Multimoding and its Suppression in Twisted Ring Counters

By W. BLEICKARDT

(Manuscript received March 21, 1968)

Many digital systems, such as PCM systems, data processing and data transmission systems, use twisted ring counters. Most of these twisted ring counters are subject to multimoding. This paper develops tools and methods for predicting all possible modes in twisted ring counters, and derives a general solution for suppressing the wrong modes. Suppression is accomplished by adding a few circuit connections from the output of certain stages to the input of another stage. The paper derives the number of necessary connection lines and their connection points for the various types of counters.

I. INTRODUCTION

Twisted ring counters of various types have been used for many years, and have been described in many publications.[1–5] They are designed for creating a well-defined periodic pulse pattern. But they all have one problem in common: under certain circumstances they can multimode, that is, they can create undesired patterns. Each mode of a counter creates a particular pattern. Only one of these modes is the desired one, the "correct mode;" the rest are all "wrong modes" and must be suppressed. To the knowledge of the author, none of the publications on twisted ring counters presents a rigorous treatment of the problem of multimoding, although it must have shown up in many instances and often was solved empirically.[5] The lack of a general theory on possible modes in twisted ring counters and on the prevention of undesired modes led to this investigation.

Terminology for the characterization of modes, and relations between the parameters, make it easy to find the entire set of possible modes for any twisted ring counter. There is a method for suppressing all wrong modes by adding a few circuit connections, and a general formula that indicates these additional connections for any individual

ring counter. The method for suppressing all wrong modes in any twisted ring counter is summarized in Section 5.5.

II. OPERATION OF TWISTED RING COUNTERS

A twisted ring counter consists of a shift register whose output is fed back over a twist to its input in a ringlike manner (Figs. 1, 2, and 3). An input clock keeps a certain pattern circulating around the ring. In the correct mode the stages create the desired pattern by switching on sequentially with subsequent clock pulses, and then switching off in the same sequence (part a of Figs. 1, 2, and 3).* With each clock pulse only one stage is switching. A counter with n stages creates a periodic pattern with a period of $2n$ time slots as shown in the first three figures. Some possible implementations of counter stages are shown in Fig. 4, using AND gates, NAND gates and set-reset flip-flops. Equivalent stages can be built by using OR gates and NOR gates, or any custom-designed circuit.

There are two general types of twisted ring counters: single-phase counters with one input clock line (example in Fig. 1), and double-phase counters with two input clock lines supplying interleaved pulses (examples in Figs. 2 and 3). Many of the single-phase counter stages, such as the ones shown in Fig. 4a and b, require short input clock pulses to prevent racing. The clock pulses must be shorter than the propagation delay of one stage. An example of a stage that does not require short clock pulses is shown in Fig. 4c.[4] Double-phase counters permit the use of simple gated set-rerest flip-flop stages (Fig. 4d and e) without the restriction of short clock pulses. Notice that in counters with an even number of stages (Fig. 2) the two clock phases are distributed in a different way from those in counters with an odd number of stages (Fig. 3).

The problem of multimoding arises whenever more than one mode can exist. In that case, errors can switch the counter to other (wrong) modes with undesired patterns. Such errors can be created by noise transients, aging components, marginal design, and so on. The first three figures show some examples of wrong modes. In general, the number of wrong modes possible increases with the number of stages of a counter, and is higher for single-phase counters than for double-phase counters. To design reliable circuits, one must prevent un-

* The numbers in parentheses in Figs. 1, 2, and 3 are a symbolic notation for different modes; they indicate the numbers of time slots a particular counter stage remains in one state. This notation is explained in Section III.

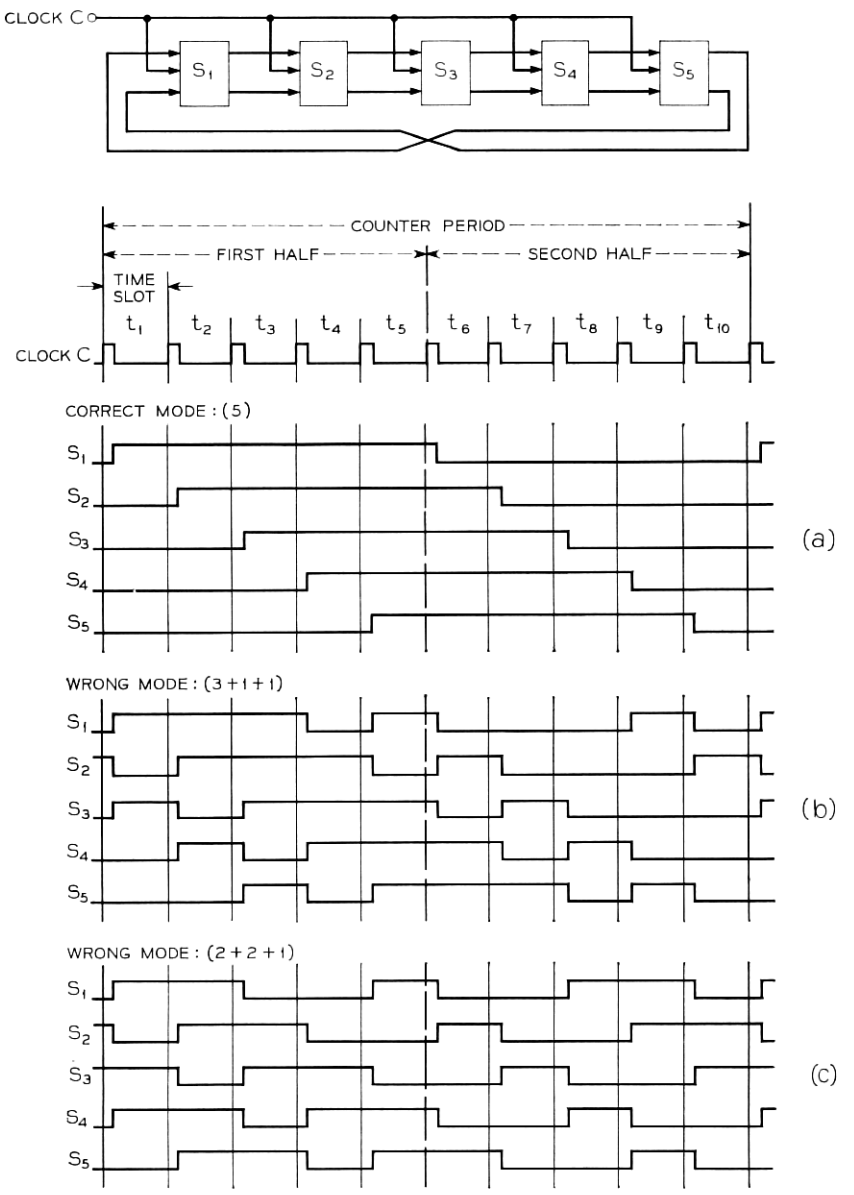


Fig. 1 — Single-phase twisted ring counter with five stages.

