MIT/LCS/TM-47

MATHEMATICAL FOUNDATIONS OF FLIP - FLOPS

Vera Pless

June 1974

TM-47

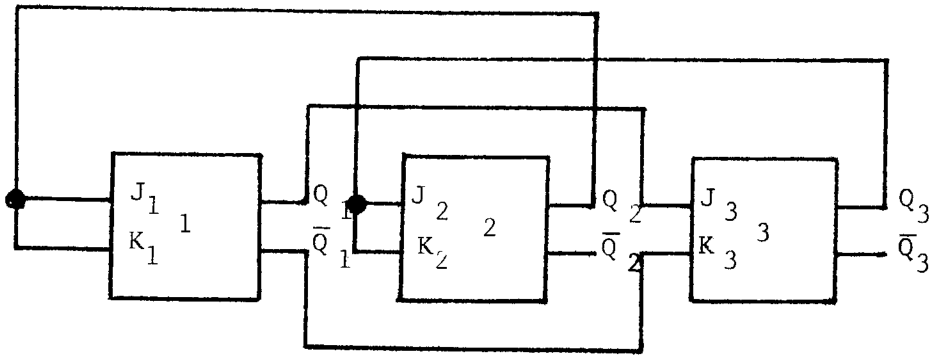
Mathematical Foundations of Flip-Flops

by Vera Pless

Section 1 - Introduction

My interest in this topic was aroused by conversations with Edward Fredkin and Frank Manning. I found Frank Manning's thesis [2] on this topic a useful reference. The main purpose of this paper is to lay a mathematical basis for the study of flip-flops. I feel that this approach will lead, in further studies, to important practical results although the alert reader can see some practical applications in this work.

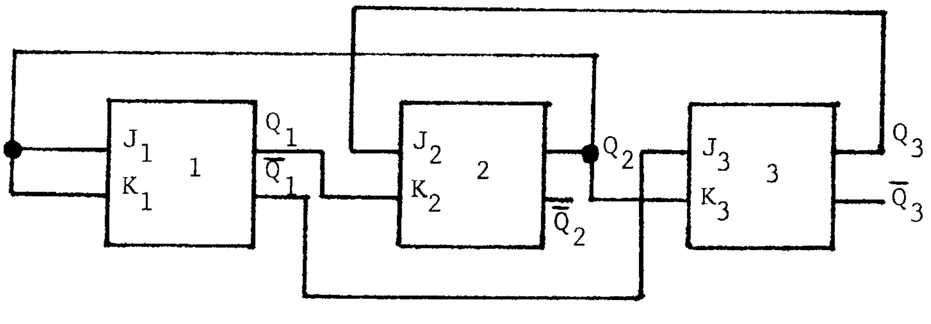
A J-K flip-flop is a device with 2 inputs (zero or one) and two outputs, one of which is always the complement of the other. A counter is a set of n J-K flip-flops whose inputs are either constants (0 or 1) or outputs of other flip-flops in the counter. In this paper we only consider J-K flip-flops whose inputs are outputs of the other flip-flops. Whenever we refer to an n-counter, we shall mean such a counter with no constant inputs. As in Manning's paper we do not consider flip-flops whose inputs arise from their own outputs. The specification of the connections is called the connection list (abbreviated (CL) of the counter. Consider the following example when n = 3.



J_i and K_i are the inputs to flip-flop i, and Q_i and $\bar{Q}_i$ (the complement of Q_i) are its outputs. In this example Q_2 is connected to J_1 and K_1, Q_3 is connected to J_2 and K_2, and Q_1 is connected to J_3 while $\bar{Q}_1$ is connected to K_3. For brevity we represent this connection list as follows:

$$\begin{array}{ccc} 1 & 2 & 3 \\ \hline (2,2) & (3,3) & (1,\bar{1}) \end{array}$$

The numbers on the first line represent the flip-flops i while the pairs below describe the counters connected to J_i and K_i. Clearly it is possible to translate a description like this into a diagram as above or to go from a diagram to such a shortened description. The first of a pair of components refers to the J input, the second to the K input. We call this description a connection list. For example, the connection list $\begin{array}{ccc} 1 & 2 & 3 \\ \hline (2,2) & (3,1) & (\bar{1},2) \end{array}$ describes the following 3-counter.



After a clock pulse, the outputs of each flip-flop change depending on the previous outputs and the inputs according to the following rules.

Input (0,0) leaves the output unchanged.

Input (1,1) changes output.

Input (0,1) produces a 0 output.

Input (1,0) produces a 1 output.

Thus if the 3-counter is $\frac{1}{(2,2)} \quad \frac{2}{(3,3)} \quad \frac{3}{(\bar{1},1)}$ and the first output is (0,0,0), the sequence of outputs and consequent inputs is as follows:

0	0	0
(0,0)	(0,0)	(1,0)
0	0	1
(0,0)	(1,1)	(1,0)
0	1	1
(1,1)	(1,1)	(1,0)
1	0	1
(0,0)	(1,1)	(0,1)
1	1	0
(1,1)	(0,0)	(0,1)
0	1	0
(1,1)	(0,0)	(1,0)
1	1	1
(1,1)	(1,1)	(0,1)
0	0	0

The triples are the outputs of this 3-counter in the order in which they are generated. The pairs beneath an output are the next input sequences, i.e., the next set of (J_i,K_i), $i = 1,2,3$.

From now on when we speak of an n-counter we shall mean one with a fixed connection list. In section 2 the number of n-counters is shown to be $[2n - 2]^{2n}$. Also in section 2 we associate a matrix M with an n-counter. Given any output sequence S, S determines the next input sequence. Here we show that this input sequence is actually $\underline{S}M$ where $\underline{S}$ is a vector determined by S. In addition we give an algorithm (2.4) for determining which, if any, n-counter can produce a fixed sequence of outputs. M is easily determined by the connection list of an n-counter and vice versa.

In section 3 we give a simple algebraic transformation N which operates on outputs and produces the next output from a given output. N is easily determined by M and conversely. For the example above, the transformation N is $\frac{a}{(a + b)} \quad \frac{b}{(b + c)} \quad \frac{c}{\bar{a}}$. All additions are mod 2. Applying this transformation to 000 yields the same output sequence as before. Each output is computed from the previous one by the above output rule. Previously, we determined this sequence of outputs by using the inputs and the rules for obtaining an output from a given input. Using N we can go directly from one output to the next output.

0	0	0
0	0	1
0	1	1
1	0	1
1	1	0
0	1	0
1	1	1
0	0	0

Simple rules for going from M to N and back again are given. Conditions under which N can be represented by a matrix are given. The complete form of a transformation N which correspond to the action of an n-counter is determined. Algorithm 3.2, using the N transformation, decides which, if any, n-counters produce a given sequence of outputs.

