Ternary Rate-Multipliers

H. T. MOUFTAH, K. C. SMITH, AND Z. G. VRANESIC

***Abstract*—This correspondence describes an application of COS/MOS integrated circuits in the design of ternary rate-multipliers. The logical implementation of two types of ternary rate-multiplier is described. The first type produces an unevenly spaced output pulse train, while in the second one the output pulses are uniformly spaced. Advantages and disadvantages of the two types are presented.**

***Index Terms*—Integrated circuits, rate-multipliers, three-valued logic.**

INTRODUCTION

The rate-multiplier is a device which produces an output pulse train whose frequency is proportional to the product of the frequency of a reference clock and a number (the multiplier) whose value is fixed at any given instant of time. A number of reports have been published recently on the design of binary and binary coded decimal (BCD) rate-multipliers [1]–[7], [11]. None of them considered the possibility of using nonbinary logic in such designs, which may offer the advantages of improved speed and reduced cost. This paper presents the design of rate-multipliers using three-valued logic.

Rate-multipliers may be split into two major classes depending on their output pulse distribution: unevenly or uniformly spaced. The first type is usually composed of an up-counter (or a shift register [11]) and a small amount of control logic [1], [2], [7]. The second type is more sophisticated and requires more circuitry. It is generally composed of a full adder, an accumulator, and some control logic [3], [7]. However, it may also be realized using a programmable counter, control logic, and an additional clock pulse [5], [6]. Advantages and disadvantages of these two approaches are discussed in [7]. In the present paper the feasibility of the ternary realization of each rate-multiplier type will be discussed.

In a three-valued rate-multiplier the input reference frequency as well as the output frequency are binary (pulses from low to high) while all logic elements (counters and gates) and the input multiplier number are ternary. They are ternary in the sense that a signal on any single wire may have one out of three possible values (0, 1, or 2). Physically, these three values will be represented by −4 V for 0 (low level), 0 V for 1 (intermediate), and +4 V for 2 (high). The ternary circuits used are designed and realized with commercially available COS/MOS integrated circuit packages, namely, the CD4007 and CD4016.

LOGICAL DESIGN DESCRIPTION

The three-valued algebra used in the present design was described in [9]. Briefly, it is composed of the ternary AND (product function), OR (sum function), and three types of inversions (positive, simple, and negative inverters). The ternary AND and ternary OR are defined as follows:

$$x \cdot y = \min(x, y) \tag{1}$$

$$x + y = \max(x, y) \tag{2}$$

where $x, y \in L$, $L = \{0, 1, 2\}$.

The simple ternary inverter (STI) is defined by

$$\overline{x^1} = 2 - x. \tag{3}$$

The positive ternary inverter (PTI) and the negative ternary inverter (NTI) are both defined by

$$\overline{x^i} = \begin{cases} i & \text{if } x \neq i \\ 2 - i & \text{if } x = i \end{cases} \tag{4}$$

where i takes the value of two for the PTI and zero for the NTI operator. The minus sign in (3) and (4) represents arithmetic subtraction. The ternary NAND and NOR operators are defined as inversions (positive, simple, and negative) of the ternary AND and OR operators, respectively. Circuit realizations of these operators using COS/MOS integrated circuits have been presented in [9].

The block diagram of a ternary rate-multiplier is shown in Fig. 1. It is composed essentially of a ternary up-counter and some control logic. Each stage of this counter consists of a ternary master–slave T-type flip-flop (3-flop) which has a triggering input T and inputs P, Z, and N for resetting or presetting the 3-flop to the "2" (positive), "1" (zero), and "0" (negative) states, respectively [10]. At each clock pulse (low to high) the 3-flop changes its state so that

$$Q(t + 1) = (Q(t) + 1) \bmod 3 \tag{5}$$

where $Q(t)$ and $Q(t + 1)$ are the outputs of the 3-flop before and after the clock pulse is applied, and the plus sign represents arithmetic addition.