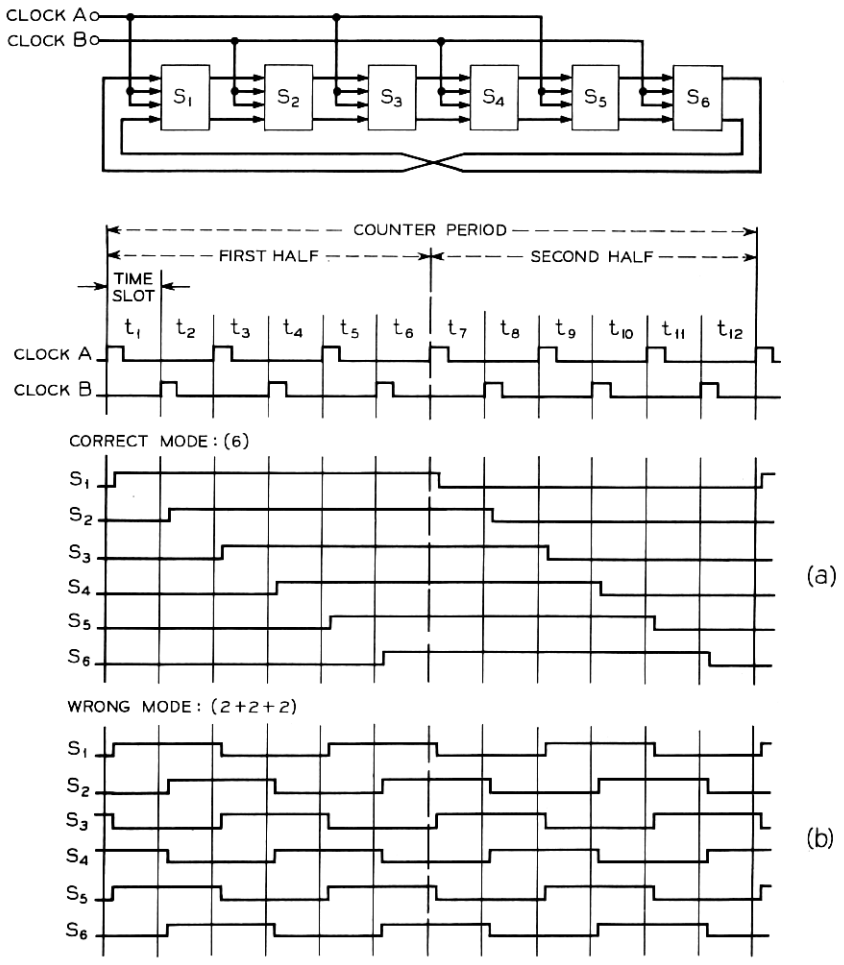


Fig. 2 — Double-phase twisted ring counter with even number of stages (six stages).

desired patterns from circulating for more than a very short time (typically less than one counter period).

III. GENERAL CHARACTERIZATION OF MODES

There is a unique way in which a pattern, that is, a sequence of states 0 or 1, is circulated around the counter ring. Any pattern is shifted by one stage per time slot, as can be seen from the pulse dia-