In section 4 we describe the directed graph of an n-counter. The vertices of this graph are the 2^n possible outputs with an edge going from vertex S_i to vertex S_j if S_i exactly preceeds S_j. An n-counter is said to be cyclic if its graph is the union of disjoint cycles. In other words, an n-counter is cyclic if starting from any output, there is a sequence of outputs which returns to it. It is shown that an n-counter is cyclic if and only if there is an i so that $N^i = I$ where I is the identity transformation. Special results for the case when N is a permutation matrix are given. For any n-counter, if k is the l.c.m. of the cycle lengths then there is a geometrically determined non-negative integer j so that k is the smallest positive integer with the property that $N^{k+j} = N^j$. For a cyclic counter, $j = 0$.

In section 5 we relate an n-counter to a group H. H is generated by all permutations on the n objects $a_1, \ldots, a_n$ (with simultaneous action on $\bar{a}_1, \ldots, \bar{a}_n$) and the permutations interchanging a_i and $\bar{a}_i$, $i = 1, \ldots, n$. The order of H is $(n!)2^n$. The subgroup of H leaving the N transformation of an n-counter C invariant is called the group G of C. Each element in G is an isomorphism of the graph of C. Two n-counters are called equivalent if there is an element in H sending one onto the other. Equivalent n-counters have isomorphic graphs and isomorphic groups. The number of counters equivalent to C equals $n!\ 2^n$ divided by the order of G.

Since equivalent n-counters have the same properties, if one were to classify all n-counters for a fixed n it is sufficient to do this by equivalence classes. This procedure is illustrated in section 6 for all 2-counters. The sixteen 2-counters fall into 4 equivalence classes, three of them cyclic.

Section 2 The M-matrix

Suppose we have an n-counter C whose connection list is the following: $\frac{1}{(a_1,a_1')}$ $\frac{2}{(a_2,a'_2)}$ ..... $\frac{n}{(a_n,a_n')}$. Here each a_i or a_i' is some j or $\bar{j}$. Let M be the (2n) x (2n) matrix whose rows and columns are marked 1, 1', 2, 2', ...,n, n', and whose entries are zero and the following ones.

1 in row a_i and column i if $a_i = j$

1 in row a_i and column i' if $a_i' = j$

1 in row a_i' and column i if $a_i = \bar{j}$

1 in row a_i' and column i' if $a_i' = \bar{j}$.

An M-matrix has exactly one one in any column. Further it has zeros on all 2 x 2 diagonal blocks (since we do not allow the inputs of a J-K counter to come from its own outputs). There is clearly a one to one correspondence between n-counters and 2n x 2n matrices M whose entries are 0 and 1, which have exactly one one in each column, and whose 2 x 2 diagonal blocks consist entirely of zeros. From this 1:1 correspondence we see that there are $(2n - 2)^{2n}$ n-counters.

As examples we list the M-matrices for the two connection lists which appeared as examples in sections 1.

Example 2.1

$\frac{1}{(2,2)}$ $\frac{2}{(3,1)}$ $\frac{3}{(\bar{1},2)}$

$$M = \begin{array}{c|cc|cc|cc|}
 & 1 & 1' & 2 & 2' & 3 & 3' \\
1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1' & 0 & 0 & 0 & 0 & 1 & 0 \\
\hline
2 & 1 & 1 & 0 & 0 & 0 & 1 \\
2' & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
3 & 0 & 0 & 1 & 0 & 0 & 0 \\
3' & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

Example 2.2

$$\begin{array}{ccc} 1 & 2 & 3 \\ \hline (2,2) & (3,3) & (\bar{1},1) \end{array}$$

$$M = \begin{array}{c|cc|cc|cc|} & 1 & 1' & 2 & 2' & 3 & 3' \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1' & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2' & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 3 & 0 & 0 & 1 & 1 & 0 & 0 \\ 3' & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Any output S of a given n-counter C is an n-tuple of zeros and ones. To this n-tuple associate a 2n-tuple of zeros and ones $\underline{S}$ by replacing every zero by the pair 0,1 and every 1 by the pair 1,0. The next theorem tells us how we obtain the next set of connection pairs from S and M.

Theorem 2.3. Let C be a fixed n-counter and let M be its M-matrix. If S is an output of C, then the next set of input pairs, called P, is equal to $\underline{S}$ M.

Proof: The theorem follows by the definitions of the various elements in it.

In section 1, the outputs and sets of connection pairs of example 2.2 were given. Each set of connection pairs can be found from the immediately preceeding output by the theorem. For example, given the output S = 001, we construct the next set of connection pairs as follows. $\underline{S}$ = (0,1,0,1,1,0) so that

$$\underline{S}\ M = (0,1,0,1,1,0) \left|\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right| = (0,0,1,1,1,0) = (0,0)\ (1,1)\ (1,0)$$

One of the important consequences of this theorem is algorithm 2.4 which determines the n-counter C which could produce a fixed set of outputs. Given a set of k outputs in a fixed order (i.e. k n-tuples of 0's and 1's, $S_1,\dots,S_k$) then there are the following three possibilities.

1) There is no n-counter which produces $S_1,\dots,S_k$ as a set of outputs.

2) There is exactly one n-counter which produces $S_1,\dots,S_k$ as a set of outputs.

3) There is more than one n-counter which produces $S_1,\dots,S_k$ as a set of outputs.

Algorithm 2.4 distinguishes between these three cases and in cases 2 and 3 give the specific n-counter or n-counters which produce the given sequence.

Algorithm 2.4.

Given $S_1,\dots,S_k$, k n-tuples of 0's and 1's, this algorithm gives a procedure for determining all n-counters (if any exist) which can produce this sequence of outputs.

Let $S_i = (\alpha_1,\dots\alpha_n)$ and $S_{i+1} = (\beta_1,\dots,\beta_n)$ then there are exactly 2 input pairs which take any α_j to β_j. If $\alpha_j = \beta_j = 0$ these are (0,0) and (0,1). If $\alpha_j = \beta_j = 1$, these are (0,0) and (1,0). If $\alpha_j = 0$ and $\beta_j = 1$, these are (1,1) and (1,0), while if $\alpha_j = 1$ and $\beta_j = 0$, these are (1,1) and (0,1).

Let $\underline{S}_1,\dots,\underline{S}_k$ be the k (2n)-tuples of 0's and 1's associated with $S_1,\dots,S_k$. We start with a partially filled in M and compute $\underline{S}_i$ M. As we explain below this procedure determines other entries in M. We stop when we have computed all $\underline{S}_i M$, $i = 1,\dots,k-1$. We then know that either M cannot exist (so no n-counter could produce the sequence of outputs), or exactly which matrices M could realize this sequence.

