{DD} , 0, and $-V_{DD}$. This inverter is realized with one CD 4007 IC [9] and two resistors.

The COSMOS encoder is shown in Fig. 8. It uses the same IC with separated a and b inputs. Its physical oper-

TABLE III

x	X	i_{2D}	Z _{2D}	z_{2D}
0	$X < V_{2D}$	= 0	High	2
1	$X < V_{2D}$	= 0	High	2
2	$X > V_{2D}$	≠ 0	Low	0

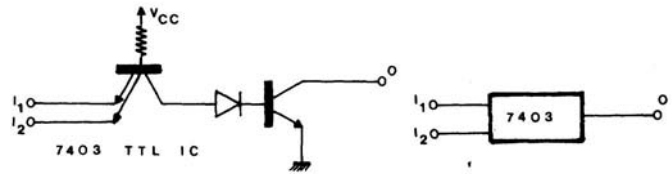


Fig. 4. 7403 TTL IC symbol.

TABLE IV

x	X	i_D	i_{2D}	Z	z
0	$X < V_D$	= 0	= 0	High	2
1	$V_D < X < V_{2D}$	≠ 0	= 0	Middle	1
2	$X > V_{2D}$	≠ 0	≠ 0	Low	0

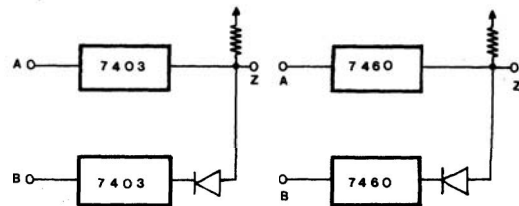


Fig. 5. Ternary TTL encoder.

TABLE V

a	b	i_D	i_{2D}	Z	z
0	0	= 0	= 0	H	2
0	2	= 0	≠ 0	M	1
2	0	≠ 0	= 0	L	0
2	2	≠ 0	≠ 0	L	0

TABLE VI

a	b	z
0	0	2
0	2	1
2	0	0

ation is shown in Table IX and its logical function is given in Table X. The ? notation indicates that the corresponding output is unspecified. The relation between the a and b inputs (Table XI) and the 6 pairs z_i, z_j are given in Table XII. They are different from the corresponding TTL ones. The COSMOS decoder is shown in Fig. 9.

To synthesize the z_i functions, we use MAX and MIN functions inherent in simple connections of MOS devices (Figs. 10 and 11). The input of a single p (or n) channel transistor produces the same output as the MAX (or MIN)

TABLE VII

	z_2 z_1	z_2 z_0	z_1 z_2	z_1 z_0	z_0 z_2	z_0 z_1
a	\bar{z}_2 \bar{z}_1	\bar{z}_2 z_0	\bar{z}_1 \bar{z}_2	\bar{z}_1 z_0	z_0	z_0
b	\bar{z}_2	\bar{z}_2	z_1	1	\bar{z}_2	z_1

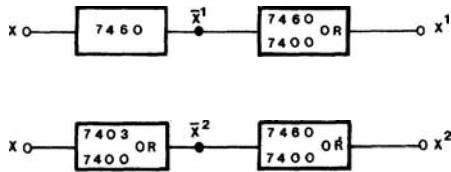


Fig. 6. TTL threshold detectors.

TABLE VIII

x	x^1	x^2	$\bar{x}^1(x)$	$\bar{x}^1(x^1)$	$\bar{x}^1(x^2)$	$\bar{x}^2(x)$	$\bar{x}^2(x^1)$	$\bar{x}^2(x^2)$
0	0	0	2	2	2	2	2	2
1	2	0	0	0	2	2	0	2
2	2	2	0	0	0	0	0	0

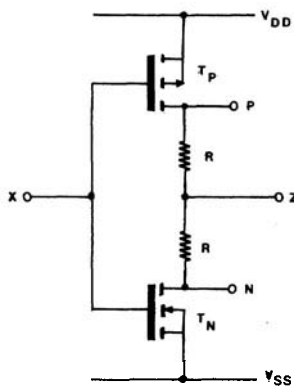


Fig. 7. COSMOS ternary inverter.

TABLE IX

x	X	Tp	Tn	P	N	Z	z
0	$X < V_{SS} + V_n$	on	off	V_{DD}	?	V_{DD}	2
1	$V_{SS} + V_n < X < V_{DD} - V_p$	on	on	V_{DD}	V_{SS}	$\frac{V_{DD} + V_{SS}}{2}$	1
2	$X > V_{DD} - V_p$	off	on	?	V_{SS}	V_{SS}	0

of inputs of serial p (or n) channel transistors, or as the MIN (or MAX) of inputs of parallel p (or n) channel transistors.

ECL TECHNOLOGY

Dunderdale presented in [2] the ECL ternary inverter and the ECL ternary Post cycling gate. These circuits are shown in Figs. 12 and 13. Their operating is shown in Table XIII (inverter) and Table XIV (cycling gate). There are two ways to realize the ECL encoder. The first one uses two I-current generators. The second one uses one I-current generator and one 2I-current generator.