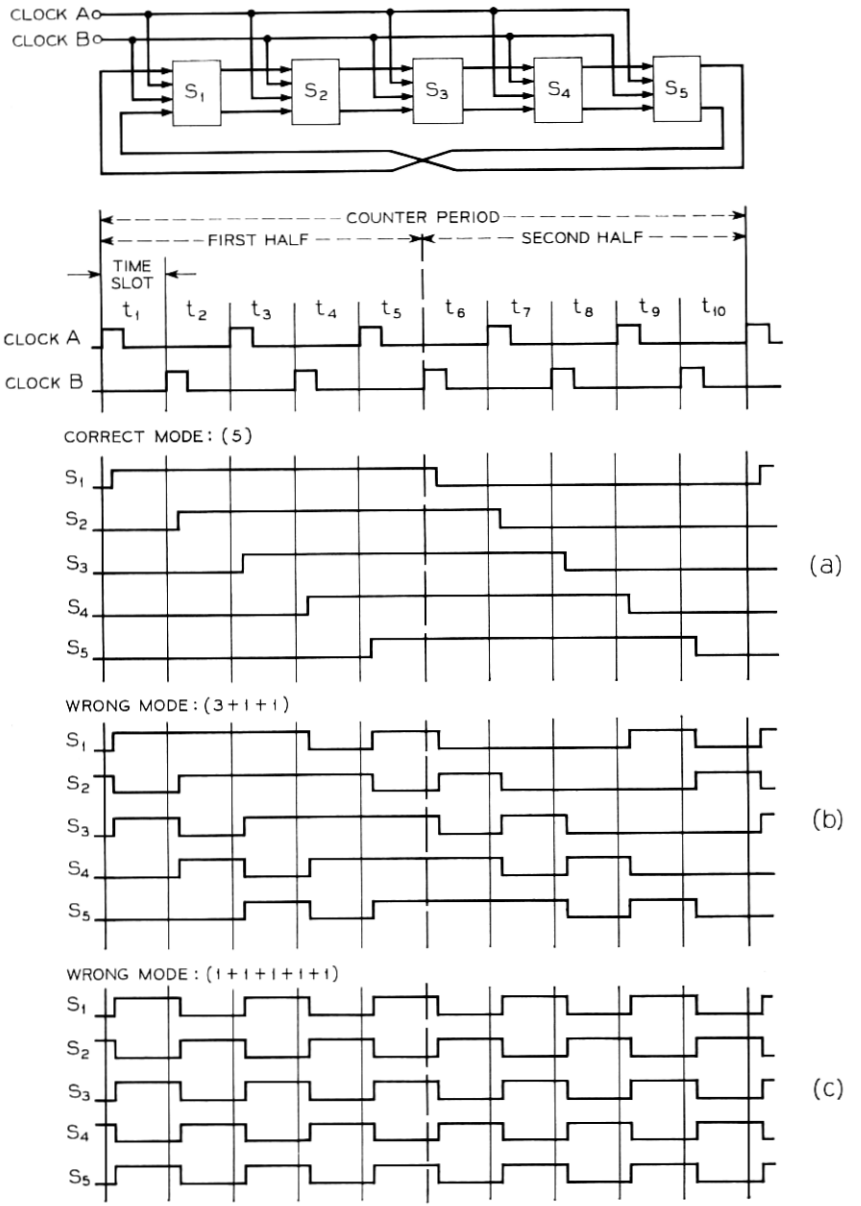


Fig. 3 — Double-phase twisted ring counter with odd number of stages (five stages).

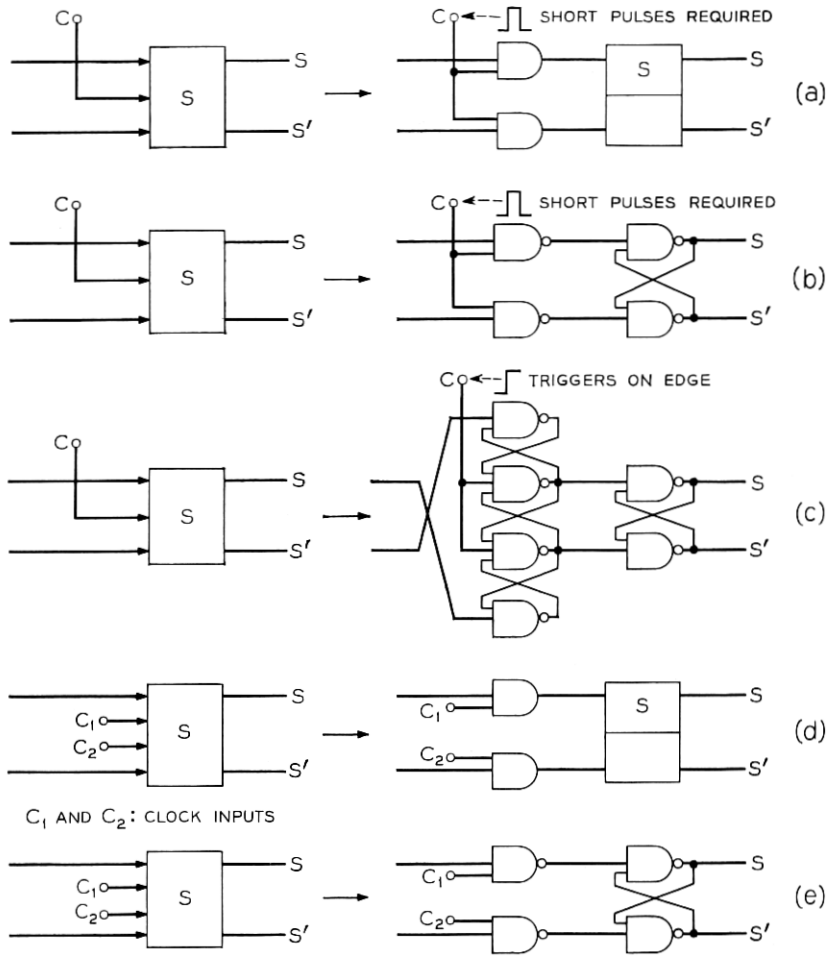


Fig. 4 — Implementation of counter stages with AND gates, NAND gates, and set-reset flip-flops; a, b, and c for single-phase counters, d and e for double-phase counters.

grams in Figs. 1, 2 and 3. The state of the last stage appears in inverted form at the first stage in the subsequent time slot. For a counter with n stages, the pattern, seen as a time sequence at each stage, repeats itself in inverted form after n time slots; the whole counter period is $2n$ time slots long.

This well-defined behavior allows us to reconstruct the entire pulse

diagram for a particular mode, if we only know the states of all n stages at any one time, or if we know a sequence of n states at any single stage. Therefore, a sequence of n binary digits uniquely describes a mode.

3.1 *Definitions*

(*i*) We will call the state of a particular stage in a particular time slot an *element*. An element can have a state 0 or 1.

(*ii*) Elements in successive time slots, or in successive stages that have the same state, form a *logic group*.

(*iii*) The *size* of a logic group (g_j) is the number of its elements.

(*iv*) The smallest size logic group of a particular mode has g_{min} elements.

(*v*) The positive direction of a sequence of elements corresponds to the sequence as observed on the positive time axis. This corresponds to a sequence backwards through the stages. (This can be illustrated with Fig. 1b. The sequence 1 1 1 0 1 appears at stage S_1 in the time slot sequence t_1, t_2, t_3, t_4, t_5, and it appears at time t_5 in the stage sequence S_5, S_4, S_3, S_2, S_1.)

3.2 *Description*

For describing one particular mode, it is sufficient to write the size and sequence of the logic groups g_j that are built by n elements. The following symbolic notation is used:

$$(g_1 + g_2 + g_3 + \cdots + g_x)$$

where

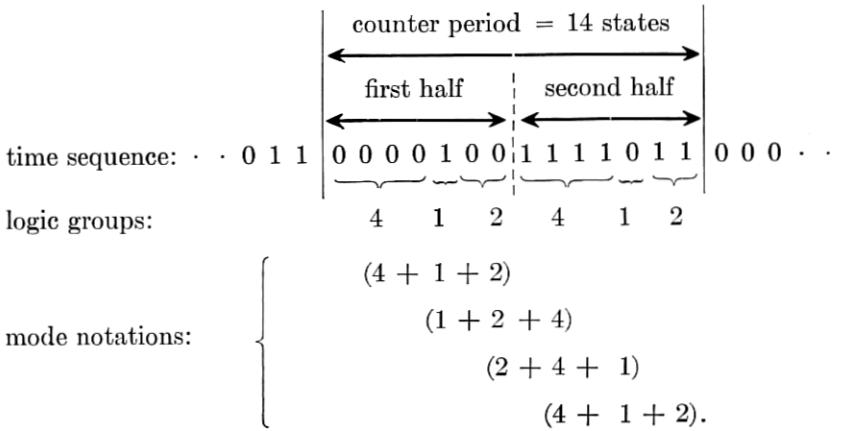
$$\sum_{j=1}^{x} g_j = n = \text{number of stages}$$

$$x = \text{odd number.}$$

For example, (3 + 1 + 1) denotes a mode of a 5-stage counter, with three logic groups, the first containing three elements, the second and third containing one element each (shown in Fig. 1b).