We start with placing 2 x 2 blocks of zeroes on the diagonal of M. As we proceed we bear in mind that each column of M must have exactly one one; the other entries are 0. So if we put a one in any column, we can make the other entries 0. We know that $\underline{S}_i$ M equals the set of input pairs which send $S_i = (\alpha_1,\ldots,\alpha_n)$ into $S_{i+1} = (\beta_1,\ldots,\beta_n)$. For each α_j and β_j there are two possible input pairs which send α_j into β_j. These are given above. Notice that in each case these pairs have a common component. Hence the product $\underline{S}_i$ M (given that S_{i+1} is the next output) produces constraints on n of the 2n columns of M. We fill in these columns to satisfy these constraints. If we cannot, then M cannot exist. When we have finished computing $\underline{S}_i$ M, i = 1, ..., k - 1, we have either determined

1) M cannot exist. This happens when the construction of M based on outputs $S_1,\ldots,S_i$ cannot possibly send S_i into $S_{(i+1)}$.

2) A specific M has been constructed (all blanks have been filled in and M sends each S_i into a set of input pairs which produces S_{i+1}).

3) More than one M is possible. This happens when no contradiction occurs but when then are still a number of blanks in M. These can then be filled in in any way providing each column of M has exactly one 1. Each of these ways corresponds to a different n-counter.

Another algorithm (3.2) based on the N-transformation is given in section 3. Both algorithms accomplish the same task.

Example 2.5: We give an example of the use of algorithm (2.4). Assume we have the following sequence of outputs.

000,	011,	110,	100,	111,	001,	000.
S_1	S_2	S_3	S_4	S_5	S_6	S_7

The first partial M is

$$\begin{array}{|cc|cc|cc|} 0&0&&&&\\ 0&0&&&&\\ \hline &&0&0&&\\ &&0&0&&\\ \hline &&&&0&0\\ &&&&0&0 \end{array}$$

$\underline{S}_1 = (0,1,0,1,0,1)$. In computing $\underline{S}_1$ M we write below M the 3 sets of 2 possible input pairs. We use this to add entries to M.

$$(0,\ 1,\ 0,\ 1,\ 0,\ 1)\ \begin{array}{|cc|cc|cc|} 0&0&0&&0&\\ 0&0&&&&\\ \hline &&0&0&0&\\ 0&&0&0&&\\ \hline &&0&&0&0\\ &&&&0&0\\ \hline 0&0&1&1&1&1\\ \hline 0&1&1&0&1&0\\ \checkmark&&\checkmark&&\checkmark& \end{array}$$

Possible input pairs to send 0_1 into 0_2.

The checked columns are the ones where the connection pairs have a common entry. We are able to make entries into these columns, and these entries are shown. We continue in this fashion.

$$\underline{S}_2 M = (0,1,1,0,1,0)\ \begin{array}{|cc|cc|cc|} 0&0&0&&0&0\\ 0&0&&0&&\\ \hline &&0&0&0&\\ 0&&0&0&&0\\ \hline &&0&0&0&0\\ 0&&&&0&0\\ \hline 1&1&0&0&1&1\\ \hline 1&0&1&0&0&1\\ \checkmark&&&\checkmark&&\checkmark \end{array}$$

possible input pairs.

$\underline{S}_3$ M: (1, 0, 1, 0, 0, 1)

0	0	0		0	0	
0	0		0			
	0	0	0	0		
0		0	0		0	
		0	0	0	0	
0	0			0	0	
0	0	1	1	0	0	possible input pairs
1	0	0	1	0	1	
	✓		✓	✓		

$\underline{S}_4$ M: (1,0,0,1,0,1)

0	0	0		0	0	
0	0	0	0	0		
	0	0	0	0		
0	0	0	0	1	0	
	1	0	0	0	0	
0	0	1		0	0	
0	0	1	1	1	1	possible input pairs
1	0	1	0	1	0	
	✓	✓		✓		

$\underline{S}_5$ M: (1,0,1,0,1,0)

0	0	0	1	0	0	
0	0	0	0	0	1	
	0	0	0	0	0	
0	0	0	0	1	0	
	1	0	0	0	0	
0	0	1	0	0	0	
1	1	1	1	0	0	possible input pairs
0	1	0	1	1	0	
	✓		✓		✓	

$\underline{S}_6$ M: (0,1,0,1,1,0)

0	0	0	1	0	0	
0	0	0	0	0	1	
1	0	0	0	0	0	
0	0	0	0	1	0	
0	1	0	0	0	0	
0	0	1	0	0	0	
0	0	0	0	1	1	possible input pairs
0	1	0	1	0	1	
✓		✓			✓	

Hence M is unique and is

$$\begin{array}{cc|cc|cc}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
\hline
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0
\end{array}$$

From M we read off its 3 counter.

1	2	3
(2,3)	$(\bar{3},1)$	$(\bar{2},\bar{1})$

Note that if we had asked for a 3-counter which has S_1 through S_6 as successive outputs (with no requirements about the successor of S_6) then we would not only have the M above but also

$$M_1 = \left|\begin{array}{cc|cc|cc}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
\hline
1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0
\end{array}\right|$$

so that there would be 2 possible 3-counters, the one above and the following one corresponding to M_1:

1	2	3
(3,3)	$(\bar{3},1)$	$(\bar{2},\bar{1})$

This 3-counter takes S_1 to S_2, S_2 to S_3, S_3 to S_4, S_4 to S_5, S_5 to S_6 and S_6 to 100.

Section 3. The Transformation N

In this section we show that given any n-counter C there is a transformation N which takes any output S_i and produces the next output $S_{i+1.}$ In other words if $S_i = (\alpha_1,\ldots,\alpha_n)$, then $S_{i+1} = (\beta_1,\ldots,\beta_n)$ where each β_i can be expressed as a certain set of sums and products (mod 2) of specified α's. The rule for obtaining $S_i + 1$ from S_i is the N transformation. It is surprising that given an n-counter such an N-transformation exists. We will demonstrate how to obtain N from M and conversely. This is quite easy. The complete form of any such N will be given. N is not always a linear transformation as M is but it sometimes is linear. It is fairly obvious from properties of M whether or not N will be linear.

In order to determine N we assume we have an n-counter C and an output $S_i = (\alpha_1,\ldots,\alpha_n)$ where each α_i is 0 or 1. With each α_i is associated an input pair (a_i,a_i'). Let $S_{i+1} = (\beta_1,\ldots,\beta_n)$. Then the i^{th} component of the next output, β_i, is determined by α_i and the pair (a_i,a'_i) as follows .