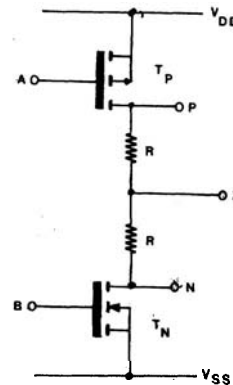


Fig. 8. COSMOS encoder.

TABLE X

a	b	Tp	Tn	P	N	Z	z
0	0	on	off	V_{DD}	?	V_{DD}	2
0	2	on	on	V_{DD}	V_{SS}	$\frac{V_{DD} + V_{SS}}{2}$	1
2	0	off	off	?	?	?	?
2	2	off	on	?	V_{SS}	V_{SS}	0

TABLE XI

a	b	z
0	0	2
0	2	1
2	2	0

TABLE XII

	z_2 z_1	z_2 z_0	z_1 z_2	z_1 z_0	z_0 z_2	z_0 z_1
a	$\bar{z}_2 \cdot \bar{z}_1$	$\bar{z}_2 \cdot z_0$	$\bar{z}_1 \cdot \bar{z}_2$	$\bar{z}_1 \cdot z_0$	z_0	z_0
b	\bar{z}_2	\bar{z}_2	$z_1 + \bar{z}_2$	$z_1 + z_0$	$\bar{z}_2 + z_0$	$z_0 + z_1$

The two ECL encoders are given in Figs. 14 and 15. Their physical working is shown in Tables XV and XVI. Their logical functions are given in Tables XVII and XVIII with the simplification illustrated in Fig. 16. As a result of the equivalence of the circuits shown in Fig. 16, the relations for a and b inputs according to the z_i, z_j may be simplified by using a' and b' input (Tables XIX and XX).

The ECL decoder is shown in Fig. 17.

IMPLEMENTATION OF TERNARY FUNCTIONS

We will give some examples of the implementation of ternary functions with TTL and COSMOS technologies. Included are some examples of simplification methods with one unary function (Post cycling function) and one binary function (comparator).

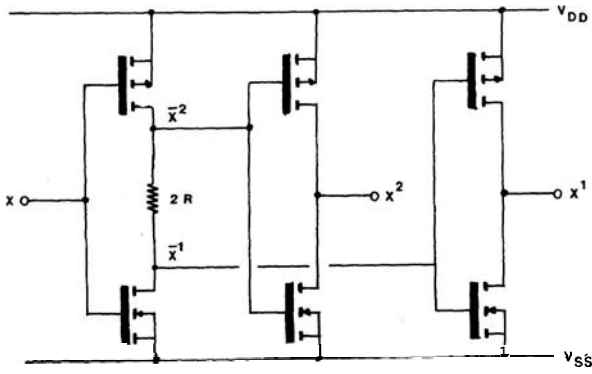


Fig. 9. COSMOS threshold decoder.

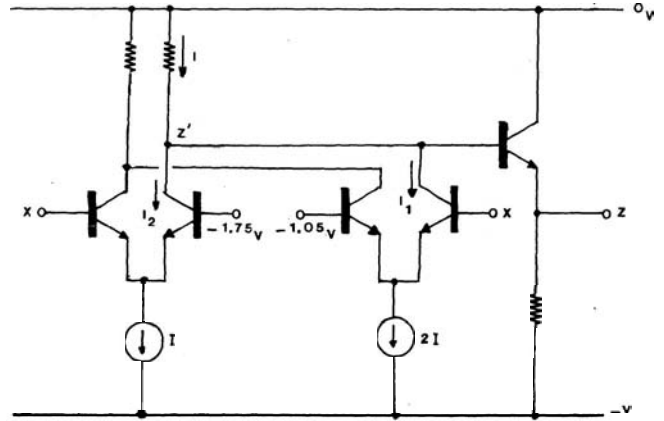


Fig. 13. ECL Post cycling gate.

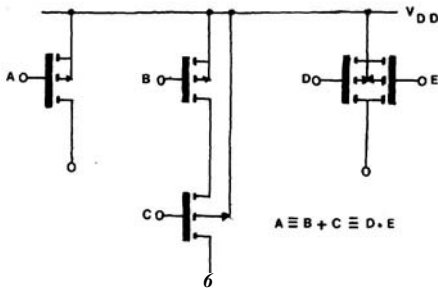


Fig. 10. p channel MOS equivalence.

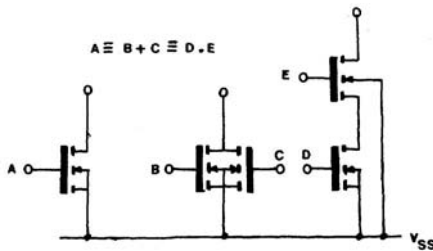


Fig. 11. n channel MOS equivalence.