This symbolic notation describes one half of the periodic cycle. Since each half is always the complement of the other half, the elements of the first and the last logic group in the mode notation have the same state. Therefore, the number x of logic groups in this notation is always an odd number. This is illustrated with a 7-stage counter, for which a time sequence of states, as observed on the oscillo-

scope connected to one of the stages, may look like this:



The period consists of $2n = 14$ states. Describing this particular mode, the logic groups built by $n = 7$ elements can be written in three different ways: $(4 + 1 + 2)$, $(1 + 2 + 4)$, and $(2 + 4 + 1)$.

These mode notations are cyclic permutations. Hence they are equivalent and describe the same mode. The 7-stage counter could have another mode with the same set of logic groups. This different mode can be described by the following three equivalent mode notations: $(4 + 2 + 1)$, $(2 + 1 + 4)$, and $(1 + 4 + 2)$. If a certain wrong mode can exist, all possible permutations can exist also.

The correct mode always is the one with $x = 1$, that is, with one single logic group of size n. All other possible modes with $x \geqq 3$ are wrong modes.

IV. PREDICTION OF POSSIBLE MODES

4.1 *Possible Logic Groups*

Not all possible partitions of n into an odd number x of logic croups result in a possible mode, because there are some restrictions in possible logic group sizes g_j for the different counter types.

In single-phase counters, the logic groups can have any even or odd number of elements, up to n, since in any time slot, either a "1" or a "0" can be shifted from any stage to the following stage (Fig. 1). This is not so in double-phase counters.

In double-phase counters with an even number of stages (Fig. 2), a clock pulse A can shift either a "1" or a "0" to any odd-numbered

stage from the preceding stage, and a clock pulse B can shift either a "1" or a "0" to any even-numbered stage from the preceding stage. This results in the restriction that only logic groups with an even number of elements can appear in a possible mode.

In double-phase counters with an odd number of stages (Fig. 3), a clock pulse A can shift a "1" to any odd-numbered stage and a "0" to any even-numbered stage, and a clock pulse B can shift a "0" to any odd-numbered stage and a "1" to any even-numbered stage, always from the preceding stage. This results in the restriction that only logic groups with an odd number of elements can appear in a possible mode.

4.2 *Examples of Possible Modes*

We are now able to predict all possible modes of a twisted ring counter with n stages by breaking n into an odd number of logic groups in all possible ways, taking the restrictions of possible logic group sizes into account. This is shown in three examples.

Example 1: A single-phase counter with $n = 6$ stages can have six different possible modes:

(6)	correct mode
(4 + 1 + 1)	wrong modes.
(3 + 2 + 1)	
(3 + 1 + 2)	
(2 + 2 + 2)	
(2 + 1 + 1 + 1 + 1)	

In this counter type, the logic groups can have an even or odd number of elements.

Example 2: A double-phase counter with an even number of $n = 6$ stages (Fig. 2) has only two possible modes:

(6)	correct mode
(2 + 2 + 2)	wrong mode.

In this counter type, the logic groups can only have an even number of elements. Because of this restriction, there are always fewer wrong modes than in a single-phase counter with the same number of stages.

Example 3: A double-phase counter with an odd number of $n = 9$

stages has ten different possible modes:

(9)	correct mode
(7 + 1 + 1)	wrong modes.
(5 + 3 + 1)	
(5 + 1 + 3)	
(5 + 1 + 1 + 1 + 1)	
(3 + 3 + 3)	
(3 + 3 + 1 + 1 + 1)	
(3 + 1 + 3 + 1 + 1)	
(3 + 1 + 1 + 1 + 1 + 1 + 1)	
(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1)	

In this counter type, the logic groups can only have an odd number of elements. In general, the higher the number n of stages, the higher is the number of wrong modes.

4.3 *Experimental Verification of Predicted Modes*

Many counters of the three types shown in Figs. 1, 2, and 3 have been built, with various numbers of stages, and with different types of stages, including all types shown in Fig. 4. All of the predicted modes for these counters have actually been observed. Any desired mode can be induced by presetting all stages before turning the clock pulses on, but only the possible modes will be able to circulate without being altered.

V. SUPPRESSION OF WRONG MODES

All wrong modes can be suppressed by adding a certain small number of circuit connections. A general method for finding the necessary and sufficient additional connections for any twisted ring counter is to find criteria that are common to all wrong modes but do not appear in the correct mode. By suppressing these criteria, all wrong modes will be prevented. To find these common criteria, it is useful to define the concept of common logic groups.

5.1 *Common Logic Groups*

For a particular counter, the common logic groups represent the set consisting of the smallest logic groups (g_{min}) from each wrong

mode. For example, the 9-stage double-phase counter, whose wrong modes are listed in example 3 of Section 4.2, has two common logic groups of sizes 1 and 3. Each wrong mode contains at least one of the common logic groups. Taking the g_{min}-values of all wrong modes as common logic groups results in the smallest possible set of logic groups with the property of each wrong mode containing at least one of these logic groups.

The size of the smallest common logic group (m_k) is equal to the smallest g_{min}-value of all wrong modes, $g_{min\ min}$. That is

$g_{min\ min}$ = 1 for single-phase counters,
$g_{min\ min}$ = 1 for double-phase counters with odd number of stages, and
$g_{min\ min}$ = 2 for double-phase counters with even number of stages.