Identity 3.1 $\beta_i = (\alpha_i\, a_i' + (1+\alpha_i)\, a_i + \alpha_i)$ (mod 2)

Identity 3.1 can be verified computationally for each specific possiblity. This identity reflects the fact that the rules governing outputs in a J-K flip flop act like an affine transformation followed by a dot product.

Using identity 3.1 we can determine the form of β_i for the three different types of connection pairs (a_i, a_i').

Case I. a) $(a_i, a_i') = (j, j)$ b) $(a_i, a_i') = (\bar{j}, \bar{j})$

In case a), $\beta_i = (\alpha_i j + (1 + \alpha_i) j + \alpha_i) \pmod 2$

$= (\alpha_i + j) \pmod 2$

Similarly, in case b) $\beta_i = (\alpha_i + \bar{j})$.

Case II. a) $(a_i, a_i') = (j, \bar{j})$ b) $(a_i, a_i') = (\bar{j}, j)$

In case a), $\beta_i = (\alpha_i \bar{j} + (1 + \alpha_i) j + \alpha_i) \pmod 2$

$= (\alpha_i (1 + j) + (1 + \alpha_i) j + \alpha_i) \pmod 2$

$= j$

Similarly, in case b) $\beta_i = \bar{j}$

Case III. a) $(a_i, a_i') = (j, k)$, b) $(a_i, a_i') = (\bar{j}, k)$,

c) $(a_i, a_i') = (j, \bar{k})$ d) $(a_i, a_i') = (\bar{j}, \bar{k})$

For case a) $\beta_i = (\alpha_i k + (1 + \alpha_i) j + \alpha_i) \pmod 2$

$= \alpha_i (j + k) + j + \alpha_i \pmod 2$

Similarly, in case c) $\beta_i = \alpha_i (j + \bar{k}) + j + \alpha_i = \alpha_i (j + k) + j$

In case d) $\beta_i = \alpha_i (\bar{j} + \bar{k}) + \bar{j} + \alpha_i = \alpha_i (j + k) + \alpha_i + j + 1$

If $S_i = (\alpha_1, \ldots, \alpha_n)$ and the next output $S_{i+1} = (\beta_1, \ldots, \beta_n)$ then β_i is always obtained from α_i by a transformation of the form given in Case I or Case II or Case III. We call N which consists of these n transformations of α_i into β_i, the transformation of the n-counter C. If all the n transformations in N fall under cases I or II, then N is a linear transformation and its matrix can be obtained from the matrix of M by putting 1's on the diagonal, for any column pairs i, i', whenever case I occurs. If all the n transformations in N fall under case II, we call N a permutation and the matrix of N is M. If in addition, N sends distinct a_i into distinct j (or their complements), then we call N a strict permutation. The matrix of a strict permutation has exactly one non-zero element in any row or column and so represents a permutation on 2n objects. Of the $(2n-2)^{2n}$ n-counters $[4(n-1)]^n$ are linear and $[2(n-1)]^n$ of the linear ones are permutations. It is clear that in all three cases the transformation N can be easily obtained from the connection list or the matrix M and conversely. We will illustrate this by examples.

Example 2.1

1	2	3
(2,2)	(3,1)	$(\bar{1},2)$

has its N transformation as follows.

a	b	c
a+b	b(c+a)+c+b	$c(a+b)+\bar{a}$

This is non-linear since the b and c images are non-linear.

To obtain any output from a given one, we merely apply these rules modulo 2, for example if, S_1 = 000, S_2 = 001 where the first 0 is $0 + 0(a + b)$, the second 0 is $0(0 + 0) + 0\ (b(c + a) + c + b)$, and the 1 is $0(0 + 0) + 1\ (c(a + b) + \bar{a})$. Clearly S_3 = 0 1 1 etc.

Example 2.2

1	2	3
(2,2)	(3,3)	$(\bar{1},1)$

Its N transformation is

a	b	c
a + b	b + c	$\bar{a}$

and is clearly linear although not a permutation since the transformations on a and b are not permutations.

$$M = \left(\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \quad \text{and } N = \left(\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

To obtain the succeeding output from a given output S, one can apply the rule given by N or else compute $\underline{S}$ N. For instance, if $S_1 = 0\ 0\ 0$, then $\underline{S}_1$ = (0, 1, 0, 1, 0, 1) and

$$\underline{S}_1 N = (0, 1, 0, 1, 0, 1)\left(\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) = (0, 1, 0, 1, 1, 0)$$ which means $S_2 = 0\ 0\ 1$.

We now give algorithm 3.2 based on the N transformation. As the previous algorithm based on the M matrix does, this algorithm determines the possible, if any, n-counters which can generate a given sequence of outputs.

Algorithm 3.2

Suppose we have a sequence of k outputs $S_1,\ldots,S_k$ which we presume to be generated by an n-counter. Then knowing the form that N can take, we can solve n(k - 1) equations for the unknown parameters. In other words, if $S_i = (a_1,\ldots, a_n)$ and $S_{i+1} = (b_1,\ldots,b_n)$ then b_j is a linear combination, with coefficients 0 and 1 of $1,a_1,\ldots,a_n$ and possibly a term in a_j times the sum of a pair from the set $\{a_1,\ldots,a_{j-1},a_{j+1},\ldots,a_n\}$. Hence there are $(n + 1) + \binom{n-1}{2}$ coefficients to determine and n(k - 1) equations. In addition, these coefficients must satisfy certain constraints, namely, there is exactly one non-zero coefficient of an a_k (for $k \neq j$) and at most one non-zero coefficient of a_j times $a_k + a_\ell$ $(k, \ell \neq j)$.

We illustrate this method by the sequence of outputs 000, 011, 100, 111, 001, 000 (this is the same example we used for algorithm 2.4). The general form for a 3-counter is given below. At the left side we list the outputs. On the same line with an output is the 3 equations it determines. The constraints mentioned above imply the three equations. $\alpha_3 + \alpha_4 = 1$, $\beta_2 + \beta_4 = 1$, and $\gamma_2 + \gamma_3 = 1$.