An NTI circuit connects the output of each 3-flop to the input of the next one. The function of this NTI circuit is to change the state of the 3-flop only when the state of the preceding 3-flop changes from high to low levels. This can be given by the following equation:

$$T_n = \overline{Q^0_{n-1}} \tag{6}$$

where T_n represents the input of the nth stage and Q_{n-1} the output of the $(n - 1)$th stage of the counter.

One of the main objectives in the design of a ternary rate-multiplier is to determine the control gating necessary to produce output pulses coinciding with the 0–1 and/or the 1–2 transitions of each 3-flop. The 0–1 transitions and the 1–2 transitions of the first 3-flop which carries the least significant trit, are such that one transition from each type occurs for every three clock pulses. Thus, the output of this 3-flop will have a frequency (number of pulses per unit time) equal to $\frac{1}{3}f$ for 0–1 or 1–2 transitions, but $\frac{2}{3}f$ for 0–2 (0–1 plus 1–2) transitions, where f is the frequency of the input clock. Similarly, the output of the second 3-flop will have transitions 0–1 (or 1–2) with frequency $\frac{1}{9}f$, and transitions 0–2 with frequency $\frac{2}{9}f$. In general, the output of the nth 3-flop will have transitions 0–1 (or 1–2) with frequency $\frac{1}{3^n}f$, and transitions 0–2 with frequency $\frac{2}{3^n}f$. Since none of these transitions (0–1 and 1–2) occur simultaneously, the pulse outputs coinciding with 0–1 and 1–2 transitions can be gated in a ternary OR to give a single pulse train output whose frequency over a specified time period is some fraction of the input frequency. In general, the output frequency of this ternary rate-multiplier will range from 0 to $(3^n - 1)/3^n f$, where n is the number of stages of the counter.

A 0–1–0 or 0–2–0 pulse occurs at the output of the ternary AND gate A_1 of the least significant trit 3-flop only when the most significant trit of the multiplier (S_n) is 1 or 2. For every three input pulses, if $S_n = 2$, two pulses must pass to A_1, if $S_n = 1$ only one pulse is required, and if $S_n = 0$ no pulses are necessary. A detector is needed to specify if S_n is 0, 1, or 2. Fig. 2(a) and (b) shows the truth table and schematic diagram, respectively, of a Z-detector designed for this purpose. It is composed of a T-gate and a J_1 operator. The T-gate function is defined as

$$T(x_1, x_2, x_3; s) = x_i \tag{7}$$

where i will be equal to 1 if s takes the value of 0, 2 if s is 1, and 3 if s is 2.

Manuscript received August 9, 1979; revised March 20, 1980. This work was supported in part by the Ministry of Education of the Government of Quebec and the Natural Sciences and Engineering Research Council of Canada.

H. T. Mouftah is with the Department of Electrical Engineering, Queen's University, Kingston, Ont., Canada.

K. C. Smith and Z. G. Vranesic are with the Department of Electrical Engineering, University of Toronto, Toronto, Ont., Canada.

0018-9340/80/1000-0929$00.75

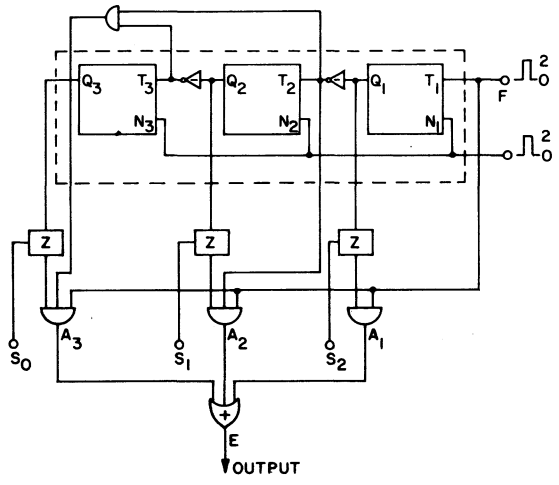


Fig. 1. A ternary rate-multiplier.