The size of the largest common logic group (m_0) is equal to the largest g_{min}-value of all wrong modes, that is, $g_{min\ max}$:

$$m_0 = g_{min\ max} \quad \text{of} \quad \sum_{i=1}^{x} g_i = n$$

with $x \geqq 3$ for wrong modes. Every possible partition of the above sum represents a possible wrong mode with a certain value g_{min}. The maximum of this value for all possible partitions is $g_{min\ max}$. It occurs with the minimum value of $x = 3$ and is

$$g_{min\ max} \leqq n/3.$$

The largest common logic group is therefore

$$m_0 \leqq n/3, \tag{1}$$

the next possible logic group size equal or less than $n/3$. This is

$$m_0 \geqq (n - 2)/3 \quad \text{for single-phase counters,} \tag{2}$$

$$m_0 \geqq (n - 4)/3 \quad \text{for double-phase counters.} \tag{3}$$

This results is only a single m_0-value in each case, when the restrictions of possible logic group sizes are taken into account. Combining the latter and expressions (1), (2), and (3) into a single expression, we get for the largest common logic group m_0:

$$m_0 = \frac{n}{3} - \frac{2p}{3} \cdot \Delta \qquad 0 \leqq \Delta \leqq 1 \tag{4}$$

with Δ chosen to make m_0 an integer, and

n = number of stages
p = 1 for single-phase counters

$p = 2$ for double-phase counters
$m_0 = 1, 2, 3, 4, \cdots$ for single-phase counters
$m_0 = 2, 4, 6, 8, \cdots$ if n = even } for double-phase counters.
$m_0 = 1, 3, 5, 7, \cdots$ if n = odd }

The set of common logic groups for a particular counter consists of the smallest and the largest common logic groups and all possible sizes of logic groups between. It is given in Table I for counters up to 20 stages. For single-phase counters with two stages and for double-phase counters with two or four stages there are no common logic groups, since these counters do not have any wrong mode.

5.2 *Suppressing the Common Logic Groups*

Suppressing all common logic groups in a counter leads, by definition, to the prevention of all possible wrong modes, and does not introduce any new modes. This section shows that there is a subset of common logic groups (Table II) whose suppression is sufficient for

TABLE I — COMMON LOGIC GROUPS

(Common logic groups are all different g_{min} values of all wrong modes)

Number of stages n	For single-phase counters	For double-phase counters	
		With even number of stages	With odd number of stages
2	—	—	
3	1		1
4	1	—	
5	1		1
6	2 1	2	
7	2 1		1
8	2 1	2	
9	3 2 1		3 1
10	3 2 1	2	
11	3 2 1		3 1
12	4 3 2 1	4 2	
13	4 3 2 1		3 1
14	4 3 2 1	4 2	
15	5 4 3 2 1		5 3 1
16	5 4 3 2 1	4 2	
17	5 4 3 2 1		5 3 1
18	6 5 4 3 2 1	6 4 2	
19	6 5 4 3 2 1		5 3 1
20	6 5 4 3 2 1	6 4 2	

suppressing all common logic groups and is thereby sufficient for preventing all wrong modes.

5.2.1 *Method of Suppressing a Group*

If we want to suppress a particular common logic group of size m_i, we must prevent one of the following two patterns consisting of an undesired sequence of ones and zeros

$$\cdots 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0 \cdots$$

$$\text{or} \quad (\cdots 1\ 1\ \underbrace{0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}_{m_i \text{ elements}}\ 1\ 1 \cdots)$$

from circulating around the counter ring. The inverse pattern, in parentheses, always appears with the first one. This suppression can be accomplished by preventing stage S_x from switching from "0" ("1") to "1" ("0") whenever stage S_{x-1-m_i} is in state "0" ("1"). The position of the patterns immediately before suppression is:

$$\begin{array}{cc} S_{x-1-m_i} & S_{x-1}\ S_x \\ \downarrow & \downarrow\downarrow \end{array}$$

$$\cdots 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0 \cdots$$

$$\text{or} \quad (\cdots 1\ 1\ \underbrace{0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}_{m_i \text{ elements}}\ 1\ 1 \cdots).$$

If stage S_x does not switch to "1" ("0") with the next clock pulse, the logic group of size m_i is prevented from passing through stage S_x. It is sufficient to suppress only one of the two patterns, since the inverse of it is then suppressed automatically.

This suppression can be implemented by adding a circuit connection from the output of stage S_{x-1-m_i} to the input of stage S_x, preventing S_x from switching from "0" to "1" whenever S_{x-1-m_i} is in state "0." This circuit connection, shown in Fig. 5a, bridges m_i stages, and therefore is called a "bridging connection"; its associated parameter m_i is called a "bridging parameter."

The bridging connection could also be made on the inverse side of the stages S_{x-1-m_i} and S_x, thus preventing S_x from switching from "1" to "0" whenever S_{x-1-m_i} is in state "1." These two bridging connections are equivalent, and one of them is sufficient. However, if both connections are applied for each m_i-value, a wrong mode is cleared within half a counter period instead of a full period.