0 0 0	$\alpha_1 a(b+c)+\alpha_2 a+\alpha_3 b+\alpha_4 c+\alpha_5$	$\beta_1 b(a+c)+\beta_2 a+\beta_3 b+\beta_4 c+\beta_5$	$\gamma_1 c(a+b)+\gamma_2 a+\gamma_3 b+\gamma_4 c+$ γ_5
0 1 1	$0 = \alpha_5$	$1 = \beta_5$	$1 = \gamma_5$
1 1 0	$1 = \alpha_3+ \alpha_4$ known	$1 = \beta_1+\beta_3+\beta_4+1$	$0 = \gamma_1+\gamma_3+\gamma_4+1$
1 0 0	$1 = \alpha_1+\alpha_2+ \alpha_3$	$0 = 1+\beta_2+\beta_3+1$	$0 = \gamma_2+\gamma_3+1$; known
1 1 1	$1 = \alpha_2$	$1 = \beta_2+1$; $\beta_2=0$ Hence $\beta_3 = 0$, $\beta_4 = 1$, $\beta_1 = 1$ Hence this transformation is determined as $\frac{b}{b(a+c)+c+1}$	$1=\gamma_2+1$; $\gamma_2 = 0$ Hence $\gamma_3 = 1$
0 0 1	$0 = 1 + \alpha_3 + \alpha_4$ known	$0 = 1(1+1)+1+1$ checks	$1=1+\gamma_4+1$: $\gamma_4=1$ and $\gamma_1=1$

0 0 0	$0 = \alpha_4$ Hence $\alpha_3 = \alpha_1 = 1$ and this transformation is $\dfrac{a}{a(b + c) + a + b}$	$0 = 0(0+1)+1+1$ checks	Hence this transformation is determined as $\dfrac{c}{c(a+b)+b+c+1}$ $0 = 1(0+0)+0+1+1$ checks

As does algorithm (2.4), algorithm (3.2) determines that a) no counter can generate the sequence of outputs, b) exactly one counter can and c) more than one counter can. In case c), some indeterminates will not be determined. They can be specified in any way consistent with the constraints (i.e. in looking for the rule taking a_j to its successor, there is exactly one non-zero coefficient of an a_k $(k \neq j)$ and at most one non-zero coefficient of a_j times $a_k + a_\ell$ $(k, \ell \neq j)$.

Section 4

The Graph of an n-Counter

If C is an n-counter then the action of C on an output is given by an N transformation of a certain form. Given such an N, we define the transformation N^i in the usual way, i.e., N^i is defined inductively on any sequence S of zeroes and ones by $N^i(S) = N(N^{i-1}(S))$. If N is of the form which gives rise to an n-counter, it does not follow that N^i is of that form, it may be or it may not be. However, N^i must be of the following general form. If $S = (a_1,\ldots,a_n)$ (all a_i's are 0 or 1) and $N^i(S) = S' = (b_1,\ldots,b_n)$ (again all b_i are 0 or 1), then the transformation taking a_i to b_i is of the following form: b_i is a linear combination of the $(2^n - 1)$ products of $a_1,\ldots,a_n$ and β can be 0 or 1. One transformation of this form is the identity transformation, denoted by I, where $I(S) = S$ for any S. I does not give rise to an n-counter, however the existence of an i such that $N^i = I$ gives important information about an n-counter as we see below.

Let C be an n-counter. We associate with C a directed graph R whose vertices are the 2^n n-tuples of 0's and 1's. We draw an arrow from a vertex S_i to another S_{i+1} if S_{i+1} is the output directly succeeding S_i. We say that the counter C has an s-cycle if R has a cycle of length ^{s}. We illustrate this by the examples below.

Example 4.1

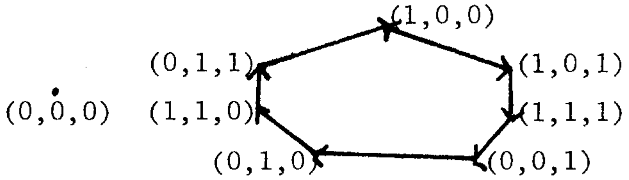
$$M = \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

N is

a	b	c
(a+b)	(b+c)	a

$N^7 = I$

R:



C has a 7-cycle and a 1-cycle

N sends (0,0,0) onto itself, and we call it a 1-cycle.

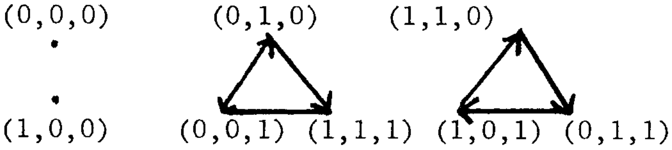
Example 4.2

$$M = \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

N is

a	b	c
(a+b)	(b+c)	b

$N^3 = I$



C has 2 3-cycles and 2 1-cycles

Example 4.3

$$M = \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

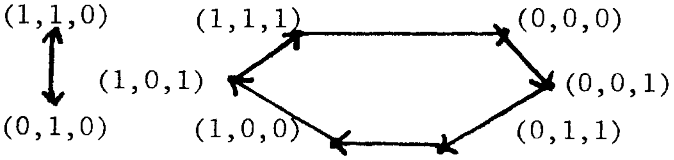
N is

a	b	c
(a+b)	(b+c)	$\bar{b}$

$N^6 = I$

C has a 6-cycle and a 2-cycle.

R:



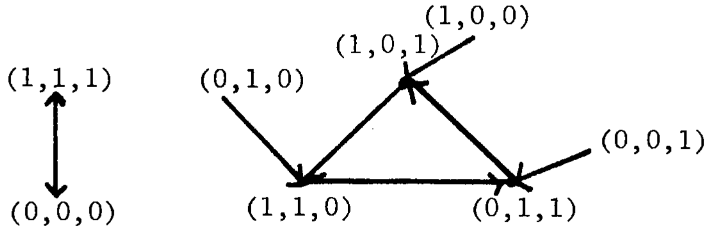
Example 4.4

$$M = \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

N is

a	b	c
(a+b)	(b+c)	(a+c)

$N^4 = N$



C has a 3-cycle and a 2-cycle.

An n counter is called cyclic if given any sequence S, there is a set of succeeding outputs $S, S_1, \ldots, S_k = S$ returning to S. In terms of the graph of C, an n counter is cyclic if and only if its graph is the union of disjoint cycles. Theorem 4.6 demonstrates the very interesting fact that an n-counter is cyclic iff there exists an i such that $N^i = I$. In order to prove this theorem we need Lemma 4.5.

First some notation.

Let $a_1, \ldots, a_n$ be n variables which assume the values 0 or 1. Let $\Omega(a_1, \ldots, a_n)$ be the set of all distinct $(2^n - 1)$ products of $a_1, \ldots, a_n$ and the number 1. We define a form on $(a_1, \ldots, a_n)$ to be a linear combination of elements of $\Omega(a_1, \ldots, a_n)$ with coefficients 0 or 1. If L is a form on $(a_1, \ldots, a_n)$ and S is an n-tuple of 0's and 1's, then L(S) is the value (either 0 or 1) which L assumes (mod 2) when the ith component of S is substituted for a_i for all i between 1 and n. An example of a form on (a_1, a_2) is $a_1 a_2 + a_2 + 1$. If S = (0,1), L(S) = 0.1 + 1 + 1 = 0 (mod 2).