The T-gate circuit is composed of a J_K operator and three ternary switches (transmission gates). The J_K operator is defined by

$$J_K(x) = \begin{cases} 2 & \text{if } x = K \\ 0 & \text{if } x \neq K \end{cases} \tag{8}$$

where K can take the values of 0, 1, or 2.

The output of the Z-detector is 0 if S is 0, $J_1(Q)$ if S is 1, and Q if S is 2. Thus, A_1 may be given by

$$A_1 = F \cdot T(0, J_1(Q_1), Q_1; S_n) \tag{9}$$

where F is the input pulse driving the counter.

Similarly, for $A_2, A_3, \cdots, A_n$. Moreover, pulses are allowed to pass at the output of each 3-flop only when all 3-flops of the previous stages are in state 0. Thus, $A_2, A_3, \cdots, A_n$ may be given by

$$A_2 = F \cdot T(0, J_1(Q_2), Q_2; S_{n-1}) \cdot \overline{Q_1^0} \tag{10}$$

$$A_3 = F \cdot T(0, J_1(Q_3), Q_3; S_{n-2}) \cdot \overline{Q_1^0} \cdot \overline{Q_2^0} \tag{11}$$

$$\vdots$$

$$A_n = F \cdot T(0, J_1(Q_n), Q_n; S_1) \cdot \overline{Q_1^0} \cdot \overline{Q_2^0} \cdots \overline{Q_{n-1}^0}. \tag{12}$$

Outputs $A_1, A_2, \cdots, A_n$ are then gated in a positive ternary OR to supply only a low to high output pulse train. The positive ternary OR function is realized by a ternary NOR gate with a PTI circuit inserted in its output. The output waveform of a ternary rate-multiplier with a two stage counter assembled in our laboratory with CD4007 and CD4016 COS/MOS integrated circuit packages is shown in Fig. 3. In this example the inputs S_1 and S_2 (the multiplier) are 2 and 1, respectively, representing the decimal number five. So, five pulses are shown on the output for every nine pulses in the input.

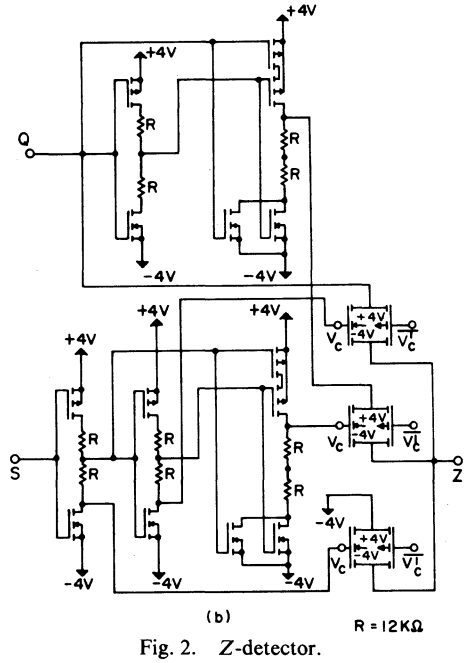
Since the AND and OR are more costly than the NAND and NOR, and since DeMorgan's law holds for the three types of inversion (positive, simple, and negative inverter), it is possible to reduce the cost of the system by replacing the ternary AND gates (of outputs $A_1, \cdots, A_n$) by ternary NAND gates and the positive ternary OR by a positive ternary NAND in Fig. 1. This is shown clearly by the following equation:

$$\overline{E = \overline{A_1 + A_2 + \cdots + A_n^1}^2} = \overline{\overline{A_1^1} \cdot \overline{A_2^1} \cdots \overline{A_n^1}}^2. \tag{13}$$

The ternary up-counter presented above has identical stages, which means that extra stages can be added or subtracted as required depending on the application of the rate-multiplier. Since the number of transistors realizing the different gates and 3-flops is about the same as in their corresponding binary COS/MOS elements [10] (except for addition of the Z-detector), and since the number of ternary counter stages required to multiply a frequency by a certain rate is smaller than in binary counters for the same rate (15 decimal digits ≡ 54 bits ≡ 34 trits, for example), it is obvious that this ternary realization offers some economic advantages over the corresponding binary one. However, this ternary rate-multiplier possesses the same disadvantage of the prototype binary counter rate-multiplier [7], namely, unevenly spaced output pulses which are associated with "roundoff error." Also, for a large number of stages, speed is limited since the counter operates asynchronously.