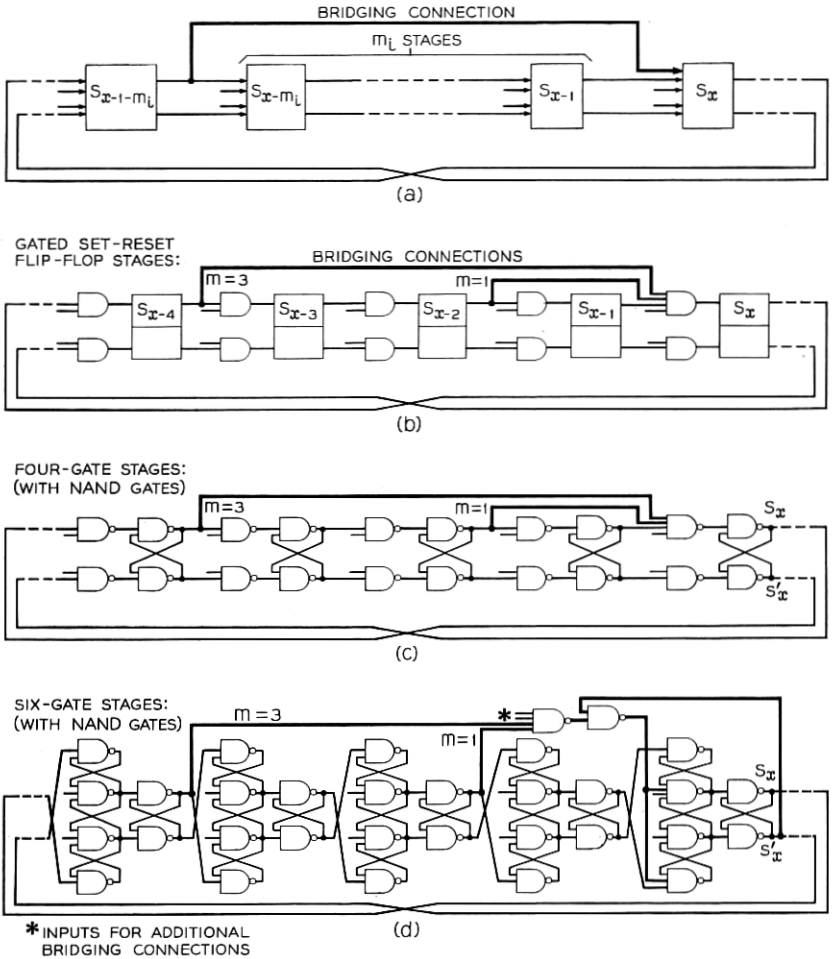


Fig. 5 — Suppression of wrong modes by adding bridging connections, bridging m_i stages. Part a shows the principle; b, c, and d show an example with the two bridging connections $m = 3$ and $m = 1$ for counters with different types of stages.

S_x may be any particular stage of the counter, but it should be the same stage for all bridging connections (although this is not essential with many counters). The correct mode is not affected by this inhibition, since in the correct mode S_{x-1-m_i} is always in state "1" ("0") when S_x is switched from "0" ("1") to "1" ("0") because m_i is always smaller than n.

5.2.2 *Automatically Eliminated Wrong Modes*

If we suppress one common logic group of size m_i by using the described method, we prevent the pattern

$$\cdots 0 \underbrace{\cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot}_{m_i} 1\ 0 \cdots$$

and hence eliminate not only the wrong modes containing the common logic group m_i, but also all other wrong modes that show this pattern at any one position.

The remaining wrong modes, which do not contain this pattern, require additional steps for their prevention. It can be shown that if a mode with $g_{min} < m_i$ satisfies both of the following two conditions, it does not contain the above pattern and therefore is not eliminated by suppressing m_i:

(*i*) All possible sums of the elements of an even number $2v$ of consecutive logic groups must be $\leqq m_i$ for at least one value of v ($v = 1$ or 2 or 3 $\cdots$).

(*ii*) All possible sums of the elements of an odd number $2v + 1$ of consecutive logic groups must be $\geqq m_i + 1$ for the same value of v that satisfies condition *i*.

Example: Suppose we have a single-phase counter with 19 stages, and we suppress the common logic group of size $m_i = 6$ by adding a bridging connection bridging 6 stages as shown in Fig. 5a. Would the mode (2 + 3 + 3 + 1 + 5 + 1 + 4) be suppressed?

We check whether this mode satisfies both conditions. Condition *i* is satisfied with $v = 1$, since all pairs of consecutive numbers in the mode notation (2 + 3, 3 + 3, 3 + 1, 1 + 5, 5 + 1, 1 + 4, 4 + 2) sum up to $\leqq 6$. That is, all sums of the elements of a pair ($2v$) of consecutive logic groups are $\leqq m_i$. Condition *i* could not be satisfied with $v > 1$ in this example. Condition *ii* is also satisfied with $v = 1$, since all triplets of consecutive numbers in the mode notation (2 + 3 + 3, 3 + 3 + 1, 3 + 1 + 5, 1 + 5 + 1, 5 + 1 + 4, 1 + 4 + 2, 4 + 2 + 3) sum up to $\geqq 7$. That is, all sums of the elements of a triplet ($2v + 1$) of consecutive logic groups are $\geqq m_i + 1$.

The above mode satisfies both conditions, and therefore would not be eliminated by suppression of the common logic group of size $m_i = 6$.

5.2.3 *Sufficient Subset*

Suppression of a particular common logic group of size m_i generally does not prevent wrong modes with $g_{min} > m_i$, but it does prevent some of the wrong modes with $g_{min} < m_i$. In the remaining unsuppressed modes with $g_{min} < m_i$, which all satisfy the two conditions stated in Section 5.2.2, the largest possible g_{min}-value, called $g_{min\ max}$, follows from condition i:

$$\sum_{j=1}^{2v} g_j \leqq m_i \qquad (v = 1 \text{ or } 2 \text{ or } 3 \cdots).$$

Every possible partition of this sum delivers a value g_{min}. The maximum of these g_{min}-values for all possible partitions is $g_{min\ max}$. It occurs with the minimum value of $v = 1$ and is

$$g_{min\ max} \leqq m_i/2.$$

This is the next lower common logic group size m_{i+1} that must be suppressed:

$$m_{i+1} \leqq m_i/2. \tag{5}$$

m_{i+1} is the next possible logic group size equal to or less than $m_i/2$, which is

$$m_{i+1} \geqq (m_i - 1)/2 \quad \text{for single-phase counters,} \tag{6}$$

$$m_{i+1} \geqq (m_i - 3)/2 \quad \text{for double-phase counters.} \tag{7}$$

This results in only a single m_{i+1}-value in each case, when the restrictions of possible logic group sizes are taken into account.

Combining the restrictions and the inequalities (5), (6), and (7) into a single expression, we get for the next lower common logic group m_{i+1} that must be suppressed:

$$m_{i+1} = \frac{m_i}{2} - (p - \tfrac{1}{2}) \cdot \Delta \qquad 0 \leqq \Delta \leqq 1 \tag{8}$$

with Δ chosen to make m_{i+1} an integer, and

$p = 1$	for single-phase counters
$p = 2$	for double-phase counters
$m_{i+1} = 1, 2, 3, 4, \cdots$	for single-phase counters
$m_{i+1} = 2, 4, 6, 8, \cdots$ if n = even $m_{i+1} = 1, 3, 5, 7, \cdots$ if n = odd	for double-phase counters.