Lemma 4.5

If L_1 and L_2 are two forms on $(a_1,\ldots,a_n)$ and $L_1(S) = L_2(S)$ for any n-typle S of 0's and 1's, then L_1 is identically equal to L_2.

Proof: The proof is by induction on n. If $n = 1$, $L_1 = \beta_1 a_1 + \beta_2$ where $\beta_1 \beta_2 \epsilon \{0,1\}$. Also $L_2 = \gamma_1 a_1 + \gamma_2$ for $\gamma_1, \gamma_2 \epsilon \{0,1\}$. If $S = 0$, $L_1(S) = \beta_2 = L_2(S) = \gamma_2$. If $S = 1$, $L_1(S) = \beta_1 + \beta_2 = L_2(S) = \gamma_1 + \gamma_2$ and since $\beta_2 = \gamma_2$, $\beta_1 = \gamma_1$.

Assume the lemma is true for n-1. We want to prove it true for n. L_1 and L_2 are two forms on $(a_1,\ldots,a_n)$ and $L_1(S) = L_2(S)$ for any n-tuple S of 0's and 1's by assumption. Now L_1 can be expressed as some $K_1 + K_2 a_n$ where K_1 and K_2 are forms on $(a_1,\ldots,a_{n-1})$. Also $L_2 = M_1 + M_2 a_n$ where M_1 and M_2 are forms on $(a_1,\ldots,a_{n-1})$. Let S be an arbitrary n-tuple whose n^{th} component is 0 and let S' be the (n-1) tuple which agrees with S on its first (n-1) components. Then $L_1(S) = K_1(S')$ and $L_2(S) = M_1(S')$. But S' is an arbitrary (n-1)-tuple and $L_1(S) = L_2(S)$ implies $K_1(S') = M_1(S')$ so that by the induction assumption K_1 is identically equal to M_1. Now let S be an arbitrary n-tuple whose nth component is 1 and let S' the (n-1)-tuple which agrees with S on its first (n-1) components. By reasoning similar to the previous situation, $K_1 + K_2$ is identically equal to $M_1 + M_2$ so that $K_2 = M_2$ from which it follows that $L_1 = L_2$.Q.E.D.

Theorem 4.6. An n-counter C is cyclic iff $N^i=I$ for some i.

Proof: If $N^i=I$ for some i, then C is clearly cyclic. Assume C is cyclic. Then for S_j any sequence of 0's and 1's, there is a j so that $N^j(S)=S$ by the definition of cyclic. Let i be the maximum of these j's. Then $N^i(S)=S$ for all S. Now N^i is a form on $(a_1,\ldots,a_n)$ and I is a form on $(a_1,\ldots,a_n)$. Since $N^i(S)=I(S)$ for any n-tuple S, then N^i is identically equal to I.

Theorem 4.7. If an n-counter C is cyclic, then $N^i=I$ for a smallest positive integer i. Further i = l.c.m. of the cycle lengths of C. Hence the length of each cycle divides i.

Proof: Let j be the l.c.m. of the cycle lengths of C. Then $N^j(S)=S$ for any output S since any S is in a cycle. Hence, $N^j=I$ so that $j\geq i$. If $j>i$, let ni be the largest multiple of i less than j. Then $j-ni<i$ so that $N^{j-ni}(S) = N^{j-ni}(N^{ni}(S)) = N^j(S) = S$ for all S. This contradicts the fact that i is the smallest positive integer such that $N^i=I$.

Corollary 4.8. If C is a cyclic counter and $N^p=I$ for p a prime, then C has at least one p-cycle and all the cycles of C are of length p or length 1.

Corollary 4.9. If C is a counter whose N transformation is a strict permutation, then C is cyclic. Further, the length of any cycle of C divides (2n)!

Proof: In this situation N is a permutation on 2n objects, hence an element of S_{2n}. So its order i must divide the order of S_{2n} which is (2n)! But this i is the smallest positive integer such that $N^i=I$.

Corollary 4.10. If C is cyclic, then $2^n = k_0 + k_1 c_1 + k_2 c_2 + .. + k_r c_r$ where k_i equals the number of cycles of length c_i and k_0 is the number of cycles of length one.

Note: If N is a permutation transformation, then $N^i = I$ iff the matrix of N to the ith power is the identity matrix. If N is a linear transformation this has to be modified as follows.

We assume N is a linear transformation. Then the columns of the matrix of N are in pairs and the transformation N can be read from its matrix. For example,

$$\begin{array}{ccc} a\bar{a} & b\bar{b} & c\bar{c} \\ \hline 10 & 00 & 01 \\ 01 & 00 & 10 \\ \hline 11 & 10 & 00 \\ 00 & 01 & 00 \\ \hline 00 & 11 & 00 \\ 00 & 00 & 00 \end{array}$$

gives $a \rightarrow a+b$, $\bar{a} \rightarrow \bar{a} + b$, $b \rightarrow b+c$, $\bar{b} \rightarrow \bar{b}+c$, $c \rightarrow \bar{a}$, $\bar{c} \rightarrow a$. Call two matrices equivalent if they define the same transformation. Another matrix equivalent to the one above is

$$\begin{array}{ccc} a\bar{a} & b\bar{b} & c\bar{c} \\ \hline 10 & 00 & 11 \\ 01 & 00 & 00 \\ \hline 01 & 10 & 10 \\ 10 & 01 & 10 \\ \hline 10 & 11 & 00 \\ 10 & 00 & 00 \end{array}$$

since this yields the transformation $a \rightarrow a+\bar{b}+c+\bar{c} = a+b$ since $c + \bar{c} = 1$, $\bar{a} \rightarrow \bar{a} + b$, $b \rightarrow b + c$, $\bar{b} \rightarrow \bar{b} + c$, $c \rightarrow a+b+\bar{b} = \bar{a}$, $\bar{c} \rightarrow a$. Then $N^i = I$ if and only if there is some matrix in the equivalence class of N^i which is the identiy matrix.

Theorem 4.10. Let C be an n-counter whose transformation is N. Let $j = \max_{\text{all n-tuples } S}$ (minimum distance to cycle).

If k is the l.c.m. of the cycles of C, then $k + j$ is the smallest positive integer such that $N^{k+j} = N^j$.

Proof: All the vertices of R which are in cycles = $\{N^j(S)$ for any n-tuple $S\}$. Hence $N^k(N^j(S)) = N^j(S)$ so that $N^{k+j}(S) = N^j(S)$ and by lemma 4.5, $N^{k+j} = N^j$. The proof of the fact that k+j is the smallest positive integer for which this is so is as in Theorem 4.7. (Note that when j=0, we have the special case $N^i=I$ discussed in Theorem 4.7).

Corollary 4.11. If k in the previous theorem is a prime p, then C has a cycle of length p and all cycles are either of length p or length 1.