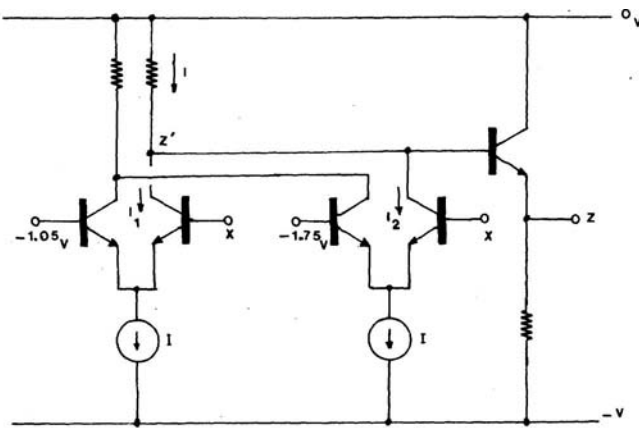


Fig. 12. ECL ternary inverter.

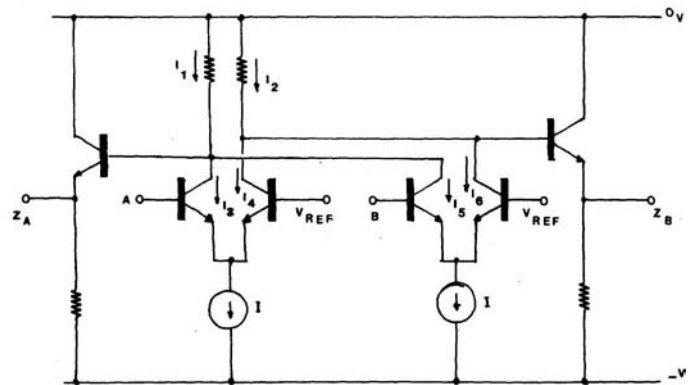


Fig. 14. ECL symmetrical encoder.

TABLE XIII

x	X	i ₁	i ₂	i	Z'	Z	z
0	-2,1V	0	0	0	0V	-0.7V	2
1	-1,4V	0	I	I	-0.7V	-1.4V	1
2	-0.7V	I	I	2I	-1,4V	-2.1V	0

TABLE XIV

x	X	i ₁	i ₂	i	Z'	Z	z
0	-2.1V	0	I	I	-0.7V	-1.4V	1
1	-1.4V	0	0	0	0V	-0.7V	2
2	-0.7V	2I	0	2I	-1.4V	-2.1V	0

Example 1— Post Function:

Consider the f₁₂₀ function. The possible pairs are

$$z_1 = \bar{x}^1 \quad z_0 = x^2 \quad z_0 = x^2 \quad z_1 = \bar{x}^1$$

$$z_2 = \bar{x}^2 \quad z_2 = x^1 \quad z_1 = \bar{x}^1 \quad z_0 = x^2.$$

With TTL implementation, the pair z₀,z₁ leads to a minimum circuit

$$a = z_0 = \bar{x}^1$$

$$b = z_1 = x^2.$$

The circuit before simplification is shown in Fig. 18. The relations $\bar{x}^1(x^2) = \bar{x}^2(x)$ (3) allows the simplification shown in Fig. 19.

With COSMOS implementation, we may also use the

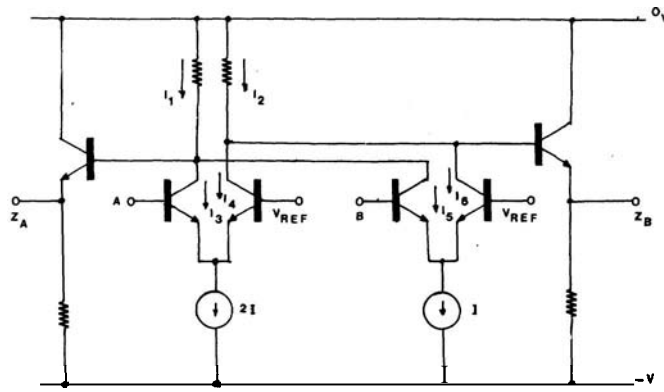


Fig. 15. ECL nonsymmetrical encoder.

TABLE XV

a	b	i_3	i_4	i_5	i_6	i_1	i_2	Z_a	Z_b	z_a	z_b
0	0	0	I	0	I	0	2I	-0.7v	-2.1v	2	0
0	2	0	I	I	0	I	I	-1.4v	-1.4v	1	1
2	0	I	0	0	I	I	I	-1.4v	-1.4v	1	1
2	2	I	0	I	0	2I	0	-2.1v	-0.7v	0	2

TABLE XVI

a	b	i_3	i_4	i_5	i_6	i_1	i_2	Z_a	Z_b	z_a	z_b
0	0	0	2I	0	I	0	3I	-0.7v	-2.8v	2	?
0	2	0	2I	I	0	I	2I	-1.4v	-2.1v	1	0
2	0	2I	0	0	I	2I	I	-2.1v	-1.4v	0	1
2	2	2I	0	I	0	3I	0	-2.8v	-0.7v	?	2