S	Q	Z
2	2	2
2	1	1
2	0	0
1	2	0
1	1	2
1	0	0
0	2	0
0	1	0
0	0	0

(a)



(b)

Fig. 2. Z-detector.

To overcome some of those disadvantages, a ternary rate-multiplier with uniformly spaced output pulses has also been realized. It is based on the idea used in [5] and [6]. The block diagram of this ternary rate-multiplier is presented in Fig. 4. It is composed of a divide-by-M ternary counter, a control 3-flop, and a subclock pulse generator. The divide-by-M ternary counter, where M is a positive integer, is formed essentially by the up-counter presented above, with a J_K circuit mounted on the output of each stage [10]. A ternary AND gate takes the outputs of those J_K circuits and feeds back to reset the counter and the control 3-flop after a fixed number of pulses depending upon what output is taken from each J_K circuit (J_0, J_1, or J_2), i.e., the multiplier number. The control 3-flop is set to two (high) for each input pulse. This enables the ternary AND gate (1) to pass a number of pulses equal to the multiplier number, after which each counter stage and the control 3-flop are reset to 0 (low). This will disable the ternary AND gate (1) until the control 3-flop is set to two again by another input pulse, and so on.

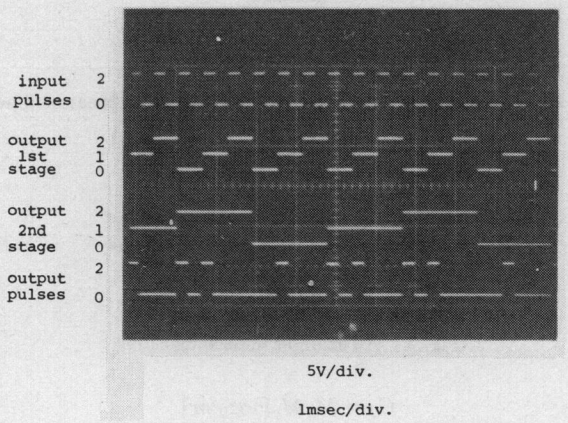


Fig. 3. Output wave forms of a ternary rate-multiplier of two stages when $S = 5$.

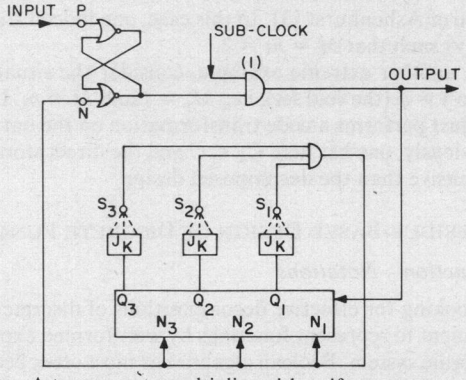


Fig. 4. A ternary rate-multiplier with uniform output pulses.

It is clear that the output pulse frequency in this case is not a fraction of the input frequency but a multiple of it. If a fraction of the input frequency is desired, an added divide-by-M ternary counter and a control 3-flop have to be used exactly as in [6]. The pulse rate f_s of the subclock must be chosen to satisfy the following condition:

$$f_s \geq 3^n f$$

where f is the fixed input pulse frequency.

In the case of equality the output pulses will be uniformly spaced. In this case the subclock must be well stabilized.

Finally, it must be mentioned that a ternary accumulator rate-multiplier, with output pulses uniformly spaced, can be directly realized using the ternary full adder and the D-type 3-flop presented in [3] and [10], respectively. This will not be described in this correspondence since it is a straightforward realization of the method described in the binary case and presented in [3] and [7].