If we suppress m_i, it is sufficient to supress m_{i+1} as the next lower common logic group, since suppression of m_i prevents all wrong modes with $m_i \geqq g_{min} > m_{i+1}$. Recursion formula (8) determines the maximum spacing of successive common logic group sizes m_i to be suppressed for sufficiently suppressing all common logic groups within the covered range. By extending this range from the largest common logic group m_0 to the smallest common logic group m_k, we get the sufficient subset of common logic groups

$$m_0 , m_1 , m_2 , \cdots , m_k$$

that must be suppressed for preventing all wrong modes. m_0 is determined by expression (4); m_1 through m_k are obtained by expression (8).

5.2.4 *Necessary Subset*

The m_i-values resulting from expressions (4) and (8)

$$m_0 , m_1 , m_2 , \cdots , m_i , \cdots , m_k$$

always represent a *sufficient* subset of common logic groups to be suppressed for preventing all wrong modes. But for some particular counters, the *necessary* subset $m_0 , m_1 , m_2 , \cdots , m_j$ may be smaller by a few m_i-values. That is, the smallest values $m_{j+1} \cdots m_k$ of the set are not necessary. There is not a simple expression like (4) and (8) for giving only the necessary m_i-values but, for a particular counter, they may be found by using the two conditions in Section 5.2.2, which have not yet been used to their full extent in Section 5.2.3. In a first step, the last value m_k is left off and a check is made whether any wrong mode exists that could satisfy both conditions for the remaining m_i-values. Such modes can be found by listing all possible combinations of logic groups that satisfy those two conditions (for $1 \leqq v \leqq m_i/2$). If there is no mode consisting entirely of these listed combinations, m_k is not necessary. In the next step, m_{k-1} is left off, repeating the procedure, until the last necessary value m_j is found.

For counters up to 20 stages, Table II gives the sufficient m_i-values (bridging parameters) according to expressions (4) and (8), with the unnecessary ones in parentheses.

5.3 *Implementation in Different Counter Circuits*

Each bridging parameter m_i denotes one bridging connection, bridging m_i stages, which has to be added to prevent wrong modes (as described in Section 5.2.1 and shown in Fig. 5a). Figures 5b, c, and d

TABLE II — BRIDGING PARAMETERS m_i

(Numbers without parentheses denote the necessary and sufficient bridging connections.)

Number of stages n	For single-phase counters m_i	For double-phase counters	
		With even number of stages m_i	With odd number of stages m_i
2	*	*	
3	1		1
4	1	*	
5	1		1
6	2 (1)	2	
7	2 1		1
8	2 (1)	2	
9	3 (1)		3 (1)
10	3 1	2	
11	3 (1)		3 (1)
12	4 2 (1)	4 (2)	
13	4 2 1		3 (1)
14	4 2 (1)	4 2	
15	5 2 (1)		5 (1)
16	5 2 (1)	4 (2)	
17	5 2 (1)		5 (1)
18	6 3 (1)	6 (2)	
19	6 3 1		5 (1)
20	6 3 (1)	6 2	

* No bridging parameters because these counters have no wrong modes.

show the bridging connections for the values $m = 3$ and $m = 1$ for counters with different types of stages. In counters with 6-gate stages, as shown in Fig. 5d, additional gates are required for proper suppression of common logic groups without impairment of the correct mode. For not impairing the correct mode, a feedback connection is required from the output of stage S_x. These counters need one additional gate if there is one bridging connection or two additional gates if there is more than one bridging connection.

As an example, we obtain for a 3-stage single-phase counter only one bridging parameter $m_0 = m_j = 1$. This means that only one bridging connection is needed, bridging one stage. Figure 6 shows three possible locations of the bridging connection. If bridging connections pass the twist, they must also be twisted, as illustrated in Figs. 6b and c.

For double-phase counters with an odd number of stages, one also has to make sure that the signal from stage S_{x-1-m_i} does not reach stage

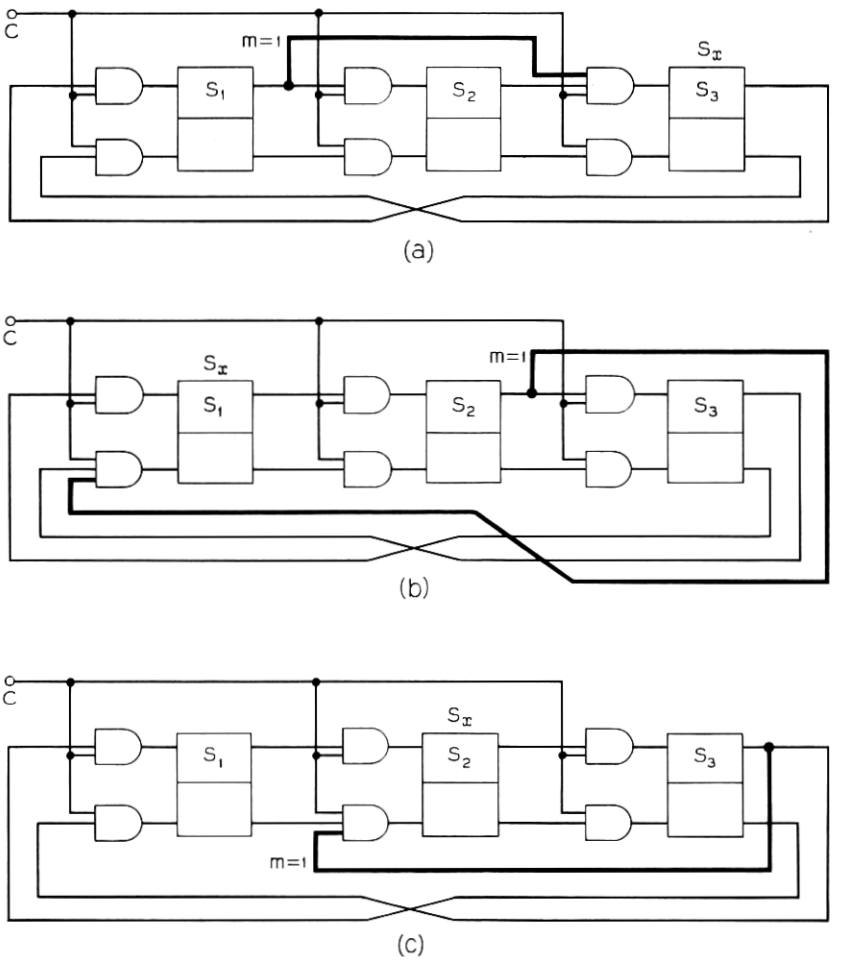


Fig. 6 — Single-phase counter with three stages requiring one bridging connection, $m = 1$. Parts a, b, and c are three equivalent solutions. If a bridging connection passes the twist, as in b and c, it must also be twisted.