TABLE XVII

a	b	z_a	z_b
0	0	2	0
0	2	1	1
2	2	0	2

TABLE XVIII

a	b	z_a		a	b	z_b
0	0	2		0	2	0
0	2	1		2	0	1
2	0	0		2	2	2

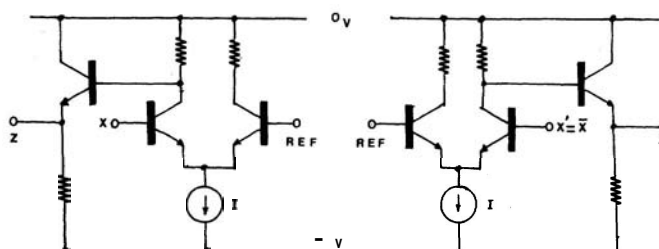


Fig. 16. ECL circuits equivalence.

TABLE XIX

z_a	$z_2 \quad z_1$	$z_2 \quad z_0$	$z_1 \quad z_2$	$z_1 \quad z_0$	$z_0 \quad z_2$	$z_0 \quad z_1$
a	$\overline{z_2 + z_1}$	$\overline{z_2 + \overline{z_0}}$	$\overline{z_1 + z_2}$	$\overline{z_1 + \overline{z_0}}$	z_0	z_0
a'	$z_2 + z_1$	$z_2 + \overline{z_0}$	$z_1 + z_2$	$z_1 + \overline{z_0}$		
b	$\overline{z_2}$	$\overline{z_2}$	$z_1 + \overline{z_2}$	z_1	$\overline{z_2} + z_0$	$z_0 + z_1$
b'	z_2	z_2				
z_b	$z_2 \quad z_1$	$z_2 \quad z_0$	$z_1 \quad z_2$	$z_1 \quad z_0$	$z_0 \quad z_2$	$z_0 \quad z_1$
a	z_2	z_2				
a'	$\overline{z_1}$	$\overline{z_2}$	$z_1 + \overline{z_2}$	$z_1 + z_0$	$z_0 + \overline{z_2}$	$z_0 + z_1$
b	$z_1 + z_2$	$z_2 + \overline{z_0}$	$z_1 + z_2$	$z_1 + \overline{z_0}$	$\overline{z_0}$	$\overline{z_0}$
b'					z_0	z_0

TABLE XX

z_a	$z_2 \quad z_1$	$z_2 \quad z_0$	$z_1 \quad z_2$	$z_1 \quad z_0$	$z_0 \quad z_2$	$z_0 \quad z_1$
a	$\overline{z_2 + z_1}$	$\overline{z_0 + z_2}$			z_0	z_0
a'	$z_2 + z_1$	$\overline{z_0} + z_2$	$z_1 + z_2$	$z_1 + \overline{z_0}$	$\overline{z_0}$	$\overline{z_0}$
b			z_1	z_1		
b'	$z_2 + \overline{z_1}$	$z_2 + z_0$	$\overline{z_1}$	$\overline{z_1}$	$z_0 + z_2$	$z_0 + \overline{z_1}$
z_b	$z_2 \quad z_1$	$z_2 \quad z_0$	$z_1 \quad z_2$	$z_1 \quad z_0$	$z_0 \quad z_2$	$z_0 \quad z_1$
a	$z_2 + z_1$	$z_2 + \overline{z_0}$	$z_1 + z_2$	$z_1 + \overline{z_0}$	$\overline{z_0}$	$\overline{z_0}$
a'						
b	$z_2 + \overline{z_1}$	$z_2 + z_0$	$\overline{z_1} + z_2$	$\overline{z_1} + z_0$	$z_0 + z_2$	$z_0 + \overline{z_1}$
b'						

$$z_0 = \overline{x^1}y^1 + \overline{x^2}y^2 \tag{5}$$

$$z_1 = \overline{x^1} + y^2 + \overline{x^2}y^1. \tag{6}$$

TTL implementation using the pair z_0, z_1 , (Table VI) and De Morgan's laws leads to the following equations.

$$a = \overline{\overline{x^1}y^1} \cdot \overline{\overline{x^2}y^2}$$

$$b = \overline{x^1 \cdot \overline{y^2} \cdot \overline{x^2}y^1}.$$

Before simplification the corresponding scheme is shown in Fig. 21. Wired-AND capabilities allow us to minimize the number of 7460 gates by synthesizing $a' = \overline{a}$ and $b' = \overline{b}$ at the output of the encoder.

$$a' = \overline{\overline{x^1} \cdot y^1} \cdot \overline{\overline{x^2}y^2}$$

$$b' = \overline{x^1 - \overline{y^2} \cdot \overline{x^2}y^1}.$$

TABLE XXI

	x			
y		0	1	2
0		1	2	2
1		0	1	2
2		0	0	1

The simplified scheme is shown in Fig. 22. COSMOS implementation may use the pair $z_2 z_1$.