Conclusion

Two methods of design of ternary rate-multipliers with COS/MOS integrated circuits has been presented. Such ternary rate multipliers may be used in various applications, including numerical control, instrumentation, digital filtering, and frequency synthesis. They provide some economic advantages over the corresponding binary designs. However, they can be used efficiently when the speed of operation is not of great significance. This ternary realization of rate-multipliers demonstrates, again, the feasibility of the practical realization of three-valued logic to the design of digital systems.

References

[1] A. Dunworth, "Digital instrumentation for angular velocity and acceleration," *IEEE Trans. Instrum. Meas.*, vol. IM-18, pp. 132–138, June 1969.

[2] A. R. Elliot, "A high-speed binary rate-multiplier," *Proc. IEEE*, vol. 59, pp. 1256–1257, Aug. 1971.

[3] R. M. M. Oberman, "A flexible rate-multiplier circuit with uniform pulse distribution outputs," *IEEE Trans. Comput.*, vol. C-21, pp. 896–899, Aug. 1972.

[4] V. C. V. Pratapa Reddy and K. P. Rajappan, "Decade rate-multiplier," *Proc. IEEE*, vol. 60, p. 759, June 1972.

[5] K. P. Rajappan and V. C. V. Pratapa Reddy, "Digital interpolation," *Int. J. Electron.*, vol. 36, no. 2, pp. 163–165, 1974.

[6] V. C. V. Pratapa Reddy and K. P. Rajappan, "A new type of rate-multiplier," *Int. J. Electron.*, vol. 39, no. 2, pp. 233–235, 1975.

[7] J. B. Peatman, *The Design of Digital Systems*. New York: McGraw-Hill, 1972.

[8] "COS/MOS digital integrated circuits," Selection Guide Data Application Notes, RCA Solid State Databook Series, SSD-203C, 1975.

[9] H. T. Mouftah and I. B. Jordan, "A design technique for an integrable ternary arithmetic unit," in *Proc. 1975 Int. Symp. Multiple Valued Logic*, pp. 359–372.

[10] H. T. Mouftah and I. B. Jordan, "Design of ternary COS/MOS memory and sequential circuits," *IEEE Trans. Comput.*, vol. C-26, pp. 281–288, Mar. 1977.

[11] W. H. Ninke and G. R. Ritchie, "Shift register binary rate-multipliers," *IEEE Trans. Comput.*, vol. C-26, pp. 276–278, Mar. 1977.

Read-Only Memory Implementation of Discrete Functions

MARC DAVIO

***Abstract*—Residue-based designs are shown to exist for a particular class of discrete functions called *regular functions*. If the design uses read-only memories (ROM's) only, it may be shown that the cost of the residue-based design may become significantly smaller than the cost of the straightforward tabular approach when the number of distinct function values is small with respect to the number of points in the definition domain.**

***Index Terms*—Logic design, read-only memories, residue arithmetic.**

I. Introduction

Since the early work of Svoboda [1], residue-based arithmetic has currently been the subject of a number of research works, and a number of publications [2], [3] have been devoted to the topic. Residue-based arithmetic actually exhibits a number of attractive features such as the low complexity of addition and multiplication. It, however, suffers some drawbacks: the most frequently mentioned ones are the complexity of division and the difficulty of overflow detection. Another disadvantage of residue-based design, which will be discussed in the present correspondence, is the relatively high cost of the back-conversion from the residue form to the usual binary representation. The complexity of this back-conversion process (Chinese remainder theorem or mixed radix conversion) is such that applications of residue arithmetic seem to be confined to situations involving many additions and/or multiplications and few conversions to the binary form.

However, in a recent paper [4], Jullien successfully applies residue arithmetic techniques to the design of a multiplier modulo a prime number; by successfully, we mean that his overall design (including the back-conversion from the residue to the binary form) uses less bits of read-only memory (ROM) storage than the direct ROM storage of the corresponding multiplication table. This is quite an incentive result: it indeed provides us with an example of isolated computation in which the cost of the back-conversion from residue to binary does not prevent one from using the residue-based design. The natural

Manuscript received August 7, 1979; revised February 11, 1980 and April 10, 1980.

The author is with Philips Research Laboratory, Brussels, Belgium.