Section 5. The Group of a Counter

We consider the symmetric group on n letters S_n to act on the 2n letters $a_1,\ldots,a_n,\bar{a}_1,\ldots,\bar{a}_n$ by having any permutation π in S_n act on $a_1,\ldots,a_n$ and simultaneously in the same fashion on $\bar{a}_1,\ldots,\bar{a}_n$. We let H be the group of permutations on the 2n letters $a_1,\ldots,a_n$, $\bar{a}_1,\ldots,\bar{a}_n$ generated by the above representation of S_n and the transposition $(a_1,\bar{a}_1)$. H also contains the transpositions $(a_2,\bar{a}_2)$, $(a_3,\bar{a}_3),\ldots,(a_n,\bar{a}_n)$.

Theorem 5.1 The order of H is $n!\ 2^n$.

Proof: Consider the homomorphism of H onto S_n gotten by identifying a_i and $\bar{a}_i$ for all i, $1 \leq\ i \leq n$. That this mapping is a well-defined homomorphism onto S_n follows from the definition of H. The kernel of this homomorphism is the subgroup of order 2^n generated by the n transpositions $(a_i,\bar{a}_i)$, $1 <\ i < n$. Hence the order of H = the order of S_n times $2^n = n!\ 2^n$.

Consider an n-counter C and its defining transformations N. N consists of n transformations $\varphi_i(a_1,\ldots,a_n)$, $i=1,\ldots,n$ where a_i goes into $\varphi_i(a_1,\ldots,a_n)$ and φ_i is either a permutation, linear transformation or a non-linear transformation of the form given in section 3. We consider a permutation π in H to send N into another N transformation by sending a_i into $\pi(a_i)$ and $\varphi_i(a_1,\ldots,a_n)$ into $\varphi_i\ (\pi(a_1),\ldots,\pi(a_n))$.

Theorem 5.2 If π is in H and N is the transformation of an n-counter C, then $\pi(N)$ is also the transformation of an n-counter.

Proof: This follows from the fact that $\overline{\pi(a_i)} = \pi(\bar{a}_i)$. Clearly π preserves permutations, linear transformations and the special form of non-linear transformations which the N transformations of a counter can assume.

Definition: If C is an n-counter and N is its transformation, then the subgroup G of H which sends N onto itself is called the group of N.

Definition: Two n-counters C_1 and C_2 with transformations N_1 and N_2 are called equivalent if $N_2 = \pi(N_1)$ for some π in H.

Theorem 5.3. If an n-counter C has group G, then the number of counters equivalent to C is $\dfrac{2^n x n!}{\text{order of } G}$.

Proof: This is so since every coset of G in H corresponds to a distinct counter equivalent to C and conversely.

We illustrate these ideas with examples.

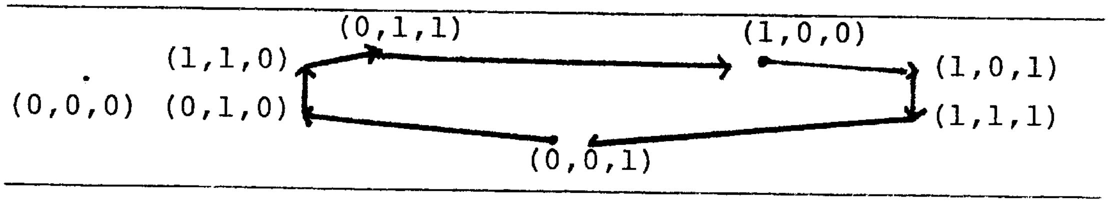
Example 4.1

N	a	b	c
	(a+b)	(b+c)	a

$N^7 = I$

C.L.	1	2	3
	(2,2)	(3,3)	(1,$\bar{1}$)

R:

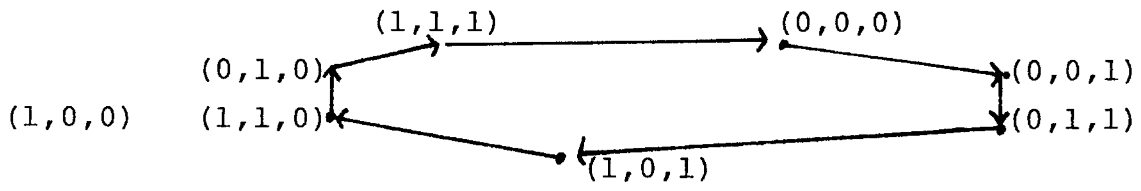


If $\pi = (a, \bar{a})$ then $\pi(N) = N_1$ is as follows.

$N_1 \begin{array}{ccc} a & b & c \\ \hline a+b & b+c & \bar{a} \end{array}$ [applying π to a we have $\frac{\bar{a}}{\bar{a}+b}$ so that $\frac{a}{a+b}$]

$N_1^7 = I$

R_1:



Note that R_1 can be obtained from R by complementing the coordinate of each triple (this is the permutation $(a,\bar{a})$).

C. L. $\begin{array}{ccc} 1 & 2 & 3 \\ \hline (2,2) & (3,3) & (1,\bar{1}) \end{array}$

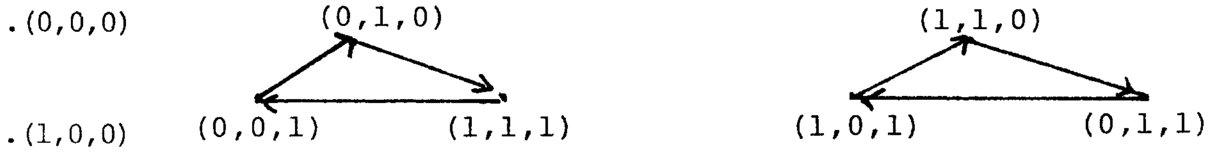
The only permutation in H sending N onto itself is the identity. Hence there are $2^3.3! = 48$ counters equivalent to N.

Example 4.2

C. L. $\frac{1}{(2,2)}$ $\frac{2}{(3,3)}$ $\frac{3}{(2,\bar{2})}$

N is $\begin{array}{ccc} a & b & c \\ \hline a+b & b+c & b \end{array}$ $N^3 = I$

R:



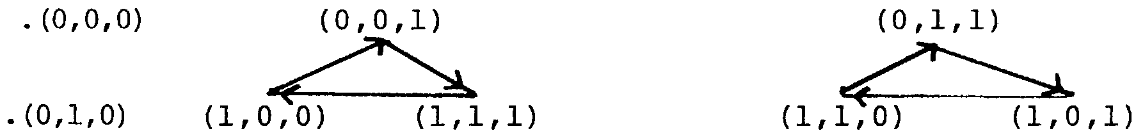
Here G is the group of order 2 consisting of the transposition (a, $\bar{a}$) and the identity. Hence there are 24 counters equivalent to N. Notice that complementing the first component yields an isomorphism of R.