$$a = \overline{z^2} \cdot z^1 \tag{Table XII}.$$

$$b = \overline{z^2}$$

$$z_2 = x^1 \overline{y^1} + x^2 \overline{y^2} \quad \text{with } z_2 < z_1.$$

$$z_1 = \overline{y^1} + x^2 + x^1 \overline{y^2}$$

$$\overline{z_2} = \overline{x^1 \overline{y^1} \cdot x^2 \cdot \overline{y^2}}$$

$$\overline{z_1} = y^1 \cdot \overline{x^2} \cdot x^1 \overline{y^2} \quad \text{with } \overline{z_1} < \overline{z_2}.$$

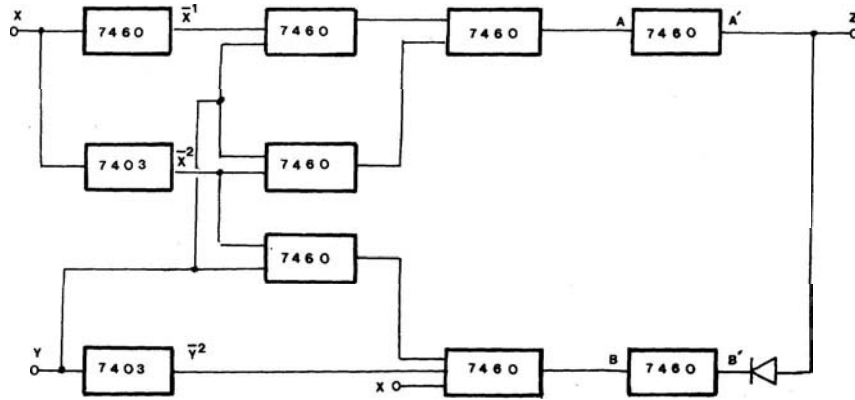


Fig. 21. Nonsimplified TTL comparator.

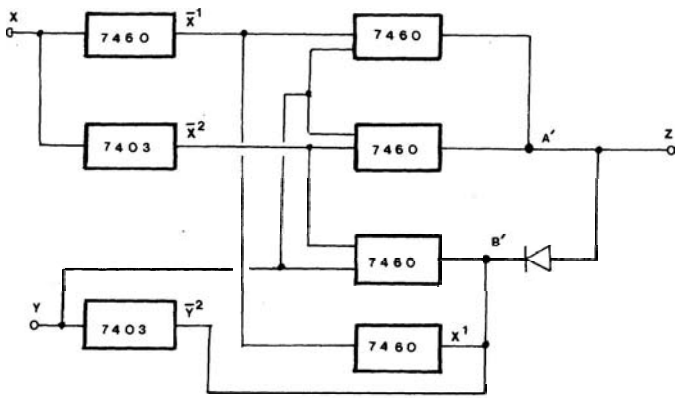


Fig. 22. Simplified TTL comparator.

TABLE XXII

		x					
		0	1	2			
y	0	1	2	2	1	$\bar{x}^1 y^1$	
	1	0	1	2	2	$\bar{x}^2 y^2$	
	2	0	0	1	1	$\bar{x}^1 y^1$	
		3	1	2	5	4	
							5 $\bar{x}^2 y^1$