S_x earlier than the signal from stage S_{x-1} caused by the same clock pulse. Otherwise a pattern $\cdots 1 + 1 + 1 + 1 + 1 \cdots$ might not be prevented under certain worst case propagation delays of the logic circuits involved. It is easy to assure this timing condition if logic gates are used that also provide a complementary output (as is the case in emitter-coupled gates). Figure 7 shows such an example with NOR/OR gates. In

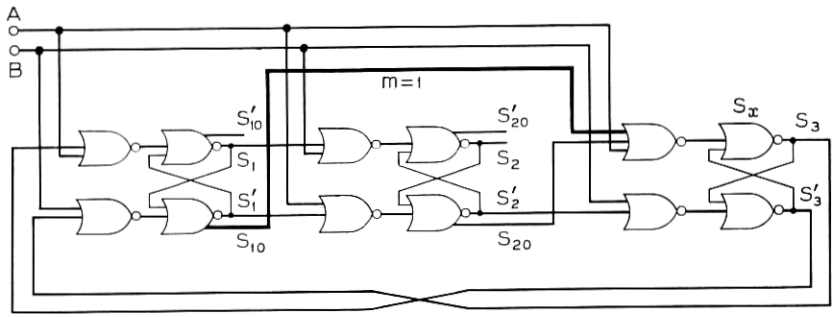


Fig. 7 — Double-phase counter with three stages (with NOR/OR gates). Use of complementary gate outputs (OR output S_{10} instead of NOR output S_1, and S_{20} instead of S_2) for increasing the permissible gate propagation delay tolerance range.

the case of a wrong mode (1 + 1 + 1) in this double-phase counter with three stages, output S_{10} appears one gate propagation delay later than S_1 upon an input pulse A, and output S_{20} one propagation delay earlier than S_2 upon the same input pulse A. This is sufficient to meet the above timing condition. If complementary gate-outputs are not available, a small delay may be introduced into the bridging connections. This additional timing condition does not exist in single-phase counters and in double-phase counters with an even number of stages.

5.4 *Experimental Verification*

Proper suppression of all wrong modes by bridging connections determined according to the described procedures has been verified experimentally with counters of all three types (Figs. 1, 2, and 3), with different stages (Fig. 4) and with many different values of n. Counters for which the necessary set of bridging connections is smaller than the sufficient set resulting from the formulas were given special attention.

5.5 *Summary: Suppression of Wrong Modes*

A small number of additional circuit connections (bridging connections) are sufficient for suppressing all wrong modes in a twisted ring counter. The bridging connections are determined by the bridging parameters m_i, which can be found by the formula:

$$m_0 = \frac{n}{3} - \frac{2p}{3} \cdot \Delta \qquad 0 \leqq \Delta \leqq 1$$

$$m_{i+1} = \frac{m_i}{2} - (p - \tfrac{1}{2}) \cdot \Delta$$

with Δ chosen to make m_0 and m_{i+1} integers, and

$$
\begin{aligned}
n &= \text{number of counter stages} \\
p &= 1 && \text{for single-phase counters} \\
p &= 2 && \text{for double-phase counters} \\
m_i &= 1, 2, 3, 4, \cdots && \text{for single-phase counters} \\
m_i &= 2, 4, 6, 8, \cdots \text{ if } n = \text{even} \\
m_i &= 1, 3, 5, 7, \cdots \text{ if } n = \text{odd} && \text{for double-phase counters} \\
i &= 0, 1, 2, 3, \cdots, j, \cdots, k.
\end{aligned}
$$

Each of the resulting bridging parameters m_i denotes one bridging connection in the circuit, which bridges m_i stages (Fig. 5). The bridging connection can be located anywhere in the counter ring; if it passes the twist, it must also be twisted. See Fig. 6.

The resulting $k + 1$ bridging parameters denote a sufficient set of $k + 1$ bridging connections in every case. For certain counters, however, the necessary set of $j + 1$ bridging connections is slightly smaller; it can be determined by the procedure described in Section 5.2.4.

Table II gives the $k + 1$ bridging parameters according to the above formula for different counter types up to 20 stages. The bridging parameters denoting unnecessary bridging connections according to the above procedure are in parentheses.

VI. CONCLUSION

Tools and methods for predicting and suppressing wrong modes in twisted ring counters have been developed. As a result we have gained a better insight into the multimoding mechanism and obtained a simple method for preventing multimoding. This method is summarized, and the required additional circuit connections are given in Table II for counters up to 20 stages.

VII. ACKNOWLEDGMENTS

I wish to thank Mr. D. Koehler for many helpful suggestions and his critical reading of the manuscript, and Mr. W. J. Kusner for experimentally verifying various results of this study.

REFERENCES

1. Ware, W. H., "The Logical Principles of a New Kind of Binary Counter," Proc. IRE, *41*, No. 10 (October 1953), pp. 1429–1437.

2. Tarczy-Hornoch, Z., "Five-Binary Counting Technique Makes Faster Decimal-Counting Units," Elec. Design, *9*, No. 2 (January 1961), pp. 34–37.
3. Millman, J., and Taub, H., *Pulse, Digital, and Switching Waveforms,* New York: McGraw-Hill Book Co., Inc., 1965, p. 682.
4. Boag, T. R., "Practical Applications of an IC Counter/Shift Register," Elec. Design News, *10*, No. 7 (June 1965), pp. 18–33.
5. "TTL Integrated Circuits: Counters and Shift Registers," Texas Instruments, Inc., Application Report SC 10660 (March 1968), pp. 28–29.