The permutation (a, b, c) produces the following counter N_1 equivalent to N.

N_1	a	b	c
	c	b+c	a+c

CL.	1	2	3
	(3,3)	(3,3)	(1,1)

R_1



The points of R_1 could be calculated directly or by cyclically permuting the coordinates of R.

Example 4.3 is sent into itself by (a, $\bar{a}$) so that its group has order 2. Hence there are 24 counters equivalent to it. The group of example 4.4 is exactly I so that there are 48 counters equivalent to it.

Theorem 5.4 Equivalent counters have isomorphic graphs and isomorphic groups. Also if N_1 and N_2 are equivalent, then $N_1^i = N_1^j$ iff $N_2^i = N_2^j$.

Proof: If N_1 and N_2 are equivalent counters and π is the element of H such that $\pi(N_1) = N_2$, then π is the isomorphism which sends the graph of N_1 onto the graph of N_2. If G is the group of N_2 then $\pi^{-1} G \pi$ is the group of N_1 so they are isomorphic.

Theorem 5.5 If C is an n-counter, and G is its group, then each permutation in G is an isomorphism of the graph of C.

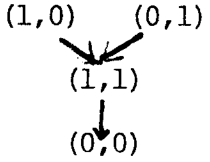
Note that there can be isomorphisms of the graph which do not arise from permutations in H.

Since equivalent counters have isomorphic graphs, if one were to search through all n-counters, for a fixed n, to find which cycle lengths are possible, it is enough to examine only one n-counter in each equivalence class. To show how this might be done, we completely classify all 2-counters in section 6.

Section 6. The Classification of all 2-counters

There are 16 different 2-counters. All N for 2-counters are linear. H here is the dihedral group of order 8. H is non-abelian.

1) $M = \left[\begin{array}{cc} 00 & 11 \\ 00 & 00 \\ \hline 11 & 00 \\ 00 & 00 \end{array}\right]$ N: $\frac{a}{a+b}$ $\frac{b}{a+b}$ G has order 2 and contains (a,b). Hence there are 4 counters in this equivalence class.

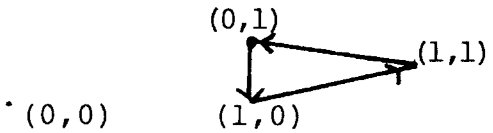


$N^3 = N^2$

CL. $\frac{1}{(2,2)}$ $\frac{2}{(1,1)}$

2) $M = \left[\begin{array}{cc} 00 & 10 \\ 00 & 01 \\ \hline 11 & 00 \\ 00 & 00 \end{array}\right]$ N: $\frac{a}{a+b}$ $\frac{b}{a}$ G = I. Hence there are 8 of these.

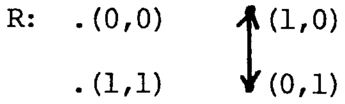
$N^3 = I$



C.L. $\frac{1}{(2,2)}$ $\frac{2}{(1,\bar{1})}$

3) $M = \left[\begin{array}{cc} 00 & 10 \\ 00 & 01 \\ \hline 10 & 00 \\ 01 & 00 \end{array}\right]$ N: $\frac{a}{b}$ $\frac{b}{a}$ G is a group of order 4 generated by the permutation (a,b) and $(a,\bar{a})$ $(b,\bar{b})$. Hence there are 2 of these

$N^2 = I$



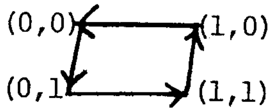
$\bar{\text{C}}$.L. $\frac{1}{(2,\bar{2})}$ $\frac{2}{(1,\bar{1})}$

4) $M = \left[\begin{array}{cc} 00 & 01 \\ 00 & 10 \\ \hline 10 & 00 \\ 01 & 00 \end{array}\right]$ N: $\frac{a}{b}$ $\frac{b}{\bar{a}}$ G is a group of order 4 generated by the permutation $(a, b, \bar{a}, \bar{b})$. Hence there are 2 of these.

$N^4 = I$.

C.L. $\frac{1}{(2,\bar{2})}$ $\frac{2}{(\bar{1},1)}$

R:



These 4 cases add up to 16 counters. Hence these are all. Counters 3) and 4) have permutation N's so that for these cases the matrix of N equals the matrix of M. All counters in classes 2, 3, and 4 are cyclic.

A J-K flip-flop is a particular type of 2-input, 2-output finite-state automation. For more on this see [1] and [3].

1. Hennie, Frederick C., Finite-State Models for Logical Machines , John Wiley Sons, Inc., New York, London, and Sydney, 1968.
2. Manning, Frank, "Autonomous, Synchronous Counters Constructed only of J-K Flips",MAC TR-96, May 1972.
3. Minsky, Marvin, Computation; Finite and Infinite Machines, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1967.

BIBLIOGRAPHIC DATA SHEET	1. Report No. MAC TM- 47	2.	3. Recipient's Accession No.

4. Title and Subtitle

Mathematical Foundations of Flip-Flops

5. Report Date : Issued June 1974

6.

7. Author(s)
Vera Pless

8. Performing Organization Rept. No. MAC TM- 47

9. Performing Organization Name and Address

PROJECT MAC; MASSACHUSETTS INSTITUTE OF TECHNOLOGY:
545 Technology Square, Cambridge, Massachusetts 02139

10. Project/Task/Work Unit No.

11. Contract/Grant No.
N00014-70-A-0362-0006

12. Sponsoring Organization Name and Address
Office of Naval Research
Department of the Navy
Information Systems Program
Arlington, Va 22217

13. Type of Report & Period Covered: Interim Scientific Report

14.

15. Supplementary Notes

16. Abstracts

The main purpose of this paper is to lay a mathematical basis for the study of flip-flops. This approach will lead, in further studies, to important practical results although the alert reader can see some practical applications in this work.

A J-K flip-flop is a device with 2 inputs (zero or one) and two outputs, one of which is always the complement of the other. A counter is a set of J-K flip-flops whose inputs are either constants (0 or 1) or outputs of other flip-flops in the counter. In this paper we only consider J-K flip-flops whose inputs are outputs of the other flip-flops. Whenever we refer to an n-counter, we shall mean such a counter with no constant inputs.The specification of the connections is called connection list.

17. Key Words and Document Analysis. 17a. Descriptors

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement
Approved for Public Release;
Distribution Unlimited

19. Security Class (This Report)
UNCLASSIFIED

20. Security Class (This Page
UNCLASSIFIED

21. No. of Pages
33

22. Price

THIS FORM MAY BE REPRODUCED