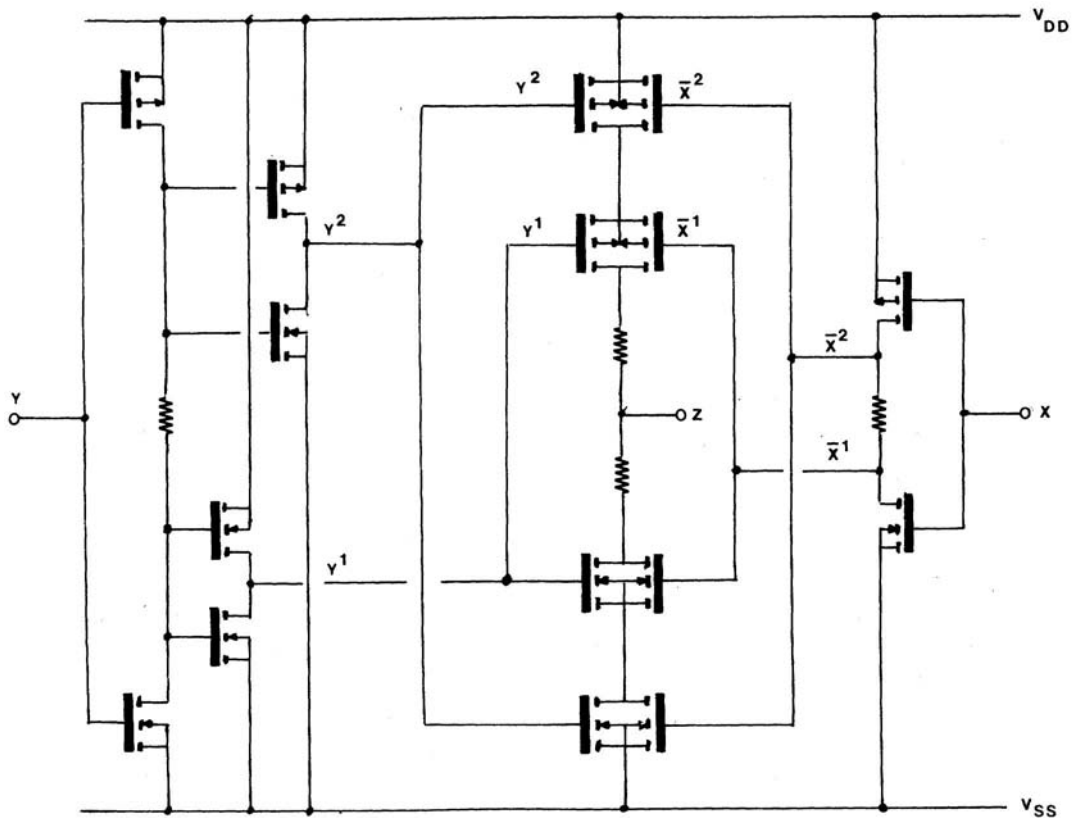


Fig. 23. COSMOS comparator.

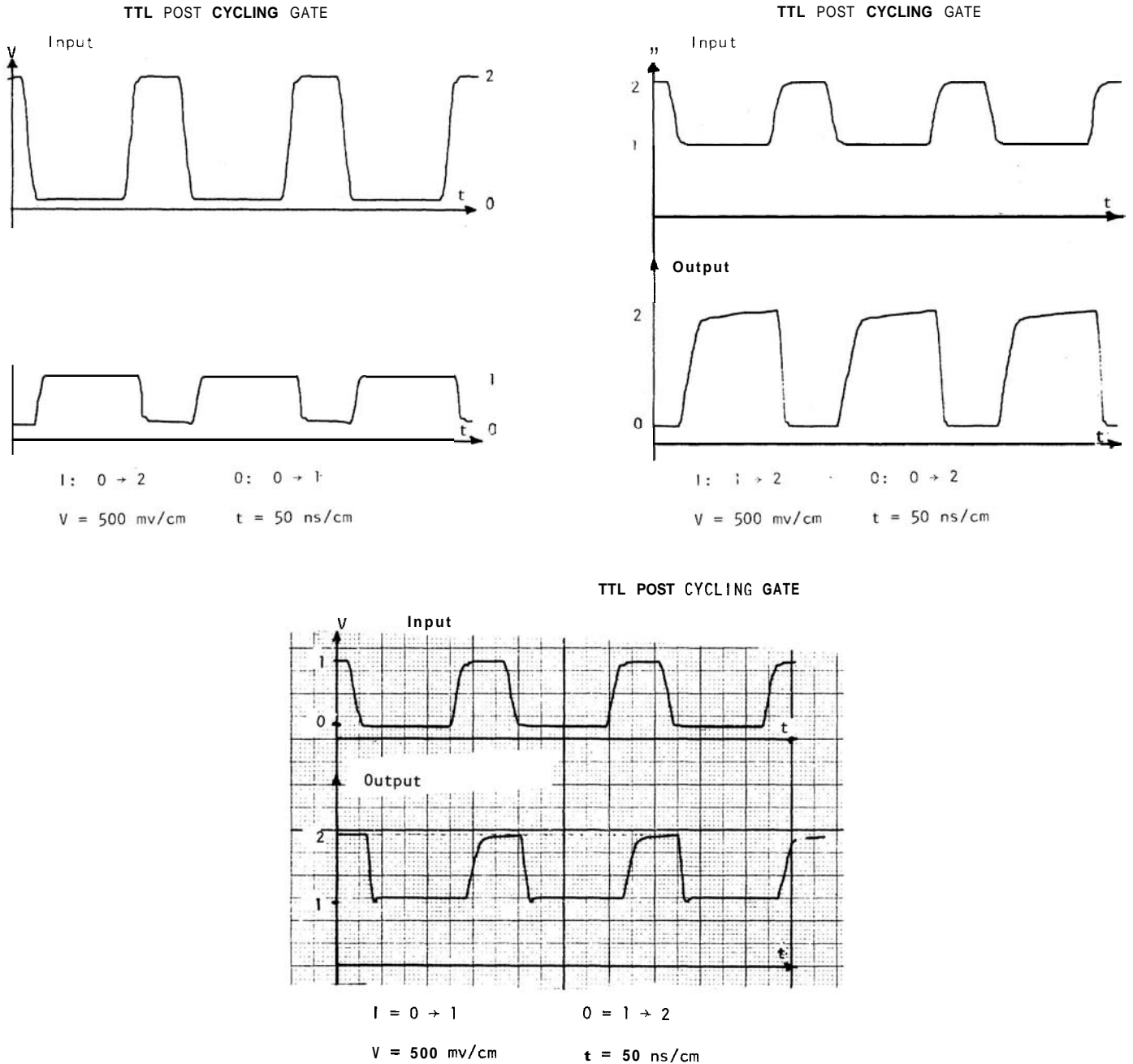


Fig. 24. Dynamical characteristics of TTL Post cycling gate.

$$a = \bar{z}_1 \cdot \bar{z}_2 = \bar{z}_1 = y^1 \cdot \bar{x}^2 \cdot \overline{x^1 \bar{y}^2}$$

$$= y^1 \cdot \bar{x}^2 \cdot \bar{x}^1 + y^1 \cdot \bar{x}^2 \cdot y^2$$

$$b = \bar{z}_2 = \overline{x^1 \bar{y}^1} \cdot \overline{x^2 \cdot \bar{y}^2}$$

with the inequalities $\bar{x}^1 < \bar{x}^2$ and $y^1 > y^2$

$$a = y^1 \bar{x}^1 + y^2 \bar{x}^2$$

$$b = (y^1 + \bar{x}^1) \cdot (y^2 + \bar{x}^2)$$

The complete COSMOS comparator is shown in Fig. 23. Figs. 24 and 25 show the dynamic characteristics of the TTL and COSMOS cycling gates described.

CONCLUSION

This paper has defined the conditions required to implement ternary circuits with binary technologies. For a given technology, we must have two different threshold levels and then combine the components to get 3 stable states. When these two requirements are found, it is easy to realize the threshold detector (decoder) and the output binary to ternary conversion circuit (encoder). Our method allows one to implement any 3-valued function as soon as the decoder and the encoder are defined for a given technology. Used with simplification rules, the general method leads to minimum circuits. This work will be extended with

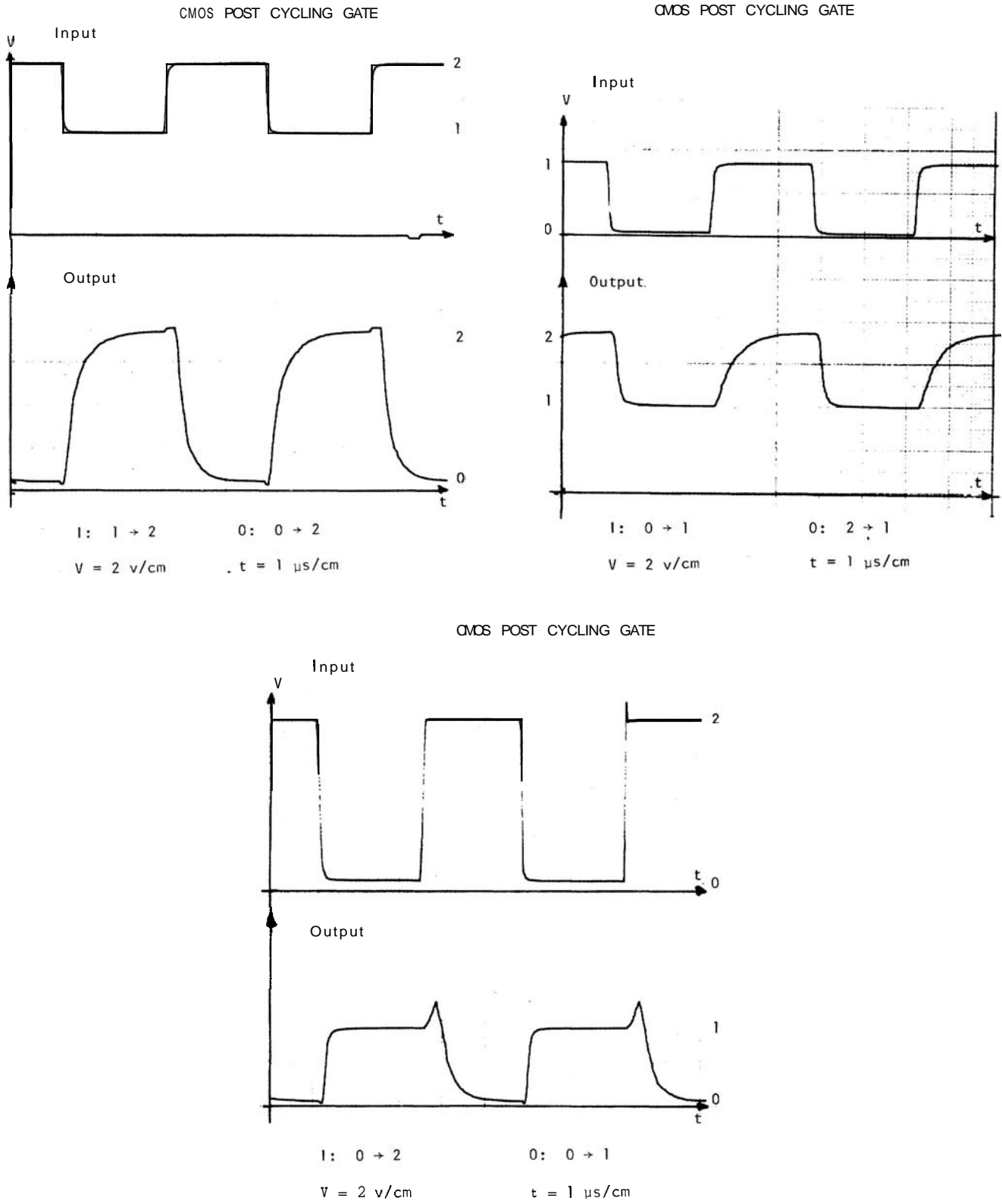
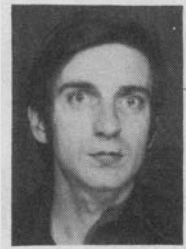


Fig. 25. Dynamical characteristics of COSMOS Post cycling gate.

further investigation into other technologies (1^2L) and with the generalization to n -valued circuits.

REFERENCES

- [1] H. Dunderdale, "Current mode circuits for ternary logic realizations," *Electron. Lett.*, pp. 575-577, 1969.
- [2] —, "Current mode circuits for the unary functions of a ternary variable," *Electron. Lett.*, pp. 15-16, 1970.
- [3] D. Etiemble and M. Israel, "A new concept for ternary logic elements," in *Proc. 1974 Int. Symp. Multiple-Valued Logic*, pp. 437-455, May 1974.
- [4] H. Mouftah and I. Jordan, "Integrated circuits for ternary logic," in *Proc. 1974 Int. Symp. Multiple-Valued Logic*, pp. 285-302, May 1974.
- [5] J. Birk and D. Farmer, "An algebraic method for designing multivalued logic circuits using principally binary components," *IEEE Trans. Comput.*, vol. C-24, pp. 1101-1104, Nov. 1975.
- [6] D. Etiemble and M. Israel, "Noise margins in multivalued logic circuits," presented at the 1976 Int. Symp. Multiple-Valued Logic, May 1976, Brochure IP 76-18-Université Paris.
- [7] H. Mouftah and I. Jordan, "Design of ternary COSMOS memory and sequential circuits," *IEEE Trans. Comput.*, vol. C-26, pp. 281-288, Mar. 1977.
- [8] The TTL Data Book for Design Engineers, Texas Instruments CC 411.
- [9] *COS/MOS Digital Integrated Circuits, Selection Guide Data Applications Notes*, RCA Solid State Data Book's Series SSD 203 A.



Daniel Etiemble was born in France in 1947. He received the Electrical Engineer degree in 1969 from the Institut National des Sciences Appliquées, Lyon, France, and the Ph.D. degree in engineering in 1974.

He has been Assistant Professor of Computer Science at the University of Paris, Paris, France since 1969. His research interests include logic design, multivalued logic, modeling, and computer organization.



Michel Israel was born in 1947. He received the Master of Science degree in 1973 and the 3rd Cycle Thesis degree in 1977, from the University of Paris, Paris, France.

He has been Assistant Professor at the Conservatoire National des Arts et Métiers (CNAM), Paris, France since 1973. His research interests include mainly multivalued logic.

Multivalued Integrated Injection Logic

T. TICH DAO, SENIOR MEMBER, IEEE, EDWARD J. MCCLUSKEY, FELLOW, IEEE, AND LEWIS K. RUSSELL

Abstract—A family of circuits for multivalued, in particular quaternary, integrated injection logic is described. The basic elements are the 1^2L current mirror and 1^2L threshold gates.

Index Terms—Multilevel 1^2L , multivalued logic, Post logic, quaternary logic, quaternary ROM, quaternary flip-flops, radix-4 arithmetic, threshold 1^2L .

MULTIVALUED digital circuits make feasible very compact computers because such circuits have high logic density, reduced interconnection area, and high production yields. The type of multivalued logic system to be implemented is determined mainly by the basic function which can be realized reliably in a modern technology. This paper describes a circuit family which uses 1^2L to implement a four-valued logic system which can

easily be modified to realize either a three-valued or a higher valued system.

Past attempts at developing multivalued circuits have concentrated on three-valued logic. Our choice of four-valued logic (a power of 2) is based on the belief that easy convertibility between binary systems and multivalued systems will be very important. The ability to combine the multivalued circuits with binary circuits in the same system is believed to be a necessity for the foreseeable future. Also using a higher radix (four rather than three) improves the gain (density, area, yield) from using multilevel circuits.

In any digital system signal levels can vary from their nominal levels and must be restored or requantized before the variation becomes so large that the signals lose their information content. In most practical situations the electrical level corresponding to a logic value might be deteriorated to a point where it must be restored to its normalized level, especially inside a feedback loop. Therefore, reliable means of level-detection threshold gates must be provided. It turns out that broad use of

Manuscript received March 28, 1977; revised June 22, 1977.

T. T. Dao and L. K. Russell are with Signetics Corporation, Sunnyvale, CA 94086.

E. J. McCluskey was with Signetics Corporation, Sunnyvale, CA 94086. He is now with the Digital Systems Laboratory, Stanford University, Stanford, CA 